Institiúid Teicneolaíochta Cheatharlach

INSTITUTE of TECHNOLOGY CARLOW

At the Heart of South Leinster

Technical Document

Diet Track

Diet Tracker Application

Student Name: Emmanuel Ayelabola

Student ID: C00242748

Course: Bachelor of Science (Honours) Software Development

Supervisor: Dr Chris Staff

Submission Date: 25/04/2022

Contents

Add Goal Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.drawable.GradientDrawable;
import android.os.Bundle;
import android.text.InputType;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.ScaleAnimation;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.google.gson.Gson;
import com.sdsmdg.tastytoast.TastyToast;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Random;


public class AddgoalActivity extends AppCompatActivity {

   private String Date = "";
   private String Time = "";
   private HashMap<String, Object> AddMeal = new HashMap<>();
   private HashMap<String, Object> AddGoal = new HashMap<>();
   private String tag = "";
```

```
   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();
   private ArrayList<String> Classes = new ArrayList<>();
   private ArrayList<HashMap<String, Object>> Goal_List = new
ArrayList<>();

   private LinearLayout toolbar;
   private LinearLayout main;
   private ImageView back_img;
   private LinearLayout line;
   private LinearLayout linear1;
   private LinearLayout check_lin;
   private LinearLayout linrSelectTemp;
   private LinearLayout linrSavingGoal;
   private ScrollView vscroll1;
   private LinearLayout linear19;
   private TextView textview17;
   private LinearLayout linear20;
   private LinearLayout linear27;
   private TextView textview26;
   private LinearLayout linrCustom;
   private LinearLayout linrWeight;
   private LinearLayout linrReduceWeight;
   private LinearLayout linear25;
   private TextView txtWeight;
   private LinearLayout linear26;
   private ImageView imageview9;
   private TextView txtIncreaseWeight;
   private LinearLayout linear23;
   private TextView txtReduceWeight;
   private LinearLayout linear22;
   private ImageView imageview8;
   private TextView txtReduceWeigt;
   private LinearLayout linrGainblood;
   private LinearLayout linrProtein;
   private LinearLayout linear30;
   private TextView txtGainMBlood;
   private LinearLayout linear31;
   private ImageView imageview10;
   private TextView txtGainMoreBlood;
   private LinearLayout linear32;
   private TextView txttagProtein;
   private LinearLayout linear33;
   private ImageView imageview11;
   private TextView txtProtein;
   private ImageView imageview12;
   private TextView textview24;
   private ScrollView vscroll2;
   private LinearLayout linear36;
   private ImageView imageview13;
   private TextView textview27;
   private LinearLayout linear38;
   private LinearLayout linear42;
   private LinearLayout linear46;
   private LinearLayout linear49;
   private LinearLayout linear39;
   private Button button1;
   private LinearLayout linear37;
   private EditText editGoal;
   private ImageView imageview14;
```

```
   private TextView textview28;
   private LinearLayout linear43;
   private EditText editGoalSub;
   private ImageView imageview16;
   private TextView textview30;
   private LinearLayout linear47;
   private LinearLayout linear48;
   private ImageView imageview18;
   private TextView textview32;
   private EditText editStartDay;
   private SeekBar seekbar1;
   private EditText editEndDay;
   private TextView textview33;
   private LinearLayout linear50;
   private TextView textview34;
   private CheckBox importantcheckbox;
   private TextView characters;

   private Calendar Cal = Calendar.getInstance();
   private SharedPreferences AllMeals;
   private SharedPreferences Settings;
   private SharedPreferences AllGoals;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.addgoal);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);

      //hide completed checkBox linear
      check_lin.setVisibility(View.GONE);


      //Design the edit text for goal title and goal subtitle
      editGoal.setBackground(new GradientDrawable() {
         public GradientDrawable getIns(int a, int b) {
            this.setCornerRadius(a);
            this.setColor(b);
            return this;
         } }.getIns((int)15, 0xFFE0E0E0));
      editGoalSub.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) {
         this.setCornerRadius(a);
         this.setColor(b);
         return this;
      } }.getIns((int)15, 0xFFE0E0E0));
      button1.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) {
         this.setCornerRadius(a);
         this.setColor(b);
         return this;
      } }.getIns((int)15, 0xFF3F51B6));
      editEndDay.setEnabled(false);
      editStartDay.setEnabled(false);
   }

   private void initialize(Bundle savedInstanceState) {
      toolbar =  findViewById(R.id.toolbar);
      main =  findViewById(R.id.main);
      back_img =  findViewById(R.id.back_img);
```

```
        line =  findViewById(R.id.line);
        linear1 =  findViewById(R.id.linear1);
        check_lin =  findViewById(R.id.check_lin);
        linrSelectTemp =  findViewById(R.id.linrSelectTemp);
        linrSavingGoal =  findViewById(R.id.linrSavingGoal);
        vscroll1 =  findViewById(R.id.vscroll1);
        linear19 =  findViewById(R.id.linear19);
        textview17 = (TextView) findViewById(R.id.textview17);
        linear20 =  findViewById(R.id.linear20);
        linear27 =  findViewById(R.id.linear27);
        textview26 = (TextView) findViewById(R.id.textview26);
        linrCustom =  findViewById(R.id.linrCustom);
        linrWeight =  findViewById(R.id.linrWeight);
        linrReduceWeight =  findViewById(R.id.linrReduceWeight);
        linear25 =  findViewById(R.id.linear25);
        txtWeight = (TextView) findViewById(R.id.txtWeight);
        linear26 =  findViewById(R.id.linear26);
        imageview9 =  findViewById(R.id.imageview9);
        txtIncreaseWeight = (TextView) findViewById(R.id.txtIncreaseWeight);
        linear23 =  findViewById(R.id.linear23);
        txtReduceWeight = (TextView) findViewById(R.id.txtReduceWeight);
        linear22 =  findViewById(R.id.linear22);
        imageview8 =  findViewById(R.id.imageview8);
        txtReduceWeigt = (TextView) findViewById(R.id.txtReduceWeigt);
        linrGainblood =  findViewById(R.id.linrGainblood);
        linrProtein =  findViewById(R.id.linrProtein);
        linear30 =  findViewById(R.id.linear30);
        txtGainMBlood = (TextView) findViewById(R.id.txtGainMBlood);
        linear31 =  findViewById(R.id.linear31);
        imageview10 =  findViewById(R.id.imageview10);
        txtGainMoreBlood = (TextView) findViewById(R.id.txtGainMoreBlood);
        linear32 =  findViewById(R.id.linear32);
        txttagProtein = (TextView) findViewById(R.id.txttagProtein);
        linear33 =  findViewById(R.id.linear33);
        imageview11 =  findViewById(R.id.imageview11);
        txtProtein = (TextView) findViewById(R.id.txtProtein);
        imageview12 =  findViewById(R.id.imageview12);
        textview24 = (TextView) findViewById(R.id.textview24);
        vscroll2 =  findViewById(R.id.vscroll2);
        linear36 =  findViewById(R.id.linear36);
        imageview13 =  findViewById(R.id.imageview13);
        textview27 = (TextView) findViewById(R.id.textview27);
        linear38 =  findViewById(R.id.linear38);
        linear42 =  findViewById(R.id.linear42);
        linear46 =  findViewById(R.id.linear46);
        linear49 =  findViewById(R.id.linear49);
        linear39 =  findViewById(R.id.linear39);
        button1 = (Button) findViewById(R.id.button1);
        linear37 =  findViewById(R.id.linear37);
        editGoal = (EditText) findViewById(R.id.editGoal);
        imageview14 =  findViewById(R.id.imageview14);
        textview28 = (TextView) findViewById(R.id.textview28);
        linear43 =  findViewById(R.id.linear43);
        editGoalSub = (EditText) findViewById(R.id.editGoalSub);
        imageview16 =  findViewById(R.id.imageview16);
        textview30 = (TextView) findViewById(R.id.textview30);
        linear47 =  findViewById(R.id.linear47);
        linear48 =  findViewById(R.id.linear48);
        imageview18 =  findViewById(R.id.imageview18);
        textview32 = (TextView) findViewById(R.id.textview32);
        editStartDay = (EditText) findViewById(R.id.editStartDay);
```

```
        seekbar1 = (SeekBar) findViewById(R.id.seekbar1);
        editEndDay = (EditText) findViewById(R.id.editEndDay);
        textview33 = (TextView) findViewById(R.id.textview33);
        linear50 =  findViewById(R.id.linear50);
        textview34 = (TextView) findViewById(R.id.textview34);
        importantcheckbox = (CheckBox) findViewById(R.id.importantcheckbox);
        characters = (TextView) findViewById(R.id.characters);
        AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);
        Settings = getSharedPreferences("Settings", Activity.MODE_PRIVATE);
        AllGoals = getSharedPreferences("AllGoals", Activity.MODE_PRIVATE);

        //Exit the activity onClick
        back_img.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                finish();
            }
        });


        //Hide the linearLayout: template for creating goal(fragment1)
        // Show the linearLayout for entering goal manually
        linrCustom.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                linrSelectTemp.setVisibility(View.GONE);
                linrSavingGoal.setVisibility(View.VISIBLE);
                clickAnimation(linrCustom);
            }
        });

        //Custom goal: set Goal for weight
        //Hide the linearLayout: template for creating goal(fragment1)
        //EditText Title get the text for Weight goal

        linrWeight.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                clickAnimation(linrWeight);
                editGoal.setText(txtWeight.getText().toString());
                linrSelectTemp.setVisibility(View.GONE);
                linrSavingGoal.setVisibility(View.VISIBLE);
                editGoal.setEnabled(false);
                tag = txtIncreaseWeight.getText().toString();
            }
        });


        //Custom goal: set Goal for Reduce Weight
        //Hide the linearLayout: template for creating goal(fragment1)
        //EditText Title get the text for Reduce Weight goal
        linrReduceWeight.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                linrSelectTemp.setVisibility(View.GONE);
                linrSavingGoal.setVisibility(View.VISIBLE);
                editGoal.setText(txtReduceWeight.getText().toString());
                editGoal.setEnabled(false);
                clickAnimation(linrReduceWeight);
                tag = txtReduceWeigt.getText().toString();
            }
```

```
        });

        //Custom goal: set Goal for Reduce Weight
        //Hide the linearLayout: template for creating goal(fragment1)
        //EditText Title get the text for Gain Blood goal
        linrGainblood.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                linrSelectTemp.setVisibility(View.GONE);
                linrSavingGoal.setVisibility(View.VISIBLE);
                editGoal.setEnabled(false);
                editGoal.setText(txtGainMBlood.getText().toString());
                clickAnimation(linrGainblood);
                tag = txtGainMoreBlood.getText().toString();
            }
        });

        //Custom goal: set Goal for Reduce Weight
        //Hide the linearLayout: template for creating goal(fragment1)
        //EditText Title get the text for Protein goal
        linrProtein.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                linrSelectTemp.setVisibility(View.GONE);
                linrSavingGoal.setVisibility(View.VISIBLE);
                editGoal.setText(txttagProtein.getText().toString());
                editGoal.setEnabled(false);
                clickAnimation(linrProtein);
                tag = txttagProtein.getText().toString();
            }
        });


        //Button saves the entered goal details by the user
        //Creating a HashMap(AddGoal) to get different detail
        button1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                AddGoal = new HashMap<>();
                AddGoal.put("Goal", editGoal.getText().toString());
                AddGoal.put("GoalSub", editGoalSub.getText().toString());
                AddGoal.put("Tag", tag);
                AddGoal.put("Date", Date);
                AddGoal.put("Time", Time);
                AddGoal.put("Start", editStartDay.getText().toString());
                AddGoal.put("End", editEndDay.getText().toString());
                if (importantcheckbox.isChecked()) {
                    AddGoal.put("Completed", "True");
                }
                else {
                    AddGoal.put("Completed", "False");
                }
                //Arraylist(Goal_List to get the different data to a list
                Goal_List.add(AddGoal);
                //Then save the data in the list to Json
                AllGoals.edit().putString("Goals", new
Gson().toJson(Goal_List)).commit();
                TastyToast.makeText(getApplicationContext(), "Your Goal Have
Been Successfully Added ", TastyToast.LENGTH_LONG, TastyToast.SUCCESS);
                finish();
            }
```

```
      });

      //onProgressChanged of this seekbar extends the target date for goal
created
      //Max value of seekbar is 365(days)
      //Min is 4(days)
      seekbar1.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
         @Override
         public void onProgressChanged (SeekBar param1, int param2, boolean
param3) {
            final int progressValue = param2;
            editEndDay.setText(String.valueOf((long)(progressValue)));
         }

         @Override
         public void onStartTrackingTouch(SeekBar param1) {

         }

         @Override
         public void onStopTrackingTouch(SeekBar param2) {

         }
      });
   }


   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   public void Typeface () {

   }


   public void Capitalize (final TextView Edittext) {

Edittext.setRawInputType(InputType.TYPE_CLASS_TEXT|InputType.TYPE_TEXT_FLAG
_CAP_SENTENCES);
   }


   public void getSettingsData () {

   }


   public void textSize (final TextView TextView1, final double size) {
      int j = (int) size;
      TextView1.setTextSize(j);
   }
```

```
   public void Add (final String Colour, final ImageView Imageview) {
      Imageview.getDrawable().setColorFilter(Color.parseColor(Colour),
PorterDuff.Mode.SRC_IN);
   }


   public void clickAnimation (final View view) {
      ScaleAnimation fade_in = new ScaleAnimation(0.9f, 1f, 0.9f, 1f,
Animation.RELATIVE_TO_SELF, 0.5f, Animation.RELATIVE_TO_SELF, 0.7f);
      fade_in.setDuration(300);
      fade_in.setFillAfter(true);
      view.startAnimation(fade_in);
   }


   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }

   @Deprecated
   public int getRandom(int min, int max) {
      Random random = new Random();
      return random.nextInt(max - min + 1) + min;
   }

   @Deprecated
   public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
      ArrayList<Double> result = new ArrayList<Double>();
      SparseBooleanArray arr = _list.getCheckedItemPositions();
      for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
         if (arr.valueAt(iIdx))
         result.add((double)arr.keyAt(iIdx));
      }
      return result;
   }

   @Deprecated
   public float getDip(int input) {
      return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
   }

   @Deprecated
```

```
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}


package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748
//
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.drawable.GradientDrawable;
import android.os.Bundle;
import android.text.Editable;
import android.text.InputType;
import android.text.TextWatcher;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import com.sdsmdg.tastytoast.TastyToast;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Random;


public class AddMealActivity extends AppCompatActivity {

    private String Date = "";
    private String Time = "";
    private HashMap<String, Object> AddMeal = new HashMap<>();
```

```
   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();
   private ArrayList<String> Classes = new ArrayList<>();

   private LinearLayout toolbar;
   private LinearLayout main;
   private ImageView back_img;
   private EditText title_edittext;
   private LinearLayout line;
   private LinearLayout linear1;
   private LinearLayout check_lin;
   private ScrollView vscroll1;
   private LinearLayout linear2;
   private LinearLayout linear5;
   private EditText edit_ingredient;
   private LinearLayout linear4;
   private EditText edit_method;
   private LinearLayout linear3;
   private LinearLayout linear6;
   private LinearLayout linear7;
   private EditText edit_calories;
   private LinearLayout linear10;
   private EditText editServing;
   private LinearLayout linear9;
   private TextView txtClassOfFood;
   private Spinner spinner1;
   private LinearLayout linear8;
   private Button button1;
   private ImageView imageview3;
   private TextView textview1;
   private ImageView imageview2;
   private TextView textview2;
   private ImageView imageview1;
   private TextView textview3;
   private EditText note_edittext;
   private EditText editminutes;
   private ImageView imageview4;
   private TextView textview4;
   private ImageView imageview6;
   private TextView textview6;
   private ImageView imageview5;
   private TextView textview5;
   private CheckBox importantcheckbox;
   private TextView characters;

   private Calendar Cal = Calendar.getInstance();
   private SharedPreferences AllMeals;
   private SharedPreferences Settings;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.add_meal);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      toolbar =  findViewById(R.id.toolbar);
```

```
      main =  findViewById(R.id.main);
      back_img =  findViewById(R.id.back_img);
      title_edittext = (EditText) findViewById(R.id.title_edittext);
      line =  findViewById(R.id.line);
      linear1 =  findViewById(R.id.linear1);
      check_lin =  findViewById(R.id.check_lin);
      vscroll1 =  findViewById(R.id.vscroll1);
      linear2 =  findViewById(R.id.linear2);
      linear5 =  findViewById(R.id.linear5);
      edit_ingredient = (EditText) findViewById(R.id.edit_ingredient);
      linear4 =  findViewById(R.id.linear4);
      edit_method = (EditText) findViewById(R.id.edit_method);
      linear3 =  findViewById(R.id.linear3);
      linear6 =  findViewById(R.id.linear6);
      linear7 =  findViewById(R.id.linear7);
      edit_calories = (EditText) findViewById(R.id.edit_calories);
      linear10 =  findViewById(R.id.linear10);
      editServing = (EditText) findViewById(R.id.editServing);
      linear9 =  findViewById(R.id.linear9);
      txtClassOfFood = (TextView) findViewById(R.id.txtClassOfFood);
      spinner1 = (Spinner) findViewById(R.id.spinner1);
      linear8 =  findViewById(R.id.linear8);
      button1 = (Button) findViewById(R.id.button1);
      imageview3 =  findViewById(R.id.imageview3);
      textview1 = (TextView) findViewById(R.id.textview1);
      imageview2 =  findViewById(R.id.imageview2);
      textview2 = (TextView) findViewById(R.id.textview2);
      imageview1 =  findViewById(R.id.imageview1);
      textview3 = (TextView) findViewById(R.id.textview3);
      note_edittext = (EditText) findViewById(R.id.note_edittext);
      editminutes = (EditText) findViewById(R.id.editminutes);
      imageview4 =  findViewById(R.id.imageview4);
      textview4 = (TextView) findViewById(R.id.textview4);
      imageview6 =  findViewById(R.id.imageview6);
      textview6 = (TextView) findViewById(R.id.textview6);
      imageview5 =  findViewById(R.id.imageview5);
      textview5 = (TextView) findViewById(R.id.textview5);
      importantcheckbox = (CheckBox) findViewById(R.id.importantcheckbox);
      characters = (TextView) findViewById(R.id.characters);
      AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);
      Settings = getSharedPreferences("Settings", Activity.MODE_PRIVATE);

      //onBackPressed and btnCreateMeal will perform click on back_img,
which will collect and save the user
      //entered food.
      back_img.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            if (title_edittext.getText().toString().equals("") &&
note_edittext.getText().toString().equals("")) {
               ToastUtil.showMessage(getApplicationContext(), "Enter Meal
Name !!");
               android.view.inputmethod.InputMethodManager imm =
(android.view.inputmethod.InputMethodManager)getSystemService(INPUT_METHOD_
SERVICE);

imm.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(), 0);
            }
            else {
               if (title_edittext.getText().toString().equals("") &&
!edit_ingredient.getText().toString().equals("")) {
```

```
                    ToastUtil.showMessage(getApplicationContext(), "Enter
Meal Name !!");
                }
                else {
                    if (edit_ingredient.getText().toString().equals("") &&
!title_edittext.getText().toString().equals("")) {
                        ToastUtil.showMessage(getApplicationContext(), "Enter
Your Meal Ingredients !!");
                    }
                    else {
                        if ((note_edittext.getText().toString().length() > 2)
|| (editminutes.getText().toString().length() > 2)) {
                            ToastUtil.showMessage(getApplicationContext(),
"Time can't be more than that");
                        }
                        else {

                            //Create a new HashMap to save the data collected
                            //Set a key for each data such "Ingredients" for
input of ingredients entered by user
                            AddMeal = new HashMap<>();
                            AddMeal.put("Note",
note_edittext.getText().toString());
                            AddMeal.put("Title",
title_edittext.getText().toString());
                            AddMeal.put("Method",
edit_method.getText().toString());
                            AddMeal.put("Ingredients",
edit_ingredient.getText().toString());
                            AddMeal.put("Minutes",
editminutes.getText().toString());
                            AddMeal.put("Calories",
edit_calories.getText().toString());
                            AddMeal.put("Hours",
note_edittext.getText().toString());
                            AddMeal.put("Class",
txtClassOfFood.getText().toString());
                            AddMeal.put("Serving",
editServing.getText().toString());

                            //When true it adds the item to important list
                            if (importantcheckbox.isChecked()) {
                                AddMeal.put("Important", "True");
                            }
                            else {
                                AddMeal.put("Important", "False");
                            }
                            AddMeal.put("Date", Date);
                            AddMeal.put("Time", Time);
                            AddMeal.put("Select", "False");

                            //Create an ArrayList HashMap to save all the data
from the HashMap created

                            Notes_List.add(AddMeal);

                            //Create a sharedPreference, note that the
ArrayList HashMap is converted to Json
                            //Save the data into the sharedPreference

                            AllMeals.edit().putString("Meals", new
```

```
Gson().toJson(Notes_List)).commit();
                        TastyToast.makeText(getApplicationContext(), "Your
Meal Have Been Successfully Added ", TastyToast.LENGTH_LONG,
TastyToast.SUCCESS);
                        finish();
                    }
                }
            }
        }
    });

    //This spinner selects the type of food class and if null, it's saved
as unknown

    spinner1.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> param1, View param2, int
param3, long param4) {
            final int position = param3;
            //Sets the textFoodCLass to Unknown
            if (position == 0) {
                txtClassOfFood.setText("Unknown");
                spinner1.setSelection((int)(0));
            }

            //Sets the textFoodCLass to Carbohydrate
            if (position == 1) {
                txtClassOfFood.setText(Classes.get((int)(position)));
                spinner1.setSelection((int)(1));
            }
            //Sets the textFoodCLass to protein
            if (position == 2) {
                txtClassOfFood.setText(Classes.get((int)(position)));
                spinner1.setSelection((int)(2));
            }
            //Sets the textFoodCLass to Fats And Oil
            if (position == 3) {
                txtClassOfFood.setText(Classes.get((int)(position)));
                spinner1.setSelection((int)(3));
            }

            //Sets the textFoodCLass to Mineral
            if (position == 4) {
                txtClassOfFood.setText(Classes.get((int)(position)));
                spinner1.setSelection((int)(4));
            }
            //Sets the textFoodCLass to Vitamins
            if (position == 5) {
                txtClassOfFood.setText(Classes.get((int)(position)));
                spinner1.setSelection((int)(5));
            }
            //Sets the textFoodCLass to Fiber
            if (position == 6) {
                txtClassOfFood.setText(Classes.get((int)(position)));
                spinner1.setSelection((int)(6));
            }
        }

        @Override
```

```
        public void onNothingSelected(AdapterView<?> param1) {

        }
      });

      //back_img performs click which saves the entered data
      button1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            back_img.performClick();
         }
      });

      note_edittext.addTextChangedListener(new TextWatcher() {
         @Override
         public void onTextChanged(CharSequence param1, int param2, int
param3, int param4) {
            final String _charSeq = param1.toString();
            if (_charSeq.length() > 2) {

            }
         }

         @Override
         public void beforeTextChanged(CharSequence param1, int param2, int
param3, int param4) {

         }

         @Override
         public void afterTextChanged(Editable param1) {

         }
      });
   }

   private void initializeLogic() {
      ONCREATE();

      //Spinner data: Classes of food
      //Creates an ArrayList String and then add the different types of
food classes

      Classes.add("Select Food Class");
      Classes.add("Carbohydrate");
      Classes.add("Protein");
      Classes.add("Fats And Oil");
      Classes.add("Minerals");
      Classes.add("Vitamins");
      Classes.add("Fibre And Roughage's");

      //setAdapter for the spinner
      spinner1.setAdapter(new ArrayAdapter<String>(getBaseContext(),
android.R.layout.simple_spinner_dropdown_item, Classes));
      ((ArrayAdapter)spinner1.getAdapter()).notifyDataSetChanged();

      //Designing
      edit_calories.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)30, 0xFFEEEEEE));
      note_edittext.setBackground(new GradientDrawable() { public
```

```
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)30, 0xFFEEEEEE));
      button1.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFF3F51B6));
      editminutes.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)30, 0xFFEEEEEE));
      editServing.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)30, 0xFFEEEEEE));
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   @Override
   public void onBackPressed() {
      back_img.performClick();
   }
   public void Typeface () {

   }


   public void ONCREATE () {

      //Get current date
      Cal = Calendar.getInstance();
      Typeface();
      if (!AllMeals.getString("Meals", "").equals("")) {
         Notes_List = new Gson().fromJson(AllMeals.getString("Meals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      }

      //This is useful to let the user know when the data is created
      Date = new SimpleDateFormat("dd MMM").format(Cal.getTime());
      Time = new SimpleDateFormat("hh:mm a").format(Cal.getTime());
      importantcheckbox.setChecked(false);
      Capitalize(title_edittext);
      Capitalize(note_edittext);

   }


   public void Capitalize (final TextView Edittext) {

Edittext.setRawInputType(InputType.TYPE_CLASS_TEXT|InputType.TYPE_TEXT_FLAG
_CAP_SENTENCES);
   }
```

```
   public void textSize (final TextView TextView1, final double size) {
      int j = (int) size;
      TextView1.setTextSize(j);
   }


   public void Add (final String Colour, final ImageView Imageview) {
      Imageview.getDrawable().setColorFilter(Color.parseColor(Colour),
PorterDuff.Mode.SRC_IN);
   }


   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }

   @Deprecated
   public int getRandom(int min, int max) {
      Random random = new Random();
      return random.nextInt(max - min + 1) + min;
   }

   @Deprecated
   public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
      ArrayList<Double> result = new ArrayList<Double>();
      SparseBooleanArray arr = _list.getCheckedItemPositions();
      for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
         if (arr.valueAt(iIdx))
         result.add((double)arr.keyAt(iIdx));
      }
      return result;
   }

   @Deprecated
   public float getDip(int input) {
      return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
   }

   @Deprecated
   public int getDisplayWidthPixels() {
      return getResources().getDisplayMetrics().widthPixels;
   }
```

```
    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Create Account Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.DatePickerDialog;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.DialogFragment;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
```

```
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.GenericTypeIndicator;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Random;


public class CreateaccountActivity extends AppCompatActivity {
   private FirebaseDatabase firebase = FirebaseDatabase.getInstance();

   private String fontName = "";
   private String typeace = "";
   private double num = 0;
   private HashMap<String, Object> sa_map = new HashMap<>();
   private String name1 = "";
   private HashMap<String, Object> user = new HashMap<>();

   private ArrayList<String> gender = new ArrayList<>();

   private LinearLayout linear_bg;
   private LinearLayout linear_log_bg;
   private LinearLayout linearsign_bg;
   private ImageView imageview1;
   private LinearLayout linear_log;
   private TextView textview1;
   private EditText edittext1;
   private TextView textview3;
   private LinearLayout linear_email;
   private TextView textview2;
   private LinearLayout linear_pass;
   private Button button_login;
   private LinearLayout linear5;
   private LinearLayout linear4;
   private EditText in_email;
   private EditText in_pass;
   private ImageView imgViewPassword;
   private TextView textview4;
   private TextView signup;
   private LinearLayout linear14;
   private LinearLayout linearsign;
   private ImageView imageview;
   private ScrollView vscroll1;
   private Button buttonsignup;
   private LinearLayout linear12;
   private LinearLayout linear15;
   private TextView textview5;
   private EditText edittext21;
   private TextView textview9;
   private LinearLayout linear_name;
   private TextView textview6;
   private LinearLayout lin_email;
   private TextView textview10;
   private LinearLayout linr_age;
   private TextView textview11;
   private LinearLayout linr_height;
   private TextView textview12;
   private LinearLayout linr_weight;
   private TextView txt_gender;
```

```
   private LinearLayout linr_gender;
   private TextView textview7;
   private LinearLayout lin_pass;
   private EditText up_name;
   private EditText up_email;
   private EditText edittext2;
   private ImageView imgAge;
   private EditText up_height;
   private EditText up_weight;
   private Spinner spin_gender;
   private EditText edittext4;
   private EditText up_pass;
   private ImageView imageview_up;
   private TextView textview8;
   private TextView signUp;

   private Intent sa_int = new Intent();
   private FirebaseAuth auth;
   private OnCompleteListener<Void> auth_updateEmailListener;
   private OnCompleteListener<Void> auth_updatePasswordListener;
   private OnCompleteListener<Void> auth_emailVerificationSentListener;
   private OnCompleteListener<Void> auth_deleteUserListener;
   private OnCompleteListener<Void> auth_updateProfileListener;
   private OnCompleteListener<AuthResult> auth_phoneAuthListener;
   private OnCompleteListener<AuthResult> auth_googleSignInListener;
   private OnCompleteListener<AuthResult> auth_create_user_listener;
   private OnCompleteListener<AuthResult> authsign_in_listener;
   private OnCompleteListener<Void> _auth_reset_password_listener;
   private DatabaseReference sa_db = firebase.getReference("sa_db");
   private ChildEventListener sa_db_child_listener;
   private SharedPreferences name;
   private Calendar c = Calendar.getInstance();

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.createaccount);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);

      //Check if the user is logged in already and if logged in
      //Redirect to MainActivity
         sa_db.removeEventListener(sa_db_child_listener);
         if ((FirebaseAuth.getInstance().getCurrentUser() != null)) {
            sa_int.setClass(getApplicationContext(), MainActivity.class);
            sa_int.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(sa_int);
            finish();
         }
         linearsign_bg.setVisibility(View.GONE);
         edittext2.setEnabled(false);
         {
            android.graphics.drawable.GradientDrawable designUI = new
android.graphics.drawable.GradientDrawable();
            int d = (int)
getApplicationContext().getResources().getDisplayMetrics().density;
            designUI.setColor(0xFFF2F2F2);designUI.setCornerRadii(new
float[]{
               d*30,d*30,d*30 ,d*30,0,0 ,0,0});
            android.graphics.drawable.RippleDrawable designUIRD = new
android.graphics.drawable.RippleDrawable(new
```

```
                android.content.res.ColorStateList(new int[][]{new
int[]{}}, new int[]{0xFFF2F2F2}), designUI, null);
            linear_log.setBackground(designUIRD);
        }


        {
            android.graphics.drawable.GradientDrawable designUI = new
android.graphics.drawable.GradientDrawable();
            int d = (int)
getApplicationContext().getResources().getDisplayMetrics().density;
            designUI.setColor(0xFFF2F2F2);designUI.setCornerRadii(new
float[]{
               d*30,d*30,d*30 ,d*30,0,0 ,0,0});
            android.graphics.drawable.RippleDrawable designUIRD = new
android.graphics.drawable.RippleDrawable(
                new android.content.res.ColorStateList(new int[][]{new
int[]{}},
                      new int[]{0xFFF2F2F2}), designUI, null);
            linearsign.setBackground(designUIRD);
        }

        darkIcons();
        removeScrollBar(vscroll1);
        changeActivityFont("railway_semisolid");
        ICC(imgViewPassword, "#FF615CB7", "#F2F2F2");
        ICC(imageview_up, "#FF615CB7", "#F2F2F2");
        round(linear_name, 25);
        round(linear_email, 25);
        round(linear_pass, 25);
        round(button_login, 25);
        round(lin_email, 25);
        round(lin_pass, 25);
        round(buttonsignup, 25);
        round(linr_age, 25);
        round(linr_height, 25);
        round(linr_weight, 25);
        round(linr_gender, 25);

        //Add gender to spinner
        gender.add("Select Gender");
        gender.add("Male");
        gender.add("Female");
        spin_gender.setAdapter(new ArrayAdapter<String>(getBaseContext(),
android.R.layout.simple_spinner_dropdown_item, gender));
        ((ArrayAdapter)spin_gender.getAdapter()).notifyDataSetChanged();

    }

    private void initialize(Bundle savedInstanceState) {
        linear_bg =  findViewById(R.id.linear_bg);
        linear_log_bg =  findViewById(R.id.linear_log_bg);
        linearsign_bg =  findViewById(R.id.linearsign_bg);
        imageview1 =  findViewById(R.id.imageview1);
        linear_log =  findViewById(R.id.linear_log);
        textview1 = (TextView) findViewById(R.id.textview1);
        edittext1 = (EditText) findViewById(R.id.edittext1);
        textview3 = (TextView) findViewById(R.id.textview3);
        linear_email =  findViewById(R.id.linear_email);
        textview2 = (TextView) findViewById(R.id.textview2);
        linear_pass =  findViewById(R.id.linear_pass);
```

```
      button_login = (Button) findViewById(R.id.button_login);
      linear5 =  findViewById(R.id.linear5);
      linear4 =  findViewById(R.id.linear4);
      in_email = (EditText) findViewById(R.id.in_email);
      in_pass = (EditText) findViewById(R.id.in_pass);
      imgViewPassword =  findViewById(R.id.imgViewPassword);
      textview4 = (TextView) findViewById(R.id.textview4);
      signup = (TextView) findViewById(R.id.signup);
      linear14 =  findViewById(R.id.linear14);
      linearsign =  findViewById(R.id.linearsign);
      imageview =  findViewById(R.id.imageview);
      vscroll1 =  findViewById(R.id.vscroll1);
      buttonsignup = (Button) findViewById(R.id.buttonsignup);
      linear12 =  findViewById(R.id.linear12);
      linear15 =  findViewById(R.id.linear15);
      textview5 = (TextView) findViewById(R.id.textview5);
      edittext21 = (EditText) findViewById(R.id.edittext21);
      textview9 = (TextView) findViewById(R.id.textview9);
      linear_name =  findViewById(R.id.linear_name);
      textview6 = (TextView) findViewById(R.id.textview6);
      lin_email =  findViewById(R.id.lin_email);
      textview10 = (TextView) findViewById(R.id.textview10);
      linr_age =  findViewById(R.id.linr_age);
      textview11 = (TextView) findViewById(R.id.textview11);
      linr_height =  findViewById(R.id.linr_height);
      textview12 = (TextView) findViewById(R.id.textview12);
      linr_weight =  findViewById(R.id.linr_weight);
      txt_gender = (TextView) findViewById(R.id.txt_gender);
      linr_gender =  findViewById(R.id.linr_gender);
      textview7 = (TextView) findViewById(R.id.textview7);
      lin_pass =  findViewById(R.id.lin_pass);
      up_name = (EditText) findViewById(R.id.up_name);
      up_email = (EditText) findViewById(R.id.up_email);
      edittext2 = (EditText) findViewById(R.id.edittext2);
      imgAge =  findViewById(R.id.imgAge);
      up_height = (EditText) findViewById(R.id.up_height);
      up_weight = (EditText) findViewById(R.id.up_weight);
      spin_gender = (Spinner) findViewById(R.id.spin_gender);
      edittext4 = (EditText) findViewById(R.id.edittext4);
      up_pass = (EditText) findViewById(R.id.up_pass);
      imageview_up =  findViewById(R.id.imageview_up);
      textview8 =  findViewById(R.id.textview8);
      signUp =  findViewById(R.id.signUp);
      auth = FirebaseAuth.getInstance();
      name = getSharedPreferences("name", Activity.MODE_PRIVATE);


      //This Activity registers new user's account or logs in old user

      //Button for logging in old user with their email and password
      button_login.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            if (in_email.getText().toString().equals("")) {
               ToastUtil.showMessage(getApplicationContext(), "Email is not
entered");
            }
            else {
               if (in_pass.getText().toString().equals("")) {
                  ToastUtil.showMessage(getApplicationContext(), "Password
is not entered");
```

```
                }
                else {

auth.signInWithEmailAndPassword(in_email.getText().toString(),

in_pass.getText().toString()).addOnCompleteListener(CreateaccountActivity.t
his, authsign_in_listener);
                    CoreProgressLoading(true);
                }
            }
        }
    });

    //Show or hide password
    imgViewPassword.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (num == 1) {

in_pass.setTransformationMethod(android.text.method.HideReturnsTransformati
onMethod.getInstance());

imgViewPassword.setImageResource(R.drawable.ic_visibility_off_black);
                num--;
            }
            else {
                if (num == 0) {

in_pass.setTransformationMethod(android.text.method.PasswordTransformationM
ethod.getInstance());

imgViewPassword.setImageResource(R.drawable.ic_remove_red_eye_black);
                }
                num++;
            }
        }
    });

    //Logging in page becomes invisible and sign up page becomes visible
    //linearsign_bg is a fragment for signing up
    //linear_log_bg is a fragment for logging in
    signup.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            linearsign_bg.setVisibility(View.VISIBLE);
            linear_log_bg.setVisibility(View.GONE);
        }
    });


    //Button for signing up new user
    buttonsignup.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (up_name.getText().toString().equals("")) {
                redStroke(linear_name, 25, "ffffff");
                ToastUtil.showMessage(getApplicationContext(), "Username is
empty");
            }
            else {
                if (up_name.getText().toString().length() < 4) {
```

```
                    ToastUtil.showMessage(getApplicationContext(), "Username
should be more than 4 character");
                    redStroke(linear_name, 25, "ffffff");
                }
                else {
                    greenStroke(linear_name, 25, "ffffff");
                    if (up_email.getText().toString().equals("")) {
                        ToastUtil.showMessage(getApplicationContext(), "Email
is in incorrect");
                        redStroke(lin_email, 25, "ffffff");
                    }
                    else {
                        greenStroke(lin_email, 25, "ffffff");
                        if (up_pass.getText().toString().equals("")) {
                            ToastUtil.showMessage(getApplicationContext(),
"Password can't be empty");
                            redStroke(lin_pass, 25, "ffffff");
                        }
                        else {
                            if (up_pass.getText().toString().length() < 4) {
                                ToastUtil.showMessage(getApplicationContext(),
"Password should be stronger");
                                redStroke(lin_pass, 25, "ffffff");
                            }
                            else {
                                greenStroke(lin_pass, 25, "ffffff");
                                if (edittext2.getText().toString().equals("")) {

ToastUtil.showMessage(getApplicationContext(), "Age can't be empty");
                                    redStroke(linr_age, 25, "ffffff");
                                }
                                else {
                                    if ((edittext2.getText().toString().length()
> 2) || (edittext2.getText().toString().length() < 0)) {

ToastUtil.showMessage(getApplicationContext(), "Age should be actual years
for better app experience");
                                        redStroke(linr_age, 25, "ffffff");
                                    }
                                    else {
                                        greenStroke(linr_age, 25, "ffffff");
                                        if
(up_height.getText().toString().equals("")) {

ToastUtil.showMessage(getApplicationContext(), "Height can't be empty");
                                            redStroke(linr_height, 25, "ffffff");
                                        }
                                        else {
                                            if
(up_height.getText().toString().length() < 0) {

ToastUtil.showMessage(getApplicationContext(), "Incorrect information");
                                                redStroke(linr_height, 25,
"ffffff");
                                            }
                                            else {
                                                greenStroke(linr_height, 25,
"ffffff");
                                                if
(up_weight.getText().toString().equals("")) {
```

```
ToastUtil.showMessage(getApplicationContext(), "Weight can't be empty");
                                            redStroke(linr_weight, 25,
"ffffff");
                                        }
                                        else {
                                            if
(Double.parseDouble(up_weight.getText().toString()) == 0) {

ToastUtil.showMessage(getApplicationContext(), "Weight should be intact for
better app experience");
                                                redStroke(linr_weight, 25,
"ffffff");
                                            }
                                            else {
                                                greenStroke(linr_weight, 25,
"ffffff");
                                                if
(txt_gender.getText().toString().equals("Gender")) {

ToastUtil.showMessage(getApplicationContext(), "Gender not selected");
                                                    redStroke(linr_gender, 25,
"ffffff");
                                                }
                                                else {

                                                    /*Firebase auth register
new user i.e creates new account with entered email and password*/
                                                    greenStroke(linr_gender,
25, "ffffff");
                                                    CoreProgressLoading(true);

auth.createUserWithEmailAndPassword(up_email.getText().toString().trim(),

up_pass.getText().toString().trim()).addOnCompleteListener(CreateaccountAct
ivity.this,

auth_create_user_listener);
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    });

    edittext2.addTextChangedListener(new TextWatcher() {
        @Override
        public void onTextChanged(CharSequence param1, int param2, int
param3, int param4) {
            final String charSeq = param1.toString();
            if (charSeq.length() > 2) {

edittext2.setText(edittext2.getText().toString().substring((int)(0),
(int)(2)));
```

```
            }
        }

        @Override
        public void beforeTextChanged(CharSequence param1, int param2, int
param3, int param4) {

        }

        @Override
        public void afterTextChanged(Editable param1) {

        }
    });

    //Show date picker dialog for choosing age
    imgAge.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            RippleEffects("#35F1B6", imgAge);
            showDatePickerDialog(imgAge);
        }
    });

    //Gender spinner
    spin_gender.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> param1, View param2, int
param3, long param4) {
            if (param3 == 0) {
                spin_gender.setSelection(0);
            }
            //The gender is selected to be male
            if (param3 == 1) {
                txt_gender.setText("Male");
                spin_gender.setSelection(1);
            }
            //The gender is selected to be female
            if (param3 == 2) {
                txt_gender.setText("Female");
                spin_gender.setSelection(2);
            }
        }

        @Override
        public void onNothingSelected(AdapterView<?> param1) {

        }
    });

    //Hide or show password when clicked
    imageview_up.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (num == 1) {
                //Show password

up_pass.setTransformationMethod(android.text.method.HideReturnsTransformati
onMethod.getInstance());
```

```
imageview_up.setImageResource(R.drawable.ic_visibility_off_black);
                num--;
            }
            else {

                //Hide password
                if (num == 0) {

up_pass.setTransformationMethod(android.text.method.PasswordTransformationM
ethod.getInstance());

imageview_up.setImageResource(R.drawable.ic_remove_red_eye_black);
                }
                num++;
            }
        }
    });

    //Sign up linear is visible
    signUp.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            linear_log_bg.setVisibility(View.VISIBLE);
            linearsign_bg.setVisibility(View.GONE);
        }
    });


    sa_db_child_listener = new ChildEventListener() {
        @Override
        public void onChildAdded(@NonNull DataSnapshot param1, String
param2) {
            GenericTypeIndicator<HashMap<String, Object>> ind = new
GenericTypeIndicator<HashMap<String, Object>>() {};
            final String childKey = param1.getKey();
            final HashMap<String, Object> childValue =
param1.getValue(ind);
            try{

                if ((FirebaseAuth.getInstance().getCurrentUser() != null)) {
                    if
(FirebaseAuth.getInstance().getCurrentUser().getUid().equals(childValue.get
("uid").toString())) {
                        name1 = childValue.get("username").toString();
                        name.edit().putString("name", name1).commit();
                    }
                }
            }
            catch(Exception _e){
                //do something if error occurs
                //To know error use _e.toString()

            }
        }

        @Override
        public void onChildChanged(@NonNull DataSnapshot param1, String
param2) {
            GenericTypeIndicator<HashMap<String, Object>> ind = new
GenericTypeIndicator<HashMap<String, Object>>() {};
            final String childKey = param1.getKey();
```

```
            final HashMap<String, Object> childValue =
param1.getValue(ind);

         }

         @Override
         public void onChildMoved(@NonNull DataSnapshot param1, String
param2) {

         }

         @Override
         public void onChildRemoved(@NonNull DataSnapshot param1) {
            GenericTypeIndicator<HashMap<String, Object>> ind = new
GenericTypeIndicator<HashMap<String, Object>>() {};
            final String childKey = param1.getKey();
            final HashMap<String, Object> childValue =
param1.getValue(ind);

         }

         @Override
         public void onCancelled(@NonNull DatabaseError param1) {
            final int _errorCode = param1.getCode();
            final String errorMessage = param1.getMessage();

         }
      };
      sa_db.addChildEventListener(sa_db_child_listener);

      auth_updateEmailListener = new OnCompleteListener<Void>() {
         @Override
         public void onComplete(@NonNull Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };

      auth_updatePasswordListener = new OnCompleteListener<Void>() {
         @Override
         public void onComplete(@NonNull Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };

      auth_emailVerificationSentListener = new OnCompleteListener<Void>() {
         @Override
         public void onComplete(@NonNull Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };

      auth_deleteUserListener = new OnCompleteListener<Void>() {
```

```
        @Override
        public void onComplete(@NonNull Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

        }
    };

    auth_phoneAuthListener = new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task){
            final boolean success = task.isSuccessful();
            final String errorMessage = task.getException() != null ?
task.getException().getMessage() : "";

        }
    };

    auth_updateProfileListener = new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

        }
    };

    auth_googleSignInListener = new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task){
            final boolean success = task.isSuccessful();
            final String errorMessage = task.getException() != null ?
task.getException().getMessage() : "";

        }
    };

    //Takes user's detail on firebase completeListener while registering

    auth_create_user_listener = new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";
            //Note: the uid generated for the particular user is sent to
the firebase database
            //As this will be useful in fetching user's details and editing
without having error
            if (success) {

                //Create a new HashMap to save the data inputted
                //Set a key for each data such as "email"
                sa_map = new HashMap<>();
                sa_map.put("username", up_name.getText().toString());
                sa_map.put("email",
FirebaseAuth.getInstance().getCurrentUser().getEmail());
                sa_map.put("uid",
FirebaseAuth.getInstance().getCurrentUser().getUid());
```

```
                sa_map.put("age", edittext2.getText().toString());
                sa_map.put("height", up_height.getText().toString());
                sa_map.put("weight", up_weight.getText().toString());
                sa_map.put("gender", txt_gender.getText().toString());
                //Create a database and upload user's detail

sa_db.child(FirebaseAuth.getInstance().getCurrentUser().getUid()).updateChi
ldren(sa_map);
                //Sign in the user after registration is complete with email
and password

auth.signInWithEmailAndPassword(up_email.getText().toString().trim(),

up_pass.getText().toString().trim()).addOnCompleteListener(CreateaccountAct
ivity.this, authsign_in_listener);
                CoreProgressLoading(false);
                ToastUtil.showMessage(getApplicationContext(), "Created
successfully");

                //Sends email verification for authentication

FirebaseAuth.getInstance().getCurrentUser().sendEmailVerification().addOnCo
mpleteListener(auth_emailVerificationSentListener);
            }
            else {
                if (!success) {
                    CoreProgressLoading(false);
                    ToastUtil.showMessage(getApplicationContext(),
errorMessage);
                }
            }
        }
    };

    //On sign in user firebase complete
    authsign_in_listener = new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";
            if (success) {
                sa_db.addChildEventListener(sa_db_child_listener);
                ToastUtil.showMessage(getApplicationContext(), "Sign in
successful");
                //Save the username to sharedPreference so as to appear on
mainActivity without internet connection
                name.edit().putString("name", "").commit();
                sa_int.setClass(getApplicationContext(),
MainActivity.class);
                startActivity(sa_int);
                CoreProgressLoading(false);
                finish();
            }
            else {
                if (!success) {
                    ToastUtil.showMessage(getApplicationContext(),
errorMessage);
                    CoreProgressLoading(false);
                }
            }
```

```
            }
        };

        _auth_reset_password_listener = new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> param1) {
                final boolean success = param1.isSuccessful();

            }
        };
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);
        switch (requestCode) {

            default:
            break;
        }
    }

    public void ICC(final ImageView img, final String c1, final String c2) {
        img.setImageTintList(new android.content.res.ColorStateList(new
int[][] {
                {
            -android.R.attr.state_pressed},
                {android.R.attr.state_pressed}},new int[]{Color.parseColor(c1),
Color.parseColor(c2)}));
    }


    public void round(final View viewRound, final double viewRoundSetRadius)
{
        // Gradient Drawable
        android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();

        // Color
        android.graphics.drawable.ColorDrawable cd = new
android.graphics.drawable.ColorDrawable();

        cd =
(android.graphics.drawable.ColorDrawable)viewRound.getBackground();

        int colorId = cd.getColor();
        gd.setColor(colorId);
        gd.setCornerRadius((int)viewRoundSetRadius);
        viewRound.setBackground(gd);
    }


    public void darkIcons() {

getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_
STATUS_BAR);
    }


    public void changeActivityFont(final String fontname) {
```

```
      fontName = "fonts/".concat(fontname.concat(".ttf"));
      overrideFonts(this,getWindow().getDecorView());
   }
   @SuppressLint("WrongConstant")
   private void overrideFonts(final android.content.Context context, final
View v) {

      int style = 0;

      try {
         Typeface
         typeace = Typeface.createFromAsset(getAssets(), fontName);;
         if ((v instanceof ViewGroup)) {
            ViewGroup vg = (ViewGroup) v;
            for (int i = 0;
            i < vg.getChildCount();
            i++) {
               View child = vg.getChildAt(i);
               overrideFonts(context, child);
            }
         }
         else {
            if ((v instanceof TextView)) {

               if (((TextView)
v).getTypeface().getStyle()==Typeface.NORMAL) {

                  style = 0;

               }else{

                  if (((TextView)
v).getTypeface().getStyle()==Typeface.BOLD) {

                     style = 1;

                  }else{

                     if (((TextView)
v).getTypeface().getStyle()==Typeface.ITALIC) {

                        style = 2;

                     }else{

                        if (((TextView)
v).getTypeface().getStyle()==Typeface.BOLD_ITALIC) {

                           style = 3;

                        }}}}

               ((TextView) v).setTypeface(typeace, (style));

            }
            else {
               if ((v instanceof EditText )) {
                  if (((EditText)
v).getTypeface().getStyle()==Typeface.NORMAL) {

                     style = 0;
```

```
                    }else{

                        if (((EditText)
v).getTypeface().getStyle()==Typeface.BOLD) {

                            style = 1;

                        }else{

                            if (((EditText)
v).getTypeface().getStyle()==Typeface.ITALIC) {

                                style = 2;

                            }else{

                                if (((EditText)
v).getTypeface().getStyle()==Typeface.BOLD_ITALIC) {

                                    style = 3;

                                }}}}

                        ((EditText) v).setTypeface(typeace, (style));
                    }
                    else {
                        if ((v instanceof Button)) {
                            if (((Button)
v).getTypeface().getStyle()==Typeface.NORMAL) {

                                style = 0;

                            }else{

                                if (((Button)
v).getTypeface().getStyle()==Typeface.BOLD) {

                                    style = 1;

                                }else{

                                    if (((Button)
v).getTypeface().getStyle()==Typeface.ITALIC) {

                                        style = 2;

                                    }else{

                                        if (((Button)
v).getTypeface().getStyle()==Typeface.BOLD_ITALIC) {

                                            style = 3;

                                        }}}}

                            ((Button) v).setTypeface(typeace, (style));
                        }
                    }
                }
            }
```

```
      }
      catch(Exception e)

      {

      };
   }


   public void greenStroke(final View view, final double radius, final
String color) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
      gd.setColor(Color.parseColor("#" + color.replace("#", ""))); /* color
*/
      gd.setCornerRadius((int)radius); /* radius */
      gd.setStroke(1, Color.GREEN); /* stroke heigth and color */
      view.setBackground(gd);
   }


   public void redStroke(final View view, final double radius, final String
color) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
      gd.setColor(Color.parseColor("#" + color.replace("#", ""))); /* color
*/
      gd.setCornerRadius((int)radius); /* radius */
      gd.setStroke(1, Color.RED); /* stroke heigth and color */
      view.setBackground(gd);

   }


   public void removeScrollBar(final View view) {

      view.setVerticalScrollBarEnabled(false);
      view.setHorizontalScrollBarEnabled(false);
      view.setOverScrollMode(ScrollView.OVER_SCROLL_NEVER);

   }

   //Custom loading progressBar dialog

   public void CoreProgressLoading(final boolean ifShow) {
      if (ifShow) {
         if (coreprog == null){
            coreprog = new ProgressDialog(this);
            coreprog.setCancelable(false);
            coreprog.setCanceledOnTouchOutside(false);

            coreprog.requestWindowFeature(Window.FEATURE_NO_TITLE);
            coreprog.getWindow().setBackgroundDrawable(new
android.graphics.drawable.ColorDrawable(Color.TRANSPARENT));

         }

         android.graphics.drawable.GradientDrawable gads = new
android.graphics.drawable.GradientDrawable();
         gads.setColor(Color.parseColor("#00000000"));
         gads.setCornerRadius(100);
```

```
            coreprog.getWindow().setBackgroundDrawable(gads);
            coreprog.setMessage(null);
            coreprog.show();
            coreprog.setContentView(R.layout.custom_dialog);
        }
        else {
            if (coreprog != null){
               coreprog.dismiss();
            }
        }
    }
    private ProgressDialog coreprog;
    {
    }

    // Define showDatePickerDialog(View).
    public void showDatePickerDialog(View v) {
        // Create and show a new DatePickerFragment.
        DialogFragment newFragment = new DatePickerFragment();
        newFragment.show(getSupportFragmentManager(), "datePicker");
    }

    // Define a DialogFragment class DatePickerFragment.
    public static class DatePickerFragment extends DialogFragment implements
DatePickerDialog.OnDateSetListener {
        // Define a new Calendar for present date
        Calendar now = Calendar.getInstance();
        @Override
        public Dialog onCreateDialog(Bundle savedInstanceState) {
            // Create DatePickerFragment (a DialogFragment) with a new
DatePickerDialog,
            // Present day of month, month, and year are set as the day,
month, and year of this DatePickerDialog.
            int y = now.get(Calendar.YEAR);
            int m = now.get(Calendar.MONTH);
            int d = now.get(Calendar.DAY_OF_MONTH);
            return new DatePickerDialog(getActivity(), this, y, m, d);
        }

        // When Date if birth is selected
        public void onDateSet(DatePicker view, int year, int month, int day)
{
            int mon = month +1;
            // Define a new Calendar for birth date.
            Calendar birthDay = Calendar.getInstance();
            String date = day + "/" + mon + "/" + year;
            // Define the two EditText again using their IDs in main.xml.
            EditText edittext21 = getActivity().findViewById(R.id.edittext1);
            EditText edittext22 = getActivity().findViewById(R.id.edittext2);
            edittext21.setText(date);
            // Set the selected year, month, and day as the year, month, and
day of Calendar birthDay.
            birthDay.set(Calendar.YEAR, year);
            birthDay.set(Calendar.MONTH, month);
            birthDay.set(Calendar.DAY_OF_MONTH, day);
            // find difference between present date and selected date in
milliseconds.
            double diff = (long)(now.getTimeInMillis() -
birthDay.getTimeInMillis());
            // If difference is less than 0, show message that selected date
is in future.
```

```
         if (diff < 0) {
            Toast.makeText(getContext(), "Selected date is in future.",
Toast.LENGTH_SHORT).show();
            edittext22.setText("");
         } else {
            // Get difference between years
            int years = now.get(Calendar.YEAR) -
birthDay.get(Calendar.YEAR);
            int currMonth = now.get(Calendar.MONTH) + 1;
            int birthMonth = birthDay.get(Calendar.MONTH) + 1;
            // Get difference between months
            int months = currMonth - birthMonth;
            // If month difference is negative then reduce years by one
            // and calculate the number of months.
            if (months < 0){
               years--;
               months = 12 - birthMonth + currMonth;
               if (now.get(Calendar.DATE) < birthDay.get(Calendar.DATE))
               months--;
            } else if (months == 0 && now.get(Calendar.DATE) <
birthDay.get(Calendar.DATE)){
               years--;
               months = 11;
            }
            // Calculate the days
            int days = 0;
            if (now.get(Calendar.DATE) > birthDay.get(Calendar.DATE))
            days = now.get(Calendar.DATE) - birthDay.get(Calendar.DATE);
            else if (now.get(Calendar.DATE) < birthDay.get(Calendar.DATE))
            {
               int today = now.get(Calendar.DAY_OF_MONTH);
               now.add(Calendar.MONTH, -1);
               days = now.getActualMaximum(Calendar.DAY_OF_MONTH) -
birthDay.get(Calendar.DAY_OF_MONTH) + today;
            } else {
               days = 0;
               if (months == 12){
                  years++;
                  months = 0;
               }
            }
            // Display the age in years, months and days
            edittext22.setText(years + " /nyears, " + months + " months, "
+ days + " days");
         }
      }
   }


   public void RippleEffects(final String color, final View view) {
      android.content.res.ColorStateList clr = new
android.content.res.ColorStateList(new int[][]{new int[]{}},new
int[]{Color.parseColor(color)});
      android.graphics.drawable.RippleDrawable ripdr = new
android.graphics.drawable.RippleDrawable(clr, null, null);
      view.setBackground(ripdr);
   }


   @Deprecated
   public void showMessage(String s) {
```

```
        Toast.makeText(getApplicationContext(), s,
Toast.LENGTH_SHORT).show();
    }

    @Deprecated
    public int getLocationX(View v) {
        int[] location = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int[] location = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Daily Meal Plan Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import android.graphics.drawable.GradientDrawable;
import android.os.Build;
import android.os.Bundle;
import android.os.Parcelable;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.view.WindowManager;
import android.view.animation.Animation;
import android.view.animation.ScaleAnimation;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.CalendarView;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.blogspot.atifsoftwares.animatoolib.Animatoo;
import com.google.android.material.button.MaterialButton;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import com.sdsmdg.tastytoast.TastyToast;
```

```
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Random;


public class DailymealplanActivity extends AppCompatActivity {

   private double Position = 0;
   private boolean Selected = false;
   private double Selected_Length = 0;
   private boolean IsSelectAll = false;
   private String fontName = "";
   private String typeace = "";
   private String data = "";
   private String Title = "";
   private String Calories = "";
   private HashMap<String, Object> Get_Notes = new HashMap<>();
   private String Get_Note = "";
   private HashMap<String, Object> addmeal = new HashMap<>();
   private String Date = "";
   private String Time = "";
   private boolean breakfast = false;
   private boolean lunch = false;
   private boolean snack = false;
   private boolean dinner = false;
   private double show = 0;
   private String BreakfastServing = "";
   private String LunchServing = "";
   private String SnackServing = "";
   private String DinnerSnack = "";
   private boolean empty = false;
   private double totalkcal = 0;

   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> temp_maplist = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> meal_list = new
ArrayList<>();

   private LinearLayout toolbar;
   private LinearLayout main;
   private ImageView select_all_img;
   private LinearLayout lirnrCalenderBody;
   private TextView textview26;
   private TextView textview27;
   private CalendarView calendarview1;
   private LinearLayout linear42;
   private TextView textview25;
   private Button button1;
   private LinearLayout linrMainMeal;
   private LinearLayout linrAddMeal;
   private ScrollView vscroll1;
   private LinearLayout linear19;
   private LinearLayout linear41;
   private LinearLayout linrBreakfast;
   private LinearLayout linrLunch;
   private LinearLayout linrSnack;
```

```
    private LinearLayout linrDinner;
    private MaterialButton btnSave;
    private ImageView imageview13;
    private TextView textview23;
    private TextView txtDate;
    private LinearLayout linear9;
    private LinearLayout linear11;
    private ImageView imgBreakfast;
    private TextView textview7;
    private ImageView imgAddBk;
    private LinearLayout linear13;
    private LinearLayout linear12;
    private ImageView imageview3;
    private TextView txtBreakfastTitle;
    private LinearLayout linear37;
    private TextView txtBreakfastClass;
    private TextView textview28;
    private TextView txtKCal;
    private LinearLayout linear31;
    private LinearLayout linear32;
    private ImageView imgLunch;
    private TextView textview16;
    private ImageView imgAddLn;
    private LinearLayout linear33;
    private LinearLayout linear34;
    private ImageView imageview9;
    private TextView txtLunchTitle;
    private LinearLayout linear38;
    private TextView txtLunchClass;
    private TextView textview29;
    private TextView txtLunchKcal;
    private LinearLayout linear26;
    private LinearLayout linear27;
    private ImageView imgSnack;
    private TextView textview13;
    private ImageView imgAddSk;
    private LinearLayout linear28;
    private LinearLayout linear29;
    private ImageView imageview7;
    private TextView txtSnackTitle;
    private LinearLayout linear39;
    private TextView txtSnckClass;
    private TextView textview30;
    private TextView txtSnackKcal;
    private LinearLayout linear21;
    private LinearLayout linear22;
    private ImageView imgDinner;
    private TextView textview10;
    private ImageView imgAddDn;
    private LinearLayout linear23;
    private LinearLayout linear24;
    private ImageView imageview5;
    private TextView txtDinnerTitle;
    private LinearLayout linear40;
    private TextView txtDinnerClass;
    private TextView textview31;
    private TextView txtDinnerKcal;
    private LinearLayout linear36;
    private LinearLayout linrClose;
    private TextView txtMealReceiver;
    private LinearLayout linrSearch;
```

```
    private ListView notes_list;
    private TextView textview22;
    private ImageView imageview10;
    private ImageView imageview11;
    private TextView textview21;

    private Intent To_Noteview = new Intent();
    private SharedPreferences Settings;
    private SharedPreferences AllMeals;
    private AlertDialog.Builder Delete_Dialog;
    private SharedPreferences meals;
    private Calendar Cal = Calendar.getInstance();
    private AlertDialog.Builder save;
    private SharedPreferences today;
    private AlertDialog.Builder alert;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.dailymealplan);
        initialize(savedInstanceState);
        com.google.firebase.FirebaseApp.initializeApp(this);
        initializeLogic();
    }

    private void initialize(Bundle savedInstanceState) {
        toolbar =  findViewById(R.id.toolbar);
        main =  findViewById(R.id.main);
        select_all_img =  findViewById(R.id.select_all_img);
        lirnrCalenderBody =  findViewById(R.id.lirnrCalenderBody);
        textview26 = (TextView) findViewById(R.id.textview26);
        textview27 = (TextView) findViewById(R.id.textview27);
        calendarview1 = (CalendarView) findViewById(R.id.calendarview1);
        linear42 =  findViewById(R.id.linear42);
        textview25 = (TextView) findViewById(R.id.textview25);
        button1 = (Button) findViewById(R.id.button1);
        linrMainMeal =  findViewById(R.id.linrMainMeal);
        linrAddMeal =  findViewById(R.id.linrAddMeal);
        vscroll1 =  findViewById(R.id.vscroll1);
        linear19 =  findViewById(R.id.linear19);
        linear41 =  findViewById(R.id.linear41);
        linrBreakfast =  findViewById(R.id.linrBreakfast);
        linrLunch =  findViewById(R.id.linrLunch);
        linrSnack =  findViewById(R.id.linrSnack);
        linrDinner =  findViewById(R.id.linrDinner);
        btnSave = (MaterialButton) findViewById(R.id.btnSave);
        imageview13 =  findViewById(R.id.imageview13);
        textview23 = (TextView) findViewById(R.id.textview23);
        txtDate = (TextView) findViewById(R.id.txtDate);
        linear9 =  findViewById(R.id.linear9);
        linear11 =  findViewById(R.id.linear11);
        imgBreakfast =  findViewById(R.id.imgBreakfast);
        textview7 = (TextView) findViewById(R.id.textview7);
        imgAddBk =  findViewById(R.id.imgAddBk);
        linear13 =  findViewById(R.id.linear13);
        linear12 =  findViewById(R.id.linear12);
        imageview3 =  findViewById(R.id.imageview3);
        txtBreakfastTitle = (TextView) findViewById(R.id.txtBreakfastTitle);
        linear37 =  findViewById(R.id.linear37);
        txtBreakfastClass = (TextView) findViewById(R.id.txtBreakfastClass);
        textview28 = (TextView) findViewById(R.id.textview28);
```

```
        txtKCal = (TextView) findViewById(R.id.txtKCal);
        linear31 =  findViewById(R.id.linear31);
        linear32 =  findViewById(R.id.linear32);
        imgLunch =  findViewById(R.id.imgLunch);
        textview16 = (TextView) findViewById(R.id.textview16);
        imgAddLn =  findViewById(R.id.imgAddLn);
        linear33 =  findViewById(R.id.linear33);
        linear34 =  findViewById(R.id.linear34);
        imageview9 =  findViewById(R.id.imageview9);
        txtLunchTitle = (TextView) findViewById(R.id.txtLunchTitle);
        linear38 =  findViewById(R.id.linear38);
        txtLunchClass = (TextView) findViewById(R.id.txtLunchClass);
        textview29 = (TextView) findViewById(R.id.textview29);
        txtLunchKcal = (TextView) findViewById(R.id.txtLunchKcal);
        linear26 =  findViewById(R.id.linear26);
        linear27 =  findViewById(R.id.linear27);
        imgSnack =  findViewById(R.id.imgSnack);
        textview13 = (TextView) findViewById(R.id.textview13);
        imgAddSk =  findViewById(R.id.imgAddSk);
        linear28 =  findViewById(R.id.linear28);
        linear29 =  findViewById(R.id.linear29);
        imageview7 =  findViewById(R.id.imageview7);
        txtSnackTitle = (TextView) findViewById(R.id.txtSnackTitle);
        linear39 =  findViewById(R.id.linear39);
        txtSnckClass = (TextView) findViewById(R.id.txtSnckClass);
        textview30 = (TextView) findViewById(R.id.textview30);
        txtSnackKcal = (TextView) findViewById(R.id.txtSnackKcal);
        linear21 =  findViewById(R.id.linear21);
        linear22 =  findViewById(R.id.linear22);
        imgDinner =  findViewById(R.id.imgDinner);
        textview10 = (TextView) findViewById(R.id.textview10);
        imgAddDn =  findViewById(R.id.imgAddDn);
        linear23 =  findViewById(R.id.linear23);
        linear24 =  findViewById(R.id.linear24);
        imageview5 =  findViewById(R.id.imageview5);
        txtDinnerTitle = (TextView) findViewById(R.id.txtDinnerTitle);
        linear40 =  findViewById(R.id.linear40);
        txtDinnerClass = (TextView) findViewById(R.id.txtDinnerClass);
        textview31 = (TextView) findViewById(R.id.textview31);
        txtDinnerKcal = (TextView) findViewById(R.id.txtDinnerKcal);
        linear36 =  findViewById(R.id.linear36);
        linrClose =  findViewById(R.id.linrClose);
        txtMealReceiver = (TextView) findViewById(R.id.txtMealReceiver);
        linrSearch =  findViewById(R.id.linrSearch);
        notes_list = (ListView) findViewById(R.id.notes_list);
        textview22 = (TextView) findViewById(R.id.textview22);
        imageview10 =  findViewById(R.id.imageview10);
        imageview11 =  findViewById(R.id.imageview11);
        textview21 = (TextView) findViewById(R.id.textview21);
        Settings = getSharedPreferences("Settings", Activity.MODE_PRIVATE);
        AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);
        Delete_Dialog = new AlertDialog.Builder(this);
        meals = getSharedPreferences("meals", Activity.MODE_PRIVATE);
        save = new AlertDialog.Builder(this);
        today = getSharedPreferences("today", Activity.MODE_PRIVATE);
        alert = new AlertDialog.Builder(this);


        //Get the current date and check if the user changes the date
        calendarview1.setOnDateChangeListener(new
CalendarView.OnDateChangeListener() {
```

```
        @Override
        public void onSelectedDayChange(CalendarView param1, int param2,
int param3, int param4) {
            final int year = param2;
            final int month = param3;
            final int day = param4;
            //if current date is equal to the day selected by the user
            //create a sharedPreferences with key "today" and value "true"
            //After the user has created a meal plan for the current day
            //Once the user comes back to create a meal plan for the
current day
            //Error dialog will appear
            if (day == Double.parseDouble(new
SimpleDateFormat("dd").format(Cal.getTime()))) {
               today.edit().putString("Today", "True").commit();
               txtDate.setText("Today");
            }
            else {
               txtDate.setText(String.valueOf((long)(day)));
               today.edit().putString("Today", "False").commit();
               Date = String.valueOf((long)(day));
            }
         }
      });

      //Show or hide calender
      button1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            show++;
            //Calender is visible
            if (show == 1) {
               calendarview1.setVisibility(View.VISIBLE);
               button1.setText("Minimize");
            }
            else {
               //Calender is gone
               if (show == 2) {
                  calendarview1.setVisibility(View.GONE);
                  button1.setText("Open");
                  show = 0;
               }
            }
         }
      });

      //Add breakfast by selecting from the meal list item

      linrBreakfast.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            breakfast = true;
            txtMealReceiver.setText("Add Breakfast");
            linrMainMeal.setVisibility(View.GONE);
            linrAddMeal.setVisibility(View.VISIBLE);

            imgBreakfast.setImageResource(R.drawable.circle_slice);
         }
      });
```

```
      //Add lunch by selecting from the meal list item

      linrLunch.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            txtMealReceiver.setText("Add Lunch");
            linrMainMeal.setVisibility(View.GONE);
            linrAddMeal.setVisibility(View.VISIBLE);

            imgLunch.setImageResource(R.drawable.circle_slice);
            lunch = true;
         }
      });

      //Add snack by selecting from the meal list item

      linrSnack.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            txtMealReceiver.setText("Add Snack");
            linrMainMeal.setVisibility(View.GONE);
            linrAddMeal.setVisibility(View.VISIBLE);

            imgSnack.setImageResource(R.drawable.circle_slice);
            snack = true;
         }
      });

      //Add dinner by selecting from the meal list item

      linrDinner.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            txtMealReceiver.setText("Add Dinner");
            linrMainMeal.setVisibility(View.GONE);
            linrAddMeal.setVisibility(View.VISIBLE);

            imgDinner.setImageResource(R.drawable.circle_slice);
            dinner = true;
         }
      });

      //onClick to save the entered meal plan by the user
      btnSave.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            //Get the current date
            Cal = Calendar.getInstance();
            if (empty) {
               //check if the current date == date selected for creating
meal via the calenderView
               //This is done to avoid creating multiple plan for the same
day
               if (today.getString("Today", "").equals("False")) {
                  if (linrMainMeal.getVisibility() == View.VISIBLE) {
                     totalkcal =
(Double.parseDouble(txtKCal.getText().toString())
                           +
Double.parseDouble(txtKCal.getText().toString()))
                           +
(Double.parseDouble(txtKCal.getText().toString())
```

```
                            +
Double.parseDouble(txtKCal.getText().toString()));

                    //Create a new HashMap to save the data collected
                    //Set a key for each data such "Lunch" for input of
ingredients entered by user
                    addmeal = new HashMap<>();
                    addmeal.put("Breakfast",
txtBreakfastTitle.getText().toString());
                    addmeal.put("Breakfastclass",
txtBreakfastClass.getText().toString());
                    addmeal.put("Breakfastkcal",
txtKCal.getText().toString());
                    addmeal.put("Lunch",
txtLunchTitle.getText().toString());
                    addmeal.put("Lunchclass",
txtLunchClass.getText().toString());
                    addmeal.put("Lunchkkcal",
txtLunchKcal.getText().toString());
                    addmeal.put("Snack",
txtSnackTitle.getText().toString());
                    addmeal.put("Snackclass",
txtSnckClass.getText().toString());
                    addmeal.put("Snackkcal",
txtSnackKcal.getText().toString());
                    addmeal.put("Dinner",
txtDinnerTitle.getText().toString());
                    addmeal.put("Dinnerclass",
txtDinnerClass.getText().toString());
                    addmeal.put("Dinnerkcal",
txtDinnerKcal.getText().toString());
                    addmeal.put("Date", Date.concat(" ".concat(new
SimpleDateFormat("MMM").format(Cal.getTime()))));
                    addmeal.put("Time", Time);
                    addmeal.put("Breakfastserv", BreakfastServing);
                    addmeal.put("Lunchserv", LunchServing);
                    addmeal.put("Snackserv", SnackServing);
                    addmeal.put("Dinnerserv", DinnerSnack);
                    addmeal.put("Totalkcal",
String.valueOf((long)(totalkcal)));

                    //Create an ArrayList HashMap to save all the data
from the HashMap created

                    meal_list.add(addmeal);

                    //Create a sharedPreference, note that the ArrayList
HashMap is converted to Json
                    //Save the data into the sharedPreference
                    meals.edit().putString("Meals", new
Gson().toJson(meal_list)).commit();
                    TastyToast.makeText(getApplicationContext(), "Your
Meal Have Been Successfully Added ", TastyToast.LENGTH_LONG,
TastyToast.SUCCESS);
                    if (txtDate.getText().toString().equals(new
SimpleDateFormat("dd").format(Cal.getTime()))) {
                        today.edit().putString("Today", "True").commit();
                    }
                    else {
                        today.edit().putString("Today", "False").commit();
                    }
```

```
                    finish();
                }
                else {
                    finish();
                }
            }
            else {

                //this is where the error dialog is shown to the user
                alert.setTitle("Hmm.. Can't Create Plan!");
                alert.setMessage("Sorry, you cannot create different meal
plan for a single day. Try editing the one created first or change date");
                alert.setPositiveButton("Ok", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which)
{

                    }
                });
                alert.create().show();
            }
        }
        else {
            ToastUtil.showMessage(getApplicationContext(), "Can't save
empty meal plan");
        }
    }
});

linrClose.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        linrMainMeal.setVisibility(View.VISIBLE);
        linrAddMeal.setVisibility(View.GONE);
        clickAnimation(linrClose);
    }
});

linrSearch.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent(DailymealplanActivity.this,
SearchscanmealActivity.class));
Animatoo.animateFade(DailymealplanActivity.this);
    }
});

/* listView Adapter for the meal created by the user*/
notes_list.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> param1, View param2, int
param3, long param4) {
        final int position = param3;
        empty = true;
        if (Selected) {

            //Selected boolean is to determine if the listItem is
clicked to checked for deleting
            //IsSelectALl will select all the listItem in the when its
```

```
true

                if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Select").toString().equals("True")) {
                   Notes_List.get((int)(Notes_List.size() - 1) -
position).put("Select", "False");
                   Selected_Length--;
                   IsSelectAll = false;
                }
                else {
                   Notes_List.get((int)(Notes_List.size() - 1) -
position).put("Select", "True");
                   Selected_Length++;
                }
                refresh(Notes_List);
                _Toolbar(Selected_Length);
                if (Selected_Length == 0) {
                   IsSelectAll = false;


                   Selected = false;

((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
                }
             }
             else {
                Get_Notes = Notes_List.get((int)(Notes_List.size() - 1) -
position);
                linrMainMeal.setVisibility(View.VISIBLE);
                linrAddMeal.setVisibility(View.GONE);

                /* Adding a breakfast is by clicking the breakfast
                linearLayout which will make the boolean "breakfast" true,
initially the boolean is
                false. When a meal is clicked on the listView, linearLayout
breakfast children element
                  will receive the text from the listItem,
                   then the breakfast boolean will be false*/
                if (breakfast) {
                   imgAddBk.setVisibility(View.GONE);

txtBreakfastTitle.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString());

txtBreakfastClass.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Class").toString());
                   txtKCal.setText(Notes_List.get((int)(Notes_List.size() -
1) - position).get("Calories").toString());
                   BreakfastServing = Notes_List.get((int)(Notes_List.size()
- 1) - position).get("Serving").toString();
                   breakfast = false;
                }
                else {
                      /* Adding a lunch is by clicking the lunch
                linearLayout which will make the boolean "lunch" true,
initially the boolean is
                false. When a meal is clicked on the listView, linearLayout
lunch children element
                  will receive the text from the listItem,
                   then the lunch boolean will be false*/
                   if (lunch) {
```

```
txtLunchTitle.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString());

txtLunchClass.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Class").toString());

txtLunchKcal.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Calories").toString());
                    lunch = false;
                    SnackServing = Notes_List.get((int)(Notes_List.size()
- 1) - position).get("Serving").toString();
                    imgAddLn.setVisibility(View.GONE);
                 }
                 else {
                       /* Adding a snack is by clicking the snack
              linearLayout which will make the boolean "snack" true,
initially the boolean is
              false. When a meal is clicked on the listView, linearLayout
snack children element
                will receive the text from the listItem,
                 then the snack boolean will be false*/
                    if (snack) {

txtSnackTitle.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString());

txtSnckClass.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Class").toString());

txtSnackKcal.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Calories").toString());
                       snack = false;
                       SnackServing =
Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Serving").toString();
                       imgAddSk.setVisibility(View.GONE);
                    }
                    else {
                          /* Adding a dinner is by clicking the dinner
              linearLayout which will make the boolean "dinner" true,
initially the boolean is
              false. When a meal is clicked on the listView, linearLayout
dinner children element
                will receive the text from the listItem,
                 then the dinner boolean will be false*/
                       if (dinner) {
                          imgAddDn.setVisibility(View.GONE);

txtDinnerTitle.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString());

txtDinnerClass.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Class").toString());

txtDinnerKcal.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Calories").toString());
                          dinner = false;
                          DinnerSnack =
Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Serving").toString();
```

```
                        }
                    }
                }
            }
        }

        //check if the meal listitem size is empty
        if (Notes_List.size() == 0) {

com.google.android.material.snackbar.Snackbar.make(linrAddMeal, "No meal is
found to add",
com.google.android.material.snackbar.Snackbar.LENGTH_SHORT).setAction("Crea
te Meal", new View.OnClickListener(){
                @Override
                public void onClick(View view) {
                    startActivity(new Intent(DailymealplanActivity.this,
AddMealActivity.class));
                    Animatoo.animateFade(DailymealplanActivity.this);
                }
            }).show();
        }
        else {

        }
      }
    });
  }

  private void initializeLogic() {
//    Set the meals plans to false
//    This is helpful such that you can determine the meal plan
//    user is about enter
    breakfast = false;
    lunch = false;
    snack = false;
    dinner = false;
    show = 0;
    calendarview1.setVisibility(View.GONE);

//    Designing some interface
    linrClose.setBackground(new GradientDrawable() {
        public GradientDrawable getIns(int a, int b) {
            this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)60, 0xFFFFFFFF));
    notes_list.setBackground(new GradientDrawable() {

        public GradientDrawable getIns(int a, int b) {
            this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)30, 0xFFFFFFFF));
    linrSearch.setBackground(new GradientDrawable() {

        public GradientDrawable getIns(int a, int b) {
            this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)30, 0xFFFFFFFF));
    linrBreakfast.setBackground(new GradientDrawable()
    {
        public GradientDrawable getIns(int a, int b) {
            this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)15, 0xFFFFFFFF));
    linrLunch.setBackground(new GradientDrawable() {
```

```
         public GradientDrawable getIns(int a, int b) {
            this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)15, 0xFFFFFFFF));
      linrSnack.setBackground(new GradientDrawable() {

         public GradientDrawable getIns(int a, int b) {
            this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)15, 0xFFFFFFFF));
      linrDinner.setBackground(new GradientDrawable() {

         public GradientDrawable getIns(int a, int b) {
            this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)15, 0xFFFFFFFF));
      btnSave.setBackground(new GradientDrawable() {

         public GradientDrawable getIns(int a, int b) {
            this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)15, 0xFF3F51B6));
      changeActivityFont("railway_semisolid");
      removeScrollBar(notes_list);
      getSettingsData();
      ONCREATE();


      if (Build.VERSION.SDK_INT > Build.VERSION_CODES.KITKAT) {
         Window w =DailymealplanActivity.this.getWindow();
         w.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);

w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
w.setStatusBarColor(0xFFFFFFFF);
      }
      empty = false;
      if (!getIntent().getStringExtra("Data").equals("")) {
         Notes_List = new
Gson().fromJson(getIntent().getStringExtra("Data"), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
         notes_list.setAdapter(new Notes_listAdapter(Notes_List));
         ((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
      }
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }
// onBackPressed
   @Override
   public void onBackPressed() {
      if (Selected) {
         _UnSelectAll();
      }
      else {
         save.setTitle("Warning!!!");
         save.setMessage("Do you want to exit without saving");
         save.setPositiveButton("Yes", new
```

```
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
               finish();
            }
         });
         save.setNegativeButton("No", new DialogInterface.OnClickListener()
{
            @Override
            public void onClick(DialogInterface dialog, int which) {

            }
         });
         save.create().show();
      }
   }

   @Override
   public void onResume() {
      super.onResume();
      refresh(Notes_List);
   }
   public void setRadiusToView (final View view, final double radius, final
String Colour) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
gd.setColor(Color.parseColor(Colour)); gd.setCornerRadius((int)radius);
view.setBackground(gd);
   }


   public void setRadiusToView (final View view, final double value) {
      view.setElevation((float)value);
   }


   public void corners (final View view, final String _color1, final String
_color2, final double msgtr, final double _n1, final double _n2, final
double _n3, final double _n4) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();

      gd.setColor(Color.parseColor(_color1));

      gd.setStroke((int)msgtr, Color.parseColor(_color2));

      gd.setCornerRadii(new
float[]{(int)_n1,(int)_n1,(int)_n2,(int)_n2,(int)_n3,(int)_n3,(int)_n4,(int
)_n4});

      view.setBackground(gd);

      view.setElevation(4);
   }


   public void Add (final String Colour, final ImageView Imageview) {
      Imageview.getDrawable().setColorFilter(Color.parseColor(Colour),
PorterDuff.Mode.SRC_IN);
   }
```

```
   public void removeScrollBar (final View view) {
      view.setVerticalScrollBarEnabled(false);
view.setHorizontalScrollBarEnabled(false);
   }


   public void getSettingsData () {

   }


   public void textSize (final TextView TextView1, final double size) {
      int j = (int) size;
      TextView1.setTextSize(j);
   }


   public void _Toolbar (final double _Length) {

   }


   public void _UnSelectAll () {
      Selected = false;
      Selected_Length = 0;


      IsSelectAll = false;
      Position = 0;
      for(int i = 0; i < (int)(Notes_List.size()); i++) {
         if
(Notes_List.get((int)Position).get("Select").toString().equals("True")) {
            Notes_List.get((int)Position).put("Select", "False");
         }
         Position++;
      }
      refresh(Notes_List);
   }

// Refresh the meal list
   public void refresh (final ArrayList<HashMap<String, Object>> ListMap) {
      Parcelable state =
      notes_list.onSaveInstanceState();
      notes_list.setAdapter(new Notes_listAdapter(ListMap));
      ((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
      notes_list.onRestoreInstanceState(state);
   }


   public void selectAll () {
      if (IsSelectAll) {
         _UnSelectAll();
         IsSelectAll = false;
      }
      else {
         IsSelectAll = true;
         Position = 0;
         for(int i = 0; i < (int)(Notes_List.size()); i++) {
            if
(Notes_List.get((int)Position).get("Select").toString().equals("False")) {
```

```
                Notes_List.get((int)Position).put("Select", "True");
                Selected_Length++;
             }
             Position++;
             _Toolbar(Selected_Length);
          }
          refresh(Notes_List);
       }
    }


    public void saveData () {
       temp_maplist = new Gson().fromJson(AllMeals.getString("Meals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
       if (Notes_List.size() == 0) {
          if (getIntent().getStringExtra("Type").equals("All")) {
             AllMeals.edit().putString("Meals", new
Gson().toJson(Notes_List)).commit();
          }
          else {
             if (getIntent().getStringExtra("Type").equals("Important")) {
                Position = temp_maplist.size() - 1;
                for(int i = 0; i < (int)(temp_maplist.size()); i++) {
                   if
(temp_maplist.get((int)Position).get("Important").toString().equals("True")
) {
                      temp_maplist.remove((int)(Position));
                   }
                   Position--;
                }
                AllMeals.edit().putString("Meals", new
Gson().toJson(temp_maplist)).commit();
             }
          }
       }
       else {
          AllMeals.edit().putString("Meals", new
Gson().toJson(Notes_List)).commit();
       }
    }


    public void changeActivityFont (final String fontname) {
       fontName = "fonts/".concat(fontname.concat(".ttf"));
       overrideFonts(this,getWindow().getDecorView());
    }
    private void overrideFonts(final android.content.Context context, final
View v) {

       try {
          Typeface
          typeace = Typeface.createFromAsset(getAssets(), fontName);;
          if ((v instanceof ViewGroup)) {
             ViewGroup vg = (ViewGroup) v;
             for (int i = 0;
             i < vg.getChildCount();
             i++) {
                View child = vg.getChildAt(i);
                overrideFonts(context, child);
             }
          }
```

```
         else {
            if ((v instanceof TextView)) {
               ((TextView) v).setTypeface(typeace);
            }
            else {
               if ((v instanceof EditText )) {
                  ((EditText) v).setTypeface(typeace);
               }
               else {
                  if ((v instanceof Button)) {
                     ((Button) v).setTypeface(typeace);
                  }
               }
            }
         }
      }
      catch(Exception e)

      {

      };
   }


   public void clickAnimation (final View view) {
      ScaleAnimation fade_in = new ScaleAnimation(0.9f, 1f, 0.9f, 1f,
Animation.RELATIVE_TO_SELF, 0.5f, Animation.RELATIVE_TO_SELF, 0.7f);
      fade_in.setDuration(300);
      fade_in.setFillAfter(true);
      view.startAnimation(fade_in);
   }

// Get the meals list into a listView
   public void ONCREATE () {
      Cal = Calendar.getInstance();
//    IF the meal sharedPreference is not equal empty value then get the
meals into ArrayList
//    from the json
      if (!meals.getString("Meals", "").equals("")) {
         meal_list = new Gson().fromJson(meals.getString("Meals", ""), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      }
      Date = getIntent().getStringExtra("Date");
      Time = new SimpleDateFormat("hh:mm a").format(Cal.getTime());
   }


   public class Notes_listAdapter extends BaseAdapter {
      ArrayList<HashMap<String, Object>> data;
      public Notes_listAdapter(ArrayList<HashMap<String, Object>> arr) {
         data = arr;
      }

      @Override
      public int getCount() {
         return data.size();
      }

      @Override
      public HashMap<String, Object> getItem(int index) {
         return data.get(index);
```

```
      }

      @Override
      public long getItemId(int index) {
         return index;
      }
      @Override
      public View getView(final int position, View v, ViewGroup container)
{
         LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
         View view = v;
         if (view == null) {
            view = inflater.inflate(R.layout.custom_meal_list, null);
         }

         final LinearLayout bg =  view.findViewById(R.id.bg);
         final LinearLayout linear1 =  view.findViewById(R.id.linear1);
         final ImageView imageview1 =  view.findViewById(R.id.imageview1);
         final LinearLayout main =  view.findViewById(R.id.main);
         final LinearLayout side_lin =  view.findViewById(R.id.side_lin);
         final LinearLayout text_lin =  view.findViewById(R.id.text_lin);
         final LinearLayout title_time_lin =
view.findViewById(R.id.title_time_lin);
         final LinearLayout note_lin =  view.findViewById(R.id.note_lin);
         final TextView date = (TextView) view.findViewById(R.id.date);
         final LinearLayout linear2 =  view.findViewById(R.id.linear2);
         final TextView time = (TextView) view.findViewById(R.id.time);
         final TextView title = (TextView) view.findViewById(R.id.title);
         final TextView txtFoodClasses = (TextView)
view.findViewById(R.id.txtFoodClasses);
         final TextView note = (TextView) view.findViewById(R.id.note);
         final TextView txt_methods = (TextView)
view.findViewById(R.id.txt_methods);
         final TextView txt_calories = (TextView)
view.findViewById(R.id.txt_calories);
         final TextView txtminutes = (TextView)
view.findViewById(R.id.txtminutes);
         final TextView textview4 = (TextView)
view.findViewById(R.id.textview4);
         final TextView txtServing = (TextView)
view.findViewById(R.id.txtServing);

         linear1.setVisibility(View.GONE);

date.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
ium.ttf"), Typeface.NORMAL);

title.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansbo
ld.ttf"), Typeface.NORMAL);

time.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
ium.ttf"), Typeface.NORMAL);

note.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
ium.ttf"), Typeface.NORMAL);
         side_lin.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)60, 0xFF3F51B6));
         if (Selected) {
```

```
imageview1.setImageResource(R.drawable.ic_radio_button_off_black);
            if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Select").toString().equals("True")) {
               imageview1.setVisibility(View.VISIBLE);

imageview1.setImageResource(R.drawable.ic_radio_button_on_black);
            }
            else {

imageview1.setImageResource(R.drawable.ic_radio_button_off_black);
            }
         }
         else {
            imageview1.setVisibility(View.GONE);
         }
         if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Title")) {
            if (Settings.getString("Text Size", "").equals("Small")) {
               if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString().length() > 28) {
                  title.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Title").toString().substring((int)(0),
(int)(28)).concat("..."));
               }
               else {
                  title.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Title").toString());
               }
            }
            else {
               if (Settings.getString("Text Size", "").equals("Medium")) {
                  if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString().length() > 24) {
                     title.setText(Notes_List.get((int)(Notes_List.size() -
1) - position).get("Title").toString().substring((int)(0),
(int)(24)).concat("..."));
                  }
                  else {
                     title.setText(Notes_List.get((int)(Notes_List.size() -
1) - position).get("Title").toString());
                  }
               }
               else {
                  if (Settings.getString("Text Size", "").equals("Large"))
{
                     if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString().length() > 22) {

title.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString().substring((int)(0),
(int)(22)).concat("..."));
                     }
                     else {

title.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString());
                     }
                  }
                  else {
```

```
                }
            }
        }
        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Time")) {
            time.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Time").toString());
        }
        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Date")) {
            date.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Date").toString());
        }
        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Method")) {
            note.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Method").toString());
        }
        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Note")) {
            txt_methods.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Note").toString());
        }
        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Calories")) {
            txt_calories.setText(Notes_List.get((int)(Notes_List.size() -
1) - position).get("Calories").toString());
        }
        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Minutes")) {
            txtminutes.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Minutes").toString());
        }
        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Ingredients")) {
            txt_methods.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Ingredients").toString());
        }
        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Class")) {
            txtFoodClasses.setText(Notes_List.get((int)(Notes_List.size() -
1) - position).get("Class").toString());
        }
        note.setVisibility(View.GONE);

        return view;
    }
  }

  @Deprecated
  public void showMessage(String msg) {
     Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
  }

  @Deprecated
  public int getLocationX(View v) {
     int location[] = new int[2];
     v.getLocationInWindow(location);
     return location[0];
```

```
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Debug Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;

public class DebugActivity extends Activity {

    private String[] exceptionTypes = {
            "StringIndexOutOfBoundsException",
            "IndexOutOfBoundsException",
            "ArithmeticException",
            "NumberFormatException",
            "ActivityNotFoundException"
    };

    private String[] exceptionMessages = {
            "Invalid string operation\n",
            "Invalid list operation\n",
            "Invalid arithmetical operation\n",
            "Invalid toNumber block operation\n",
            "Invalid intent operation"
    };


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        Intent intent = getIntent();
        String errorMessage = "";
        String madeErrorMessage = "";

        if (intent != null) {
            errorMessage = intent.getStringExtra("error");

            String[] split = errorMessage.split("\n");
            //errorMessage = split[0];
            try {
                for (int j = 0; j < exceptionTypes.length; j++) {
```

```
                    if (split[0].contains(exceptionTypes[j])) {
                        madeErrorMessage = exceptionMessages[j];

                        int addIndex = split[0].indexOf(exceptionTypes[j])
+ exceptionTypes[j].length();

                        madeErrorMessage += split[0].substring(addIndex,
split[0].length());
                        madeErrorMessage += "\n\nDetailed error message:\n"
+ errorMessage;
                        break;
                    }
                }

                if (madeErrorMessage.isEmpty()) {
                    madeErrorMessage = errorMessage;
                }
            } catch (Exception e) {
                madeErrorMessage = madeErrorMessage + "\n\nError while
getting error: " + Log.getStackTraceString(e);
            }
        }

        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("An error occurred");
        builder.setMessage(madeErrorMessage);
        builder.setPositiveButton("End Application", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                finish();
            }
        });
        builder.create().show();
    }
}
```

Diary List Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;


public class DiarylistActivity extends AppCompatActivity {

   private ArrayList<HashMap<String, Object>> Food_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> Recipe_List = new
ArrayList<>();

   private LinearLayout linear1;
   private LinearLayout linear2;
   private LinearLayout toolbar;
   private ListView listview2;
   private ImageView back;
```

```
   private SharedPreferences Diary;
   private SharedPreferences Recipe;
   private Intent i = new Intent();

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.diarylist);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      linear1 =  findViewById(R.id.linear1);
      linear2 =  findViewById(R.id.linear2);
      toolbar =  findViewById(R.id.toolbar);
      listview2 = (ListView) findViewById(R.id.listview2);
      back =  findViewById(R.id.back);
      Diary = getSharedPreferences("Diary", Activity.MODE_PRIVATE);
      Recipe = getSharedPreferences("Recipe", Activity.MODE_PRIVATE);

//
      listview2.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
         @Override
         public void onItemClick(AdapterView<?> param1, View param2, int
param3, long param4) {
            final int position = param3;
            i.setClass(getApplicationContext(), RecipeviewActivity.class);
            i.putExtra("position",
String.valueOf((long)((Recipe_List.size() - 1) - position)));
            startActivity(i);
         }
      });
   }

   private void initializeLogic() {

      //    IF the recipe sharedPreference is not equal empty value then
get recipe into ArrayList
//    from the json
      if (!Recipe.getString("Recipe", "").equals("")) {
         Recipe_List = new Gson().fromJson(Recipe.getString("Recipe", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
         listview2.setAdapter(new Listview2Adapter(Recipe_List));
         ((BaseAdapter)listview2.getAdapter()).notifyDataSetChanged();
      }
      getDiary();
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }
```

```
   public void getDiary () {

   }

//
   public class Listview2Adapter extends BaseAdapter {
      ArrayList<HashMap<String, Object>> data;
      public Listview2Adapter(ArrayList<HashMap<String, Object>> arr) {
         data = arr;
      }

      @Override
      public int getCount() {
         return data.size();
      }

      @Override
      public HashMap<String, Object> getItem(int index) {
         return data.get(index);
      }

      @Override
      public long getItemId(int index) {
         return index;
      }
      @Override
      public View getView(final int position, View v, ViewGroup container)
{
         LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
         View view = v;
         if (view == null) {

//          inflate the with layout saveddiary
            view = inflater.inflate(R.layout.saveddiary, null);
         }

         //Define your layout elements here
         final LinearLayout linear1 =  view.findViewById(R.id.linear1);
         final LinearLayout linear4 =  view.findViewById(R.id.linear4);
         final LinearLayout linear2 =  view.findViewById(R.id.linear2);
         final LinearLayout linear3 =  view.findViewById(R.id.linear3);
         final ImageView imgFoodImg =  view.findViewById(R.id.imgFoodImg);
         final LinearLayout linear5 =  view.findViewById(R.id.linear5);
         final LinearLayout linear6 =  view.findViewById(R.id.linear6);
         final TextView txtFoodName = (TextView)
view.findViewById(R.id.txtFoodName);
         final TextView txtIngtd = (TextView)
view.findViewById(R.id.txtIngtd);
         final TextView txtDate = (TextView)
view.findViewById(R.id.txtDate);
         final TextView txtCarbo = (TextView)
view.findViewById(R.id.txtCarbo);
         final TextView txtPro = (TextView) view.findViewById(R.id.txtPro);
         final TextView txtKcal = (TextView)
view.findViewById(R.id.txtKcal);
         final TextView txtfat = (TextView) view.findViewById(R.id.txtfat);
         final TextView txtSugar = (TextView)
view.findViewById(R.id.txtSugar);
```

```
         final TextView txtSalt = (TextView)
view.findViewById(R.id.txtSalt);

         linear4.setVisibility(View.GONE);
         //Get the list item details
         if (Recipe_List.get((int)(Recipe_List.size() - 1) -
position).containsKey("Food")) {
            txtFoodName.setText(Recipe_List.get((int)(Recipe_List.size() -
1) - position).get("Food").toString());
         }
         if (Recipe_List.get((int)(Recipe_List.size() - 1) -
position).containsKey("Ingredt")) {
            txtIngtd.setText(Recipe_List.get((int)(Recipe_List.size() - 1)
- position).get("Ingredt").toString());
         }
         if (Recipe_List.get((int)(Recipe_List.size() - 1) -
position).containsKey("Date")) {
            txtDate.setText(Recipe_List.get((int)(Recipe_List.size() - 1) -
position).get("Date").toString());
         }

         return view;
      }
   }

   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }

   @Deprecated
   public int getRandom(int min, int max) {
      Random random = new Random();
      return random.nextInt(max - min + 1) + min;
   }

   @Deprecated
   public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
      ArrayList<Double> result = new ArrayList<Double>();
      SparseBooleanArray arr = _list.getCheckedItemPositions();
      for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
         if (arr.valueAt(iIdx))
         result.add((double)arr.keyAt(iIdx));
      }
      return result;
```

```
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.drawable.GradientDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.bumptech.glide.Glide;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;

import de.hdodenhof.circleimageview.CircleImageView;


public class DiaryviewActivity extends AppCompatActivity {

    private HashMap<String, Object> GetDiary = new HashMap<>();
    private double position = 0;
    private String Food = "";
```

```
   private String Ingredt = "";
   private String kcal = "";
   private String Carbo = "";
   private String Protein = "";
   private String Fat = "";
   private String Sugar = "";
   private String Salt = "";
   private String Date = "";
   private String Time = "";
   private String Img = "";
   private String Consumed = "";
   private double prevprotein = 0;
   private double prevcalories = 0;
   private double preveaten = 0;
   private double prevfat = 0;

   private ArrayList<HashMap<String, Object>> Food_List = new
ArrayList<>();

   private LinearLayout linear1;
   private LinearLayout toolbar;
   private ScrollView vscroll1;
   private Button button1;
   private ImageView back;
   private LinearLayout linear7;
   private LinearLayout linrMain;
   private TextView textview7;
   private LinearLayout linrKcal;
   private TextView textview4;
   private LinearLayout linrCarbo;
   private LinearLayout linrProtn;
   private LinearLayout linrFats;
   private LinearLayout linrSugar;
   private LinearLayout linrSa;
   private CircleImageView circleimageview1;
   private LinearLayout linear3;
   private TextView txtFoodName;
   private TextView textview8;
   private TextView txtkCal;
   private TextView textview5;
   private TextView txtCarbohydate;
   private TextView textview10;
   private TextView txtProtein;
   private TextView textview12;
   private TextView txtFat;
   private TextView textview14;
   private TextView txtSugar;
   private TextView textview16;
   private TextView txtSalt;

   private SharedPreferences Diary;
   private SharedPreferences eaten;
   private SharedPreferences calories;
   private SharedPreferences protein;
   private SharedPreferences fat;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.diaryview);
      initialize(savedInstanceState);
```

```
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      linear1 =  findViewById(R.id.linear1);
      toolbar =  findViewById(R.id.toolbar);
      vscroll1 =  findViewById(R.id.vscroll1);
      button1 = (Button) findViewById(R.id.button1);
      back =  findViewById(R.id.back);
      linear7 =  findViewById(R.id.linear7);
      linrMain =  findViewById(R.id.linrMain);
      textview7 = (TextView) findViewById(R.id.textview7);
      linrKcal =  findViewById(R.id.linrKcal);
      textview4 = (TextView) findViewById(R.id.textview4);
      linrCarbo =  findViewById(R.id.linrCarbo);
      linrProtn =  findViewById(R.id.linrProtn);
      linrFats =  findViewById(R.id.linrFats);
      linrSugar =  findViewById(R.id.linrSugar);
      linrSa =  findViewById(R.id.linrSa);
      circleimageview1 = (CircleImageView)
findViewById(R.id.circleimageview1);
      linear3 =  findViewById(R.id.linear3);
      txtFoodName = (TextView) findViewById(R.id.txtFoodName);
      textview8 = (TextView) findViewById(R.id.textview8);
      txtkCal = (TextView) findViewById(R.id.txtkCal);
      textview5 = (TextView) findViewById(R.id.textview5);
      txtCarbohydate = (TextView) findViewById(R.id.txtCarbohydate);
      textview10 = (TextView) findViewById(R.id.textview10);
      txtProtein = (TextView) findViewById(R.id.txtProtein);
      textview12 = (TextView) findViewById(R.id.textview12);
      txtFat = (TextView) findViewById(R.id.txtFat);
      textview14 = (TextView) findViewById(R.id.textview14);
      txtSugar = (TextView) findViewById(R.id.txtSugar);
      textview16 = (TextView) findViewById(R.id.textview16);
      txtSalt = (TextView) findViewById(R.id.txtSalt);
      Diary = getSharedPreferences("Diary", Activity.MODE_PRIVATE);
      eaten = getSharedPreferences("eaten", Activity.MODE_PRIVATE);
      calories = getSharedPreferences("calories", Activity.MODE_PRIVATE);
      protein = getSharedPreferences("protein", Activity.MODE_PRIVATE);
      fat = getSharedPreferences("fat", Activity.MODE_PRIVATE);

//    onClick will save the viewed recipe details
//    Using sharedPreferences, save the respective diary details with
appropriate keys and values
      button1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            calories.edit().putString("calories",
String.valueOf((long)(prevcalories +
Double.parseDouble(txtkCal.getText().toString())))).commit();
            protein.edit().putString("protein",
String.valueOf((long)(prevprotein +
Double.parseDouble(txtProtein.getText().toString())))).commit();
            fat.edit().putString("fat", String.valueOf((long)(prevfat +
Double.parseDouble(txtFat.getText().toString())))).commit();
            ToastUtil.showMessage(getApplicationContext(), "Yummy!!!");
            finish();
         }
      });
   }
```

```
   private void initializeLogic() {
      button1.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFF3F51B6));
      ONCREATE();

//    Check if the fat value has any saved data before
      if (fat.getString("fat", "").equals("")) {
//        Set the fat to 0 if there's no previous fat data saved
         prevfat = 0;
      }
      else {
//        Get the fat value if there's a saved data
         prevfat = Double.parseDouble(fat.getString("fat", ""));
      }

        //    Check if the protein value has any saved data before
      if (protein.getString("protein", "").equals("")) {
            //        Get the protein value if there's a saved data
         prevprotein = 0;
      }
      else {
            //        Get the protein value if there's a saved data
         prevprotein = Double.parseDouble(protein.getString("protein",
""));
      }

        //    Check if the calories value has any saved data before
      if (calories.getString("calories", "").equals("")) {
            //        Get the calories value if there's a saved data
         prevcalories = 0;
      }
      else {
            //        Get the calories value if there's a saved data
         prevcalories = Double.parseDouble(calories.getString("calories",
""));
      }
      try{

//        Set the text of respective data to textview
         txtFoodName.setText(getIntent().getStringExtra("Food"));
         txtkCal.setText(getIntent().getStringExtra("kCal"));
         txtProtein.setText(getIntent().getStringExtra("Protein"));
         txtFat.setText(getIntent().getStringExtra("Fat"));
         txtSalt.setText(getIntent().getStringExtra("Salt"));
         txtCarbohydate.setText(getIntent().getStringExtra("Carbo"));

Glide.with(getApplicationContext()).load(Uri.parse(getIntent().getStringExt
ra("Img"))).into(circleimageview1);
      }catch(Exception e){

      }
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {
```

```
            default:
            break;
        }
    }

    @Override
    public void onStart() {
        super.onStart();
        _Get_Diary();
    }
    public void _Get_Diary () {

    }


    public void ONCREATE () {

    }


    @Deprecated
    public void showMessage(String msg) {
        Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
    }

    @Deprecated
    public int getLocationX(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
```

```
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import android.graphics.drawable.GradientDrawable;
import android.os.Build;
import android.os.Bundle;
import android.text.InputType;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.view.WindowManager;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Random;
```

```
public class EditMealsActivity extends AppCompatActivity {

   private String Date = "";
   private String Time = "";
   private double Position = 0;
   private HashMap<String, Object> GetNote = new HashMap<>();
   private String fontName = "";
   private String typeace = "";

   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();
   private ArrayList<String> Classes = new ArrayList<>();

   private LinearLayout toolbar;
   private LinearLayout main;
   private ImageView back_img;
   private EditText title_edittext;
   private LinearLayout linear1;
   private LinearLayout line;
   private LinearLayout check_lin;
   private LinearLayout linear2;
   private LinearLayout linear3;
   private ScrollView vscroll1;
   private LinearLayout linear5;
   private LinearLayout linear6;
   private EditText edit_ingredient;
   private LinearLayout linear7;
   private EditText edit_method;
   private LinearLayout linear8;
   private LinearLayout linear9;
   private LinearLayout linear10;
   private EditText edit_calories;
   private LinearLayout linear14;
   private EditText editServing;
   private LinearLayout linear13;
   private TextView txtClassOfFood;
   private Spinner spinner1;
   private LinearLayout linear11;
   private Button button1;
   private ImageView imageview3;
   private TextView textview1;
   private ImageView imageview2;
   private TextView textview2;
   private ImageView imageview1;
   private TextView textview3;
   private EditText edittext1;
   private EditText editminutes;
   private ImageView imageview4;
   private TextView textview4;
   private ImageView imageview6;
   private TextView textview6;
   private ImageView imageview5;
   private TextView textview5;
   private CheckBox importantcheckbox;

   private Calendar Cal = Calendar.getInstance();
   private SharedPreferences AllMeals;
   private SharedPreferences Settings;
```

```
    @RequiresApi(api = Build.VERSION_CODES.M)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.edit_meals);
        initialize(savedInstanceState);
        com.google.firebase.FirebaseApp.initializeApp(this);

//      Clicking on edit button while viewing the meal in ViewMeal activity
//      The position of the meal is sent in putExtra with key "Position"
//      Get the "position" to Position int, as this would be helpful in
getting the meal to be edited
        Position =
Double.parseDouble(getIntent().getStringExtra("Position"));
        Cal = Calendar.getInstance();
        Typeface();

//      Check if the AllMeals sharedPreference contains any value
        if (!AllMeals.getString("Meals", "").equals("")) {
//          Get the value to ArrayList HashMap
            Notes_List = new Gson().fromJson(AllMeals.getString("Meals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
        }
        Date = new SimpleDateFormat("dd MMM").format(Cal.getTime());
        Time = new SimpleDateFormat("hh:mm a").format(Cal.getTime());
        Capitalize(title_edittext);
        getSettingsData();
        _GetNoteData();
    }

    private void initialize(Bundle savedInstanceState) {
        toolbar =  findViewById(R.id.toolbar);
        main =  findViewById(R.id.main);
        back_img =  findViewById(R.id.back_img);
        title_edittext = (EditText) findViewById(R.id.title_edittext);
        linear1 =  findViewById(R.id.linear1);
        line =  findViewById(R.id.line);
        check_lin =  findViewById(R.id.check_lin);
        linear2 =  findViewById(R.id.linear2);
        linear3 =  findViewById(R.id.linear3);
        vscroll1 =  findViewById(R.id.vscroll1);
        linear5 =  findViewById(R.id.linear5);
        linear6 =  findViewById(R.id.linear6);
        edit_ingredient = (EditText) findViewById(R.id.edit_ingredient);
        linear7 =  findViewById(R.id.linear7);
        edit_method = (EditText) findViewById(R.id.edit_method);
        linear8 =  findViewById(R.id.linear8);
        linear9 =  findViewById(R.id.linear9);
        linear10 =  findViewById(R.id.linear10);
        edit_calories = (EditText) findViewById(R.id.edit_calories);
        linear14 =  findViewById(R.id.linear14);
        editServing = (EditText) findViewById(R.id.editServing);
        linear13 =  findViewById(R.id.linear13);
        txtClassOfFood = (TextView) findViewById(R.id.txtClassOfFood);
        spinner1 = (Spinner) findViewById(R.id.spinner1);
        linear11 =  findViewById(R.id.linear11);
        button1 = (Button) findViewById(R.id.button1);
        imageview3 =  findViewById(R.id.imageview3);
        textview1 = (TextView) findViewById(R.id.textview1);
        imageview2 =  findViewById(R.id.imageview2);
        textview2 = (TextView) findViewById(R.id.textview2);
```

```
      imageview1 =  findViewById(R.id.imageview1);
      textview3 = (TextView) findViewById(R.id.textview3);
      edittext1 = (EditText) findViewById(R.id.edittext1);
      editminutes = (EditText) findViewById(R.id.editminutes);
      imageview4 =  findViewById(R.id.imageview4);
      textview4 = (TextView) findViewById(R.id.textview4);
      imageview6 =  findViewById(R.id.imageview6);
      textview6 = (TextView) findViewById(R.id.textview6);
      imageview5 =  findViewById(R.id.imageview5);
      textview5 = (TextView) findViewById(R.id.textview5);
      importantcheckbox = (CheckBox) findViewById(R.id.importantcheckbox);
      AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);
      Settings = getSharedPreferences("Settings", Activity.MODE_PRIVATE);

      back_img.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {

//          Hide keyboard and exit
            if (title_edittext.getText().toString().equals("") &&
edit_method.getText().toString().equals("")) {
               android.view.inputmethod.InputMethodManager imm =
(android.view.inputmethod.InputMethodManager)getSystemService(INPUT_METHOD_
SERVICE);

imm.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(), 0);
               finish();
            }
            else {
               if (title_edittext.getText().toString().equals("") &&
!edit_method.getText().toString().equals("")) {
                  ToastUtil.showMessage(getApplicationContext(), "Enter
Meal Title !!");
               }
               else {
                  if (edit_method.getText().toString().equals("") &&
!title_edittext.getText().toString().equals("")) {
                     ToastUtil.showMessage(getApplicationContext(), "Enter
Your Cooking Method !!");
                  }
                  else {

//                   For you to edit the saved meals, diary or saved items
//                   The step is as simple as this
//                   Check how the data is gotten from onCreate firstly
//                   Create an ArrayList HashMap to get the meal details
using the position int
                     // Set a key for each data such "Ingredients" for
input of ingredients entered by user
//                   don't change the key from the previous saved ones
                     Notes_List.get((int)Position).put("Note",
edit_method.getText().toString());
                     Notes_List.get((int)Position).put("Title",
title_edittext.getText().toString());
                     Notes_List.get((int)Position).put("Method",
edit_method.getText().toString());
                     Notes_List.get((int)Position).put("Calories",
edit_calories.getText().toString());
                     Notes_List.get((int)Position).put("Ingredients",
edit_ingredient.getText().toString());
                     Notes_List.get((int)Position).put("Class",
```

```
txtClassOfFood.getText().toString());
                    Notes_List.get((int)Position).put("Hours",
edittext1.getText().toString());
                    Notes_List.get((int)Position).put("Minutes",
editminutes.getText().toString());
                    Notes_List.get((int)Position).put("Serving",
editServing.getText().toString());
                    if (importantcheckbox.isChecked()) {
                       Notes_List.get((int)Position).put("Important",
"True");
                    }
                    else {
                       Notes_List.get((int)Position).put("Important",
"False");
                    }
                    Notes_List.get((int)Position).put("Date", Date);
                    Notes_List.get((int)Position).put("Time", Time);
                    AllMeals.edit().putString("Meals", new
Gson().toJson(Notes_List)).commit();
                    finish();
                 }
              }
           }
        });

        spinner1.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
           @Override
           public void onItemSelected(AdapterView<?> param1, View param2, int
param3, long param4) {
              final int position = param3;
              if (position == 0) {
                 txtClassOfFood.setText("Unknown");
                 spinner1.setSelection((int)(0));
              }
              if (position == 1) {
                 txtClassOfFood.setText(Classes.get((int)(position)));
                 spinner1.setSelection((int)(1));
              }
              if (position == 2) {
                 txtClassOfFood.setText(Classes.get((int)(position)));
                 spinner1.setSelection((int)(2));
              }
              if (position == 3) {
                 txtClassOfFood.setText(Classes.get((int)(position)));
                 spinner1.setSelection((int)(3));
              }
              if (position == 4) {
                 txtClassOfFood.setText(Classes.get((int)(position)));
                 spinner1.setSelection((int)(4));
              }
              if (position == 5) {
                 txtClassOfFood.setText(Classes.get((int)(position)));
                 spinner1.setSelection((int)(5));
              }
              if (position == 6) {
                 txtClassOfFood.setText(Classes.get((int)(position)));
                 spinner1.setSelection((int)(6));
              }
           }
```

```
        @Override
        public void onNothingSelected(AdapterView<?> param1) {

        }
      });

      button1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
          back_img.performClick();
        }
      });
    }

    @RequiresApi(api = Build.VERSION_CODES.M)
    private void initializeLogic() {

      Classes.add("Select Food Class");
      Classes.add("Carbohydrate");
      Classes.add("Protein");
      Classes.add("Fats And Oil");
      Classes.add("Minerals");
      Classes.add("Vitamins");
      Classes.add("Fibre And Roughages");
      spinner1.setAdapter(new ArrayAdapter<String>(getBaseContext(),
android.R.layout.simple_spinner_dropdown_item, Classes));
      ((ArrayAdapter)spinner1.getAdapter()).notifyDataSetChanged();
      edit_calories.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFFEEEEEE));
      edittext1.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFFEEEEEE));
      editminutes.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFFEEEEEE));
      button1.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFF3F51B6));
      editServing.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFFEEEEEE));
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

        default:
        break;
      }
    }

    @Override
    public void onBackPressed() {
      back_img.performClick();
    }
    public void Typeface () {
```

```
      changeActivityFont("railway_semisolid");
   }


   @RequiresApi(api = Build.VERSION_CODES.M)
   public void ONCREATE () {

   }


   @RequiresApi(api = Build.VERSION_CODES.M)
   public void getSettingsData () {
      if (Settings.getString("Text Size", "").equals("Small")) {

      }
      else {
         if (Settings.getString("Text Size", "").equals("Medium")) {
            textSize(title_edittext, 18);
            textSize(importantcheckbox, 18);
         }
         else {
            if (Settings.getString("Text Size", "").equals("Large")) {
               textSize(title_edittext, 20);
               textSize(importantcheckbox, 20);
            }
            else {

            }
         }
      }
      if (Settings.getString("Theme", "").equals("Default")) {
         Window w =
EditMealsActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.
FLAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_S
YSTEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#FFFFFF"));
main.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
         Add("#000000", back_img);
      }
      else {
         if (Settings.getString("Theme", "").equals("Dark")) {
            Window w =
EditMealsActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.
FLAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_S
YSTEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#000000"));
            toolbar.setBackgroundColor(0xFF263238);
            main.setBackgroundColor(0xFF263238);
            title_edittext.setTextColor(0xFFFFFFFF);
            title_edittext.setHintTextColor(0xFFFFFFFF);


            importantcheckbox.setTextColor(0xFFFFFFFF);
            check_lin.setBackgroundColor(0xFF263238);


importantcheckbox.getButtonDrawable().setColorFilter(Color.parseColor("#FFF
FFF"), PorterDuff.Mode.SRC_IN);
            Add("#FFFFFF", back_img);
         }
         else {
            if (Settings.getString("Theme", "").equals("Blue Grey")) {
               Window w =
```

```
EditMealsActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.
FLAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_S
YSTEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#FFFFFF"));
main.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
                toolbar.setBackgroundColor(0xFFFFFFFF);
                main.setBackgroundColor(0xFFFFFFFF);
                title_edittext.setTextColor(0xFF607D8B);
                title_edittext.setHintTextColor(0xFF607D8B);


                importantcheckbox.setTextColor(0xFF607D8B);

importantcheckbox.getButtonDrawable().setColorFilter(Color.parseColor("#607
D8B"), PorterDuff.Mode.SRC_IN);
                Add("#000000", back_img);
            }
            else {
                if (Settings.getString("Theme", "").equals("Orange")) {
                    Window w =
EditMealsActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.
FLAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_S
YSTEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#FFFFFF"));
main.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
                    toolbar.setBackgroundColor(0xFFFFFFFF);
                    main.setBackgroundColor(0xFFFFFFFF);
                    title_edittext.setTextColor(0xFFFF5722);
                    title_edittext.setHintTextColor(0xFFFF5722);


                    importantcheckbox.setTextColor(0xFFFF5722);

importantcheckbox.getButtonDrawable().setColorFilter(Color.parseColor("#FF5
722"), PorterDuff.Mode.SRC_IN);
                    Add("#000000", back_img);
                }
                else {
                    if (Settings.getString("Theme", "").equals("Indigo")) {
                        Window w =
EditMealsActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.
FLAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_S
YSTEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#FFFFFF"));
main.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
                        toolbar.setBackgroundColor(0xFFFFFFFF);
                        main.setBackgroundColor(0xFFFFFFFF);
                        title_edittext.setTextColor(0xFF3F51B5);
                        title_edittext.setHintTextColor(0xFF3F51B5);


                        importantcheckbox.setTextColor(0xFF3F51B5);

importantcheckbox.getButtonDrawable().setColorFilter(Color.parseColor("#3F5
1B5"), PorterDuff.Mode.SRC_IN);
                        Add("#000000", back_img);
                    }
                    else {

                    }
                }
            }
        }
    }
```

```
    }


    public void textSize (final TextView TextView1, final double size) {
        int j = (int) size;
        TextView1.setTextSize(j);
    }


    public void Capitalize (final TextView Edittext) {

Edittext.setRawInputType(InputType.TYPE_CLASS_TEXT|InputType.TYPE_TEXT_FLAG
_CAP_SENTENCES);
    }


    public void Add (final String Colour, final ImageView Imageview) {
        Imageview.getDrawable().setColorFilter(Color.parseColor(Colour),
PorterDuff.Mode.SRC_IN);
    }


    public void _GetNoteData () {

//      Get the meals into edit text from the position
        GetNote = Notes_List.get((int)Position);
        if (GetNote.get("Important").toString().equals("True")) {
            importantcheckbox.setChecked(true);
        }
        else {
            importantcheckbox.setChecked(false);
        }
        title_edittext.setText(GetNote.get("Title").toString());
        edit_ingredient.setText(GetNote.get("Ingredients").toString());
        edit_method.setText(GetNote.get("Method").toString());
        edit_calories.setText(GetNote.get("Calories").toString());
        edittext1.setText(GetNote.get("Hours").toString());
        editminutes.setText(GetNote.get("Minutes").toString());
    }


    public void changeActivityFont (final String fontname) {
        fontName = "fonts/".concat(fontname.concat(".ttf"));
        overrideFonts(this,getWindow().getDecorView());
    }
    private void overrideFonts(final android.content.Context context, final
View v) {

        try {
            Typeface
            typeace = Typeface.createFromAsset(getAssets(), fontName);;
            if ((v instanceof ViewGroup)) {
                ViewGroup vg = (ViewGroup) v;
                for (int i = 0;
                i < vg.getChildCount();
                i++) {
                    View child = vg.getChildAt(i);
                    overrideFonts(context, child);
                }
            }
            else {
```

```
            if ((v instanceof TextView)) {
               ((TextView) v).setTypeface(typeace);
            }
            else {
               if ((v instanceof EditText )) {
                  ((EditText) v).setTypeface(typeace);
               }
               else {
                  if ((v instanceof Button)) {
                     ((Button) v).setTypeface(typeace);
                  }
               }
            }
         }
      }
      catch(Exception e)

      {

      };
   }


   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }

   @Deprecated
   public int getRandom(int min, int max) {
      Random random = new Random();
      return random.nextInt(max - min + 1) + min;
   }

   @Deprecated
   public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
      ArrayList<Double> result = new ArrayList<Double>();
      SparseBooleanArray arr = _list.getCheckedItemPositions();
      for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
         if (arr.valueAt(iIdx))
         result.add((double)arr.keyAt(iIdx));
      }
      return result;
   }
```

```
    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Edit Profile Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.drawable.GradientDrawable;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.coordinatorlayout.widget.CoordinatorLayout;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.appbar.AppBarLayout;
import com.google.android.material.button.MaterialButton;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.GenericTypeIndicator;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;
```

```
public class EditprofileActivity extends AppCompatActivity {
   private FirebaseDatabase firebase = FirebaseDatabase.getInstance();

   private Toolbar _toolbar;
   private AppBarLayout _app_bar;
   private CoordinatorLayout _coordinator;
   private HashMap<String, Object> uid = new HashMap<>();
   private HashMap<String, Object> sa_map = new HashMap<>();

   private ScrollView vscroll1;
   private LinearLayout linear19;
   private LinearLayout linear3;
   private LinearLayout linear_username;
   private LinearLayout linear_email;
   private LinearLayout linear_password;
   private LinearLayout linear20;
   private MaterialButton materialbutton1;
   private TextView usernam;
   private EditText edit_user;
   private TextView email;
   private EditText edit_email;
   private TextView password;
   private EditText editHeight;
   private TextView textview1;
   private EditText editWeight;

   private DatabaseReference sa_db = firebase.getReference("sa_db");
   private ChildEventListener sa_db_child_listener;
   private FirebaseAuth auth;
   private OnCompleteListener<Void> auth_updateEmailListener;
   private OnCompleteListener<Void> auth_updatePasswordListener;
   private OnCompleteListener<Void> auth_emailVerificationSentListener;
   private OnCompleteListener<Void> auth_deleteUserListener;
   private OnCompleteListener<Void> auth_updateProfileListener;
   private OnCompleteListener<AuthResult> auth_phoneAuthListener;
   private OnCompleteListener<AuthResult> auth_googleSignInListener;
   private OnCompleteListener<AuthResult> auth_create_user_listener;
   private OnCompleteListener<AuthResult> authsign_in_listener;
   private OnCompleteListener<Void> _auth_reset_password_listener;
   private SharedPreferences name;
   private SharedPreferences userdetail;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.editprofile);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      _app_bar = (AppBarLayout) findViewById(R.id._app_bar);
      _coordinator = (CoordinatorLayout) findViewById(R.id._coordinator);
      _toolbar = (Toolbar) findViewById(R.id._toolbar);
      setSupportActionBar(_toolbar);
      getSupportActionBar().setDisplayHomeAsUpEnabled(true);
      getSupportActionBar().setHomeButtonEnabled(true);
      _toolbar.setNavigationOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View v) {
            onBackPressed();
        }
    });
    vscroll1 =  findViewById(R.id.vscroll1);
    linear19 =  findViewById(R.id.linear19);
    linear3 =  findViewById(R.id.linear3);
    linear_username =  findViewById(R.id.linear_username);
    linear_email =  findViewById(R.id.linear_email);
    linear_password =  findViewById(R.id.linear_password);
    linear20 =  findViewById(R.id.linear20);
    materialbutton1 = (MaterialButton)
findViewById(R.id.materialbutton1);
    usernam = (TextView) findViewById(R.id.usernam);
    edit_user = (EditText) findViewById(R.id.edit_user);
    email = (TextView) findViewById(R.id.email);
    edit_email = (EditText) findViewById(R.id.edit_email);
    password = (TextView) findViewById(R.id.password);
    editHeight = (EditText) findViewById(R.id.editHeight);
    textview1 = (TextView) findViewById(R.id.textview1);
    editWeight = (EditText) findViewById(R.id.editWeight);
    auth = FirebaseAuth.getInstance();
    name = getSharedPreferences("name", Activity.MODE_PRIVATE);
    userdetail = getSharedPreferences("userdetail",
Activity.MODE_PRIVATE);

    materialbutton1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

//          Update the user new detail entered
//          With the same HashMap used in creating the user detail on
firebase database
            sa_map = new HashMap<>();
            sa_map.put("height", editHeight.getText().toString());
            sa_map.put("username", edit_user.getText().toString());
            sa_map.put("weight", editWeight.getText().toString());

sa_db.child(FirebaseAuth.getInstance().getCurrentUser().getUid()).updateChi
ldren(sa_map);
            ToastUtil.showMessage(getApplicationContext(), "Updated
Successfully");
            finish();
        }
    });

    sa_db_child_listener = new ChildEventListener() {
        @Override
        public void onChildAdded(DataSnapshot param1, String param2) {
            GenericTypeIndicator<HashMap<String, Object>> ind = new
GenericTypeIndicator<HashMap<String, Object>>() {};
            final String childKey = param1.getKey();
            final HashMap<String, Object> childValue =
param1.getValue(ind);

        }

        @Override
        public void onChildChanged(DataSnapshot param1, String param2) {
            GenericTypeIndicator<HashMap<String, Object>> ind = new
```

```
GenericTypeIndicator<HashMap<String, Object>>() {};
            final String childKey = param1.getKey();
            final HashMap<String, Object> childValue =
param1.getValue(ind);

         }

         @Override
         public void onChildMoved(DataSnapshot param1, String param2) {

         }

         @Override
         public void onChildRemoved(DataSnapshot param1) {
            GenericTypeIndicator<HashMap<String, Object>> ind = new
GenericTypeIndicator<HashMap<String, Object>>() {};
            final String childKey = param1.getKey();
            final HashMap<String, Object> childValue =
param1.getValue(ind);

         }

         @Override
         public void onCancelled(DatabaseError param1) {
            final int _errorCode = param1.getCode();
            final String errorMessage = param1.getMessage();

         }
      };
      sa_db.addChildEventListener(sa_db_child_listener);

      auth_updateEmailListener = new OnCompleteListener<Void>() {
         @Override
         public void onComplete(Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };

      auth_updatePasswordListener = new OnCompleteListener<Void>() {
         @Override
         public void onComplete(Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };

      auth_emailVerificationSentListener = new OnCompleteListener<Void>() {
         @Override
         public void onComplete(Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };
```

```
        auth_deleteUserListener = new OnCompleteListener<Void>() {
            @Override
            public void onComplete(Task<Void> param1) {
                final boolean success = param1.isSuccessful();
                final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

            }
        };

        auth_phoneAuthListener = new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(Task<AuthResult> task){
                final boolean success = task.isSuccessful();
                final String errorMessage = task.getException() != null ?
task.getException().getMessage() : "";

            }
        };

        auth_updateProfileListener = new OnCompleteListener<Void>() {
            @Override
            public void onComplete(Task<Void> param1) {
                final boolean success = param1.isSuccessful();
                final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

            }
        };

        auth_googleSignInListener = new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(Task<AuthResult> task){
                final boolean success = task.isSuccessful();
                final String errorMessage = task.getException() != null ?
task.getException().getMessage() : "";

            }
        };

        auth_create_user_listener = new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(Task<AuthResult> param1) {
                final boolean success = param1.isSuccessful();
                final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

            }
        };

        authsign_in_listener = new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(Task<AuthResult> param1) {
                final boolean success = param1.isSuccessful();
                final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

            }
        };

        _auth_reset_password_listener = new OnCompleteListener<Void>() {
```

```
        @Override
        public void onComplete(Task<Void> param1) {
           final boolean success = param1.isSuccessful();

        }
      };
   }

   private void initializeLogic() {
      edit_user.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)10, 0xFFEEEEEE));
      edit_email.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)10, 0xFFEEEEEE));
      editHeight.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)10, 0xFFEEEEEE));
      editWeight.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)10, 0xFFEEEEEE));
      edit_user.setText(name.getString("name", ""));
      editWeight.setText(userdetail.getString("weight", ""));
      editHeight.setText(userdetail.getString("height", ""));
      edit_email.setText(userdetail.getString("email", ""));
      edit_email.setEnabled(false);
      materialbutton1.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFF3F51B6));
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }
```

```
    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Enter Meal Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.app.ActivityOptions;
import android.content.Intent;
import android.graphics.drawable.GradientDrawable;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
```

```
import com.airbnb.lottie.LottieAnimationView;
import com.google.android.material.button.MaterialButton;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;


public class EntermealActivity extends AppCompatActivity {

   private String data = "";
   private String type = "";

   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();

   private LinearLayout linear1;
   private LinearLayout linear3;
   private TextView textview2;
   private MaterialButton materialbutton1;
   private LinearLayout linear2;
   private LottieAnimationView lottie1;
   private TextView textview1;
   private MaterialButton materialbutton2;
   private MaterialButton materialbutton3;

   private Intent search = new Intent();
   private Intent To_Notes = new Intent();

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.entermeal);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      linear1 =  findViewById(R.id.linear1);
      linear3 =  findViewById(R.id.linear3);
      textview2 = (TextView) findViewById(R.id.textview2);
      materialbutton1 = (MaterialButton)
findViewById(R.id.materialbutton1);
      linear2 =  findViewById(R.id.linear2);
      lottie1 = (LottieAnimationView) findViewById(R.id.lottie1);
      textview1 = (TextView) findViewById(R.id.textview1);
      materialbutton2 = (MaterialButton)
findViewById(R.id.materialbutton2);
      materialbutton3 = (MaterialButton)
findViewById(R.id.materialbutton3);

      materialbutton1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            To_Notes.setAction(Intent.ACTION_VIEW);
            To_Notes.setClass(getApplicationContext(),
AddMealActivity.class);
            startActivity(To_Notes);
         }
```

```
      });

      materialbutton2.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            search.setClass(getApplicationContext(),
SearchscanmealActivity.class);
            startActivity(search);
         }
      });

      materialbutton3.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            search.setClass(getApplicationContext(),
ScancodeActivity.class);
            startActivity(search);
         }
      });
   }

   private void initializeLogic() {
      materialbutton1.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFF3F51B6));
      materialbutton2.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFF3F51B6));
      materialbutton3.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFF3F51B6));
      if (getIntent().getBooleanExtra("dT",true)) {
         // To make the dialog corners round

getWindow().getDecorView().setBackgroundResource(android.R.color.transparen
t);


         {
            android.graphics.drawable.GradientDrawable designUI = new
android.graphics.drawable.GradientDrawable();
            int d = (int)
getApplicationContext().getResources().getDisplayMetrics().density;
            designUI.setColor(0xFFFFFF);designUI.setCornerRadius(d*15);


((ViewGroup)getWindow().getDecorView()).getChildAt(0).setBackground(designU
I);
         }

         // Codes Generated by designUI
      }
      // you may face some issues if you use enable ActionBar due to dialog
theme
      data = getIntent().getStringExtra("Data");
      type = getIntent().getStringExtra("Type");
      linear1.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)20, 0xFFFFFFFF));
   }

   @Override
```

```
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   public void _dialogTheme () {
   }
   // setTheme() should be set before setContentView()
    @Override
       public void setContentView( int layoutResID) {
      if(getIntent().getBooleanExtra("dialogTheme",true)){
         supportRequestWindowFeature(Window.FEATURE_NO_TITLE);
         setTheme(R.style.Theme_AppCompat_Light_Dialog);
         setFinishOnTouchOutside(true);

         //change true to false if you want to make dialog non cancellable
when clicked outside
         //if you want to use this without app compat  change
supportRequestWindowFeature() and setTheme() to below codes.
         /*
requestWindowFeature(Window.FEATURE_NO_TITLE);
setTheme(android.R.style.Theme_Dialog);
*/
         // Calling this allows the Activity behind this one to be seen
again. Once all such Activities have been redrawn
         try {
               java.lang.reflect.Method getActivityOptions =
Activity.class.getDeclaredMethod("getActivityOptions");
getActivityOptions.setAccessible(true);
            Object options = getActivityOptions.invoke(this); Class<?>[]
classes = Activity.class.getDeclaredClasses(); Class<?>
translucentConversionListenerClazz = null;
           for (Class clazz : classes) { if
(clazz.getSimpleName().contains("TranslucentConversionListener")) {
translucentConversionListenerClazz = clazz; } }
           java.lang.reflect.Method convertToTranslucent =
Activity.class.getDeclaredMethod("convertToTranslucent",
translucentConversionListenerClazz, ActivityOptions.class);
convertToTranslucent.setAccessible(true); convertToTranslucent.invoke(this,
null, options); } catch (Throwable t) {
         }
      }
      super.setContentView(layoutResID);
   }
   {
   }


   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
```

```
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

FileUtil Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.content.ContentResolver;
import android.content.ContentUris;
import android.content.Context;
import android.database.Cursor;
import android.graphics.Bitmap;
```

```
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.ColorFilter;
import android.graphics.ColorMatrix;
import android.graphics.ColorMatrixColorFilter;
import android.graphics.LightingColorFilter;
import android.graphics.Matrix;
import android.graphics.Paint;
import android.graphics.PorterDuff;
import android.graphics.PorterDuffXfermode;
import android.graphics.Rect;
import android.graphics.RectF;
import android.media.ExifInterface;
import android.net.Uri;
import android.os.Environment;
import android.provider.DocumentsContract;
import android.provider.MediaStore;
import android.text.TextUtils;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.net.URLDecoder;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

public class FileUtil {

   private static void createNewFile(String path) {
      int lastSep = path.lastIndexOf(File.separator);
      if (lastSep > 0) {
         String dirPath = path.substring(0, lastSep);
         makeDir(dirPath);
      }

      File file = new File(path);

      try {
         if (!file.exists())
            file.createNewFile();
      } catch (IOException e) {
         e.printStackTrace();
      }
   }

   public static String readFile(String path) {
      createNewFile(path);

      StringBuilder sb = new StringBuilder();
      FileReader fr = null;
      try {
         fr = new FileReader(new File(path));

         char[] buff = new char[1024];
         int length = 0;

         while ((length = fr.read(buff)) > 0) {
```

```
            sb.append(new String(buff, 0, length));
         }
      } catch (IOException e) {
         e.printStackTrace();
      } finally {
         if (fr != null) {
            try {
               fr.close();
            } catch (Exception e) {
               e.printStackTrace();
            }
         }
      }

      return sb.toString();
   }

   public static void writeFile(String path, String str) {
      createNewFile(path);
      FileWriter fileWriter = null;

      try {
         fileWriter = new FileWriter(new File(path), false);
         fileWriter.write(str);
         fileWriter.flush();
      } catch (IOException e) {
         e.printStackTrace();
      } finally {
         try {
            if (fileWriter != null)
               fileWriter.close();
         } catch (IOException e) {
            e.printStackTrace();
         }
      }
   }

   public static void copyFile(String sourcePath, String destPath) {
      if (!isExistFile(sourcePath)) return;
      createNewFile(destPath);

      FileInputStream fis = null;
      FileOutputStream fos = null;

      try {
         fis = new FileInputStream(sourcePath);
         fos = new FileOutputStream(destPath, false);

         byte[] buff = new byte[1024];
         int length = 0;

         while ((length = fis.read(buff)) > 0) {
            fos.write(buff, 0, length);
         }
      } catch (IOException e) {
         e.printStackTrace();
      } finally {
         if (fis != null) {
            try {
               fis.close();
            } catch (IOException e) {
```

```
            e.printStackTrace();
         }
      }
      if (fos != null) {
         try {
            fos.close();
         } catch (IOException e) {
            e.printStackTrace();
         }
      }
   }
}

public static void moveFile(String sourcePath, String destPath) {
   copyFile(sourcePath, destPath);
   deleteFile(sourcePath);
}

public static void deleteFile(String path) {
   File file = new File(path);

   if (!file.exists()) return;

   if (file.isFile()) {
      file.delete();
      return;
   }

   File[] fileArr = file.listFiles();

   if (fileArr != null) {
      for (File subFile : fileArr) {
         if (subFile.isDirectory()) {
            deleteFile(subFile.getAbsolutePath());
         }

         if (subFile.isFile()) {
            subFile.delete();
         }
      }
   }

   file.delete();
}

public static boolean isExistFile(String path) {
   File file = new File(path);
   return file.exists();
}

public static void makeDir(String path) {
   if (!isExistFile(path)) {
      File file = new File(path);
      file.mkdirs();
   }
}

public static void listDir(String path, ArrayList<String> list) {
   File dir = new File(path);
   if (!dir.exists() || dir.isFile()) return;
```

```
      File[] listFiles = dir.listFiles();
      if (listFiles == null || listFiles.length <= 0) return;

      if (list == null) return;
      list.clear();
      for (File file : listFiles) {
         list.add(file.getAbsolutePath());
      }
   }

   public static boolean isDirectory(String path) {
      if (!isExistFile(path)) return false;
      return new File(path).isDirectory();
   }

   public static boolean isFile(String path) {
      if (!isExistFile(path)) return false;
      return new File(path).isFile();
   }

   public static long getFileLength(String path) {
      if (!isExistFile(path)) return 0;
      return new File(path).length();
   }

   public static String getExternalStorageDir() {
      return Environment.getExternalStorageDirectory().getAbsolutePath();
   }

   public static String getPackageDataDir(Context context) {
      return context.getExternalFilesDir(null).getAbsolutePath();
   }

   public static String getPublicDir(String type) {
      return
Environment.getExternalStoragePublicDirectory(type).getAbsolutePath();
   }

   public static String convertUriToFilePath(final Context context, final
Uri uri) {
      String path = null;
      if (DocumentsContract.isDocumentUri(context, uri)) {
         if (isExternalStorageDocument(uri)) {
            final String docId = DocumentsContract.getDocumentId(uri);
            final String[] split = docId.split(":");
            final String type = split[0];

            if ("primary".equalsIgnoreCase(type)) {
               path = Environment.getExternalStorageDirectory() + "/" +
split[1];
            }
         } else if (isDownloadsDocument(uri)) {
            final String id = DocumentsContract.getDocumentId(uri);

            if (!TextUtils.isEmpty(id)) {
               if (id.startsWith("raw:")) {
                  return id.replaceFirst("raw:", "");
               }
            }

            final Uri contentUri = ContentUris
```

```
.withAppendedId(Uri.parse("content://downloads/public_downloads"),
Long.valueOf(id));

            path = getDataColumn(context, contentUri, null, null);
         } else if (isMediaDocument(uri)) {
            final String docId = DocumentsContract.getDocumentId(uri);
            final String[] split = docId.split(":");
            final String type = split[0];

            Uri contentUri = null;
            if ("image".equals(type)) {
               contentUri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;
            } else if ("video".equals(type)) {
               contentUri = MediaStore.Video.Media.EXTERNAL_CONTENT_URI;
            } else if ("audio".equals(type)) {
               contentUri = MediaStore.Audio.Media.EXTERNAL_CONTENT_URI;
            }

            final String selection = "_id=?";
            final String[] selectionArgs = new String[]{
                  split[1]
            };

            path = getDataColumn(context, contentUri, selection,
selectionArgs);
         }
      } else if
(ContentResolver.SCHEME_CONTENT.equalsIgnoreCase(uri.getScheme())) {
         path = getDataColumn(context, uri, null, null);
      } else if
(ContentResolver.SCHEME_FILE.equalsIgnoreCase(uri.getScheme())) {
         path = uri.getPath();
      }

      if (path != null) {
         try {
            return URLDecoder.decode(path, "UTF-8");
         }catch(Exception e){
            return null;
         }
      }
      return null;
   }

   private static String getDataColumn(Context context, Uri uri, String
selection, String[] selectionArgs) {
      Cursor cursor = null;

      final String column = MediaStore.Images.Media.DATA;
      final String[] projection = {
            column
      };

      try {
         cursor = context.getContentResolver().query(uri, projection,
selection, selectionArgs, null);
         if (cursor != null && cursor.moveToFirst()) {
            final int columnindex = cursor.getColumnIndexOrThrow(column);
            return cursor.getString(columnindex);
         }
```

```
      } catch (Exception e) {

      } finally {
         if (cursor != null) {
            cursor.close();
         }
      }
      return null;
   }


   private static boolean isExternalStorageDocument(Uri uri) {
      return
"com.android.externalstorage.documents".equals(uri.getAuthority());
   }

   private static boolean isDownloadsDocument(Uri uri) {
      return
"com.android.providers.downloads.documents".equals(uri.getAuthority());
   }

   private static boolean isMediaDocument(Uri uri) {
      return
"com.android.providers.media.documents".equals(uri.getAuthority());
   }

   private static void saveBitmap(Bitmap bitmap, String destPath) {
      FileOutputStream out = null;
      FileUtil.createNewFile(destPath);
      try {
         out = new FileOutputStream(new File(destPath));
         bitmap.compress(Bitmap.CompressFormat.PNG, 100, out);
      } catch (Exception e) {
         e.printStackTrace();
      } finally {
         try {
            if (out != null) {
               out.close();
            }
         } catch (IOException e) {
            e.printStackTrace();
         }
      }
   }

   public static Bitmap getScaledBitmap(String path, int max) {
      Bitmap src = BitmapFactory.decodeFile(path);

      int width = src.getWidth();
      int height = src.getHeight();
      float rate = 0.0f;

      if (width > height) {
         rate = max / (float) width;
         height = (int) (height * rate);
         width = max;
      } else {
         rate = max / (float) height;
         width = (int) (width * rate);
         height = max;
      }
```

```
      return Bitmap.createScaledBitmap(src, width, height, true);
   }

   public static int calculateInSampleSize(BitmapFactory.Options options,
int reqWidth, int reqHeight) {
      final int width = options.outWidth;
      final int height = options.outHeight;
      int inSampleSize = 1;

      if (height > reqHeight || width > reqWidth) {
         final int halfHeight = height / 2;
         final int halfWidth = width / 2;

         while ((halfHeight / inSampleSize) >= reqHeight && (halfWidth /
inSampleSize) >= reqWidth) {
            inSampleSize *= 2;
         }
      }

      return inSampleSize;
   }

   public static Bitmap decodeSampleBitmapFromPath(String path, int
reqWidth, int reqHeight) {
      final BitmapFactory.Options options = new BitmapFactory.Options();
      options.inJustDecodeBounds = true;
      BitmapFactory.decodeFile(path, options);

      options.inSampleSize = calculateInSampleSize(options, reqWidth,
reqHeight);

      options.inJustDecodeBounds = false;
      return BitmapFactory.decodeFile(path, options);
   }

   public static void resizeBitmapFileRetainRatio(String fromPath, String
destPath, int max) {
      if (!isExistFile(fromPath)) return;
      Bitmap bitmap = getScaledBitmap(fromPath, max);
      saveBitmap(bitmap, destPath);
   }

   public static void resizeBitmapFileToSquare(String fromPath, String
destPath, int max) {
      if (!isExistFile(fromPath)) return;
      Bitmap src = BitmapFactory.decodeFile(fromPath);
      Bitmap bitmap = Bitmap.createScaledBitmap(src, max, max, true);
      saveBitmap(bitmap, destPath);
   }

   public static void resizeBitmapFileToCircle(String fromPath, String
destPath) {
      if (!isExistFile(fromPath)) return;
      Bitmap src = BitmapFactory.decodeFile(fromPath);
      Bitmap bitmap = Bitmap.createBitmap(src.getWidth(),
            src.getHeight(), Bitmap.Config.ARGB_8888);
      Canvas canvas = new Canvas(bitmap);

      final int color = 0xff424242;
      final Paint paint = new Paint();
```

```
      final Rect rect = new Rect(0, 0, src.getWidth(), src.getHeight());

      paint.setAntiAlias(true);
      canvas.drawARGB(0, 0, 0, 0);
      paint.setColor(color);
      canvas.drawCircle(src.getWidth() / 2, src.getHeight() / 2,
            src.getWidth() / 2, paint);
      paint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.SRC_IN));
      canvas.drawBitmap(src, rect, rect, paint);

      saveBitmap(bitmap, destPath);
   }

   public static void resizeBitmapFileWithRoundedBorder(String fromPath,
String destPath, int pixels) {
      if (!isExistFile(fromPath)) return;
      Bitmap src = BitmapFactory.decodeFile(fromPath);
      Bitmap bitmap = Bitmap.createBitmap(src.getWidth(), src
            .getHeight(), Bitmap.Config.ARGB_8888);
      Canvas canvas = new Canvas(bitmap);

      final int color = 0xff424242;
      final Paint paint = new Paint();
      final Rect rect = new Rect(0, 0, src.getWidth(), src.getHeight());
      final RectF rectF = new RectF(rect);
      final float roundPx = pixels;

      paint.setAntiAlias(true);
      canvas.drawARGB(0, 0, 0, 0);
      paint.setColor(color);
      canvas.drawRoundRect(rectF, roundPx, roundPx, paint);

      paint.setXfermode(new PorterDuffXfermode(PorterDuff.Mode.SRC_IN));
      canvas.drawBitmap(src, rect, rect, paint);

      saveBitmap(bitmap, destPath);
   }

   public static void cropBitmapFileFromCenter(String fromPath, String
destPath, int w, int h) {
      if (!isExistFile(fromPath)) return;
      Bitmap src = BitmapFactory.decodeFile(fromPath);

      int width = src.getWidth();
      int height = src.getHeight();

      if (width < w && height < h)
         return;

      int x = 0;
      int y = 0;

      if (width > w)
         x = (width - w) / 2;

      if (height > h)
         y = (height - h) / 2;

      int cw = w;
      int ch = h;
```

```
      if (w > width)
         cw = width;

      if (h > height)
         ch = height;

      Bitmap bitmap = Bitmap.createBitmap(src, x, y, cw, ch);
      saveBitmap(bitmap, destPath);
   }

   public static void rotateBitmapFile(String fromPath, String destPath,
float angle) {
      if (!isExistFile(fromPath)) return;
      Bitmap src = BitmapFactory.decodeFile(fromPath);
      Matrix matrix = new Matrix();
      matrix.postRotate(angle);
      Bitmap bitmap = Bitmap.createBitmap(src, 0, 0, src.getWidth(),
src.getHeight(), matrix, true);
      saveBitmap(bitmap, destPath);
   }

   public static void scaleBitmapFile(String fromPath, String destPath,
float x, float y) {
      if (!isExistFile(fromPath)) return;
      Bitmap src = BitmapFactory.decodeFile(fromPath);
      Matrix matrix = new Matrix();
      matrix.postScale(x, y);

      int w = src.getWidth();
      int h = src.getHeight();

      Bitmap bitmap = Bitmap.createBitmap(src, 0, 0, w, h, matrix, true);
      saveBitmap(bitmap, destPath);
   }

   public static void skewBitmapFile(String fromPath, String destPath,
float x, float y) {
      if (!isExistFile(fromPath)) return;
      Bitmap src = BitmapFactory.decodeFile(fromPath);
      Matrix matrix = new Matrix();
      matrix.postSkew(x, y);

      int w = src.getWidth();
      int h = src.getHeight();

      Bitmap bitmap = Bitmap.createBitmap(src, 0, 0, w, h, matrix, true);
      saveBitmap(bitmap, destPath);
   }

   public static void setBitmapFileColorFilter(String fromPath, String
destPath, int color) {
      if (!isExistFile(fromPath)) return;
      Bitmap src = BitmapFactory.decodeFile(fromPath);
      Bitmap bitmap = Bitmap.createBitmap(src, 0, 0,
            src.getWidth() - 1, src.getHeight() - 1);
      Paint p = new Paint();
      ColorFilter filter = new LightingColorFilter(color, 1);
      p.setColorFilter(filter);
      Canvas canvas = new Canvas(bitmap);
      canvas.drawBitmap(bitmap, 0, 0, p);
      saveBitmap(bitmap, destPath);
```

```
    }

    public static void setBitmapFileBrightness(String fromPath, String
destPath, float brightness) {
        if (!isExistFile(fromPath)) return;
        Bitmap src = BitmapFactory.decodeFile(fromPath);
        ColorMatrix cm = new ColorMatrix(new float[]
                {
                        1, 0, 0, 0, brightness,
                        0, 1, 0, 0, brightness,
                        0, 0, 1, 0, brightness,
                        0, 0, 0, 1, 0
                });

        Bitmap bitmap = Bitmap.createBitmap(src.getWidth(), src.getHeight(),
src.getConfig());
        Canvas canvas = new Canvas(bitmap);
        Paint paint = new Paint();
        paint.setColorFilter(new ColorMatrixColorFilter(cm));
        canvas.drawBitmap(src, 0, 0, paint);
        saveBitmap(bitmap, destPath);
    }

    public static void setBitmapFileContrast(String fromPath, String
destPath, float contrast) {
        if (!isExistFile(fromPath)) return;
        Bitmap src = BitmapFactory.decodeFile(fromPath);
        ColorMatrix cm = new ColorMatrix(new float[]
                {
                        contrast, 0, 0, 0, 0,
                        0, contrast, 0, 0, 0,
                        0, 0, contrast, 0, 0,
                        0, 0, 0, 1, 0
                });

        Bitmap bitmap = Bitmap.createBitmap(src.getWidth(), src.getHeight(),
src.getConfig());
        Canvas canvas = new Canvas(bitmap);
        Paint paint = new Paint();
        paint.setColorFilter(new ColorMatrixColorFilter(cm));
        canvas.drawBitmap(src, 0, 0, paint);

        saveBitmap(bitmap, destPath);
    }

    public static int getJpegRotate(String filePath) {
        int rotate = 0;
        try {
            ExifInterface exif = new ExifInterface(filePath);
            int iOrientation =
exif.getAttributeInt(ExifInterface.TAG_ORIENTATION, -1);

            switch (iOrientation) {
                case ExifInterface.ORIENTATION_ROTATE_90:
                    rotate = 90;
                    break;
                case ExifInterface.ORIENTATION_ROTATE_180:
                    rotate = 180;
                    break;
                case ExifInterface.ORIENTATION_ROTATE_270:
                    rotate = 270;
```

```
                break;
            }
        }
        catch (IOException e) {
            return 0;
        }

        return rotate;
    }
    public static File createNewPictureFile(Context context) {
        SimpleDateFormat date = new SimpleDateFormat("yyyyMMdd_HHmmss");
        String fileName = date.format(new Date()) + ".jpg";
        return new
File(context.getExternalFilesDir(Environment.DIRECTORY_DCIM).getAbsolutePat
h() + File.separator + fileName);
    }
}
```

Food Diary List Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import android.graphics.drawable.GradientDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.os.Parcelable;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.Animation;
import android.view.animation.ScaleAnimation;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.bumptech.glide.Glide;
import
```

```
com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import com.sdsmdg.tastytoast.TastyToast;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;


public class FoodDiaryListActivity extends AppCompatActivity {

   private FloatingActionButton fab;
   private double Position = 0;
   private boolean Selected = false;
   private double Selected_Length = 0;
   private boolean IsSelectAll = false;
   private String fontName = "";
   private String typeace = "";

   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> temp_maplist = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> Goal_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> Compltd_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> Food_List = new
ArrayList<>();

   private LinearLayout toolbar;
   private LinearLayout main;
   private ImageView back_img;
   private TextView toolbar_txt;
   private ImageView select_all_img;
   private ListView compltd_list;
   private ListView notes_list;
   private LinearLayout linrSetGoal;
   private ImageView imageview34;
   private TextView textview45;

   private Intent To_Noteview = new Intent();
   private SharedPreferences Settings;
   private SharedPreferences AllMeals;
   private AlertDialog.Builder Delete_Dialog;
   private SharedPreferences AllGoals;
   private SharedPreferences Diary;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.food_diary_list);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      fab = (FloatingActionButton) findViewById(R.id.fab);
```

```
      toolbar =  findViewById(R.id.toolbar);
      main =  findViewById(R.id.main);
      back_img =  findViewById(R.id.back_img);
      toolbar_txt = (TextView) findViewById(R.id.toolbar_txt);
      select_all_img =  findViewById(R.id.select_all_img);
      compltd_list = (ListView) findViewById(R.id.compltd_list);
      notes_list = (ListView) findViewById(R.id.notes_list);
      linrSetGoal =  findViewById(R.id.linrSetGoal);
      imageview34 =  findViewById(R.id.imageview34);
      textview45 = (TextView) findViewById(R.id.textview45);
      Settings = getSharedPreferences("Settings", Activity.MODE_PRIVATE);
      AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);
      Delete_Dialog = new AlertDialog.Builder(this);
      AllGoals = getSharedPreferences("AllGoals", Activity.MODE_PRIVATE);
      Diary = getSharedPreferences("Diary", Activity.MODE_PRIVATE);

      back_img.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            finish();
         }
      });

      select_all_img.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            selectAll();
         }
      });

      notes_list.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
         @Override
         public void onItemClick(AdapterView<?> param1, View param2, int
param3, long param4) {
            final int position = param3;
            if (Selected) {
               if (Food_List.get((int)(Food_List.size() - 1) -
position).get("Select").toString().equals("True")) {
                  Food_List.get((int)(Food_List.size() - 1) -
position).put("Select", "False");
                  Selected_Length--;
                  IsSelectAll = false;
               }
               else {
                  Food_List.get((int)(Food_List.size() - 1) -
position).put("Select", "True");
                  Selected_Length++;
               }
               refresh(Food_List);
               _Toolbar(Selected_Length);
               if (Selected_Length == 0) {
                  IsSelectAll = false;
                  select_all_img.setEnabled(true);
                  select_all_img.setVisibility(View.INVISIBLE);
                  toolbar_txt.setText("Food Diary");
                  fab(false);
                  Selected = false;

((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
               }
```

```
            }
            else {

            }
         }
      });

      notes_list.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
         @Override
         public boolean onItemLongClick(AdapterView<?> param1, View param2,
int param3, long param4) {
            final int position = param3;
            if (Selected) {
               ToastUtil.showMessage(getApplicationContext(), "Selection
Mode Is Enabled");
            }
            else {
               Selected = true;
               Food_List.get((int)(Food_List.size() - 1) -
position).put("Select", "True");
               refresh(Food_List);
               fab(true);
               Selected_Length++;
               _Toolbar(Selected_Length);
               select_all_img.setVisibility(View.VISIBLE);
               select_all_img.setEnabled(true);
            }
            return true;
         }
      });

      linrSetGoal.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {

         }
      });

      fab.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            Delete_Dialog.setTitle("Delete ?");
            Delete_Dialog.setMessage("Are You Sure To Delete
".concat(String.valueOf((long)(Selected_Length)).concat(" Selected Items ?
This Action Cannot Be Undone.")));
            Delete_Dialog.setPositiveButton("Delete", new
DialogInterface.OnClickListener() {
               @Override
               public void onClick(DialogInterface dialog, int which) {
                  Selected = false;
                  Position = Goal_List.size() - 1;
                  while(true) {
                     if (Position == -1) {
                        break;
                     }
                     else {
                        if
(Goal_List.get((int)Position).get("Select").toString().equals("True")) {
                           Goal_List.remove((int)(Position));
                        }
```

```
                        Position--;
                    }
                }
                refresh(Goal_List);
                Selected_Length = 0;
                toolbar_txt.setText("Your Goals");
                fab(false);
                select_all_img.setVisibility(View.INVISIBLE);
                saveData();
                TastyToast.makeText(getApplicationContext(),
"Successfully Deleted ", TastyToast.LENGTH_LONG, TastyToast.SUCCESS);
              }
            });
            Delete_Dialog.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
              @Override
              public void onClick(DialogInterface dialog, int which) {
                 _UnSelectAll();
              }
            });
            Delete_Dialog.create().show();
         }
      });
   }

   private void initializeLogic() {
      changeActivityFont("railway_semisolid");
      removeScrollBar(notes_list);
      getSettingsData();
      fab(false);
      select_all_img.setVisibility(View.INVISIBLE);
      select_all_img.setEnabled(false);
      if (!Diary.getString("Diary", "").equals("")) {
         Food_List = new Gson().fromJson(Diary.getString("Diary", ""), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
         notes_list.setAdapter(new Notes_listAdapter(Food_List));
         ((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
      }
      linrSetGoal.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFFFFFFFF));
      if (!Diary.getString("Diary", "").equals("")) {
         Compltd_List = new Gson().fromJson(Diary.getString("Diary", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
         compltd_list.setAdapter(new Compltd_listAdapter(Compltd_List));
         ((BaseAdapter)compltd_list.getAdapter()).notifyDataSetChanged();
      }
      if (getIntent().getStringExtra("Type").equals("All")) {
         compltd_list.setVisibility(View.GONE);
         notes_list.setVisibility(View.VISIBLE);
      }
      else {
         if (getIntent().getStringExtra("Type").equals("Completed")) {
            compltd_list.setVisibility(View.VISIBLE);
            notes_list.setVisibility(View.GONE);
         }
      }
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
```

```
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   @Override
   public void onBackPressed() {
      if (Selected) {
         _UnSelectAll();
      }
      else {
         finish();
      }
   }

   @Override
   public void onResume() {
      super.onResume();
      refresh(Food_List);
      if (!Diary.getString("Diary", "").equals("")) {
         Food_List = new Gson().fromJson(Diary.getString("Diary", ""), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      }
   }
   public void setRadiusToView (final View view, final double radius, final
String Colour) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
gd.setColor(Color.parseColor(Colour)); gd.setCornerRadius((int)radius);
view.setBackground(gd);
   }


   public void setRadiusToView (final View view, final double value) {
      view.setElevation((float)value);
   }


   public void corners (final View view, final String _color1, final String
_color2, final double msgtr, final double _n1, final double _n2, final
double _n3, final double _n4) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();

      gd.setColor(Color.parseColor(_color1));

      gd.setStroke((int)msgtr, Color.parseColor(_color2));

      gd.setCornerRadii(new
float[]{(int)_n1,(int)_n1,(int)_n2,(int)_n2,(int)_n3,(int)_n3,(int)_n4,(int
)_n4});

      view.setBackground(gd);

      view.setElevation(4);
   }
```

```
   public void Add (final String Colour, final ImageView Imageview) {
      Imageview.getDrawable().setColorFilter(Color.parseColor(Colour),
PorterDuff.Mode.SRC_IN);
   }


   public void removeScrollBar (final View view) {
      view.setVerticalScrollBarEnabled(false);
view.setHorizontalScrollBarEnabled(false);
   }


   public void getSettingsData () {

   }


   public void textSize (final TextView TextView1, final double size) {
      int j = (int) size;
      TextView1.setTextSize(j);
   }


   public void fab (final boolean visibility) {
      if (visibility) {
         fab.show();
      }
      else {
         fab.hide();
      }
   }


   public void _Toolbar (final double _Length) {

toolbar_txt.setText(String.valueOf((long)(_Length)).concat("/".concat(Strin
g.valueOf((long)(Notes_List.size()))).concat(" Selected"))));
   }


   public void _UnSelectAll () {
      Selected = false;
      Selected_Length = 0;
      toolbar_txt.setText("Your Goals");
      select_all_img.setVisibility(View.INVISIBLE);
      select_all_img.setEnabled(false);
      IsSelectAll = false;
      Position = 0;
      for(int i = 0; i < (int)(Notes_List.size()); i++) {
         if
(Notes_List.get((int)Position).get("Select").toString().equals("True")) {
            Notes_List.get((int)Position).put("Select", "False");
         }
         Position++;
      }
      refresh(Notes_List);
      fab(false);
   }
```

```
   public void refresh (final ArrayList<HashMap<String, Object>> ListMap) {
      Parcelable state =
      notes_list.onSaveInstanceState();
      notes_list.setAdapter(new Notes_listAdapter(ListMap));
      ((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
      notes_list.onRestoreInstanceState(state);
   }


   public void selectAll () {
      if (IsSelectAll) {
         _UnSelectAll();
         IsSelectAll = false;
      }
      else {
         IsSelectAll = true;
         Position = 0;
         for(int i = 0; i < (int)(Goal_List.size()); i++) {
            if
(Goal_List.get((int)Position).get("Select").toString().equals("False")) {
               Goal_List.get((int)Position).put("Select", "True");
               Selected_Length++;
            }
            Position++;
            _Toolbar(Selected_Length);
         }
         refresh(Goal_List);
      }
   }


   public void saveData () {
      temp_maplist = new Gson().fromJson(AllGoals.getString("Goals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      if (Goal_List.size() == 0) {
         if (getIntent().getStringExtra("Type").equals("All")) {
            AllGoals.edit().putString("Meals", new
Gson().toJson(Goal_List)).commit();
         }
         else {
            if (getIntent().getStringExtra("Type").equals("Completed")) {
               Position = temp_maplist.size() - 1;
               for(int i = 0; i < (int)(temp_maplist.size()); i++) {
                  if
(temp_maplist.get((int)Position).get("Completed").toString().equals("True")
) {
                     temp_maplist.remove((int)(Position));
                  }
                  Position--;
               }
               AllGoals.edit().putString("Goals", new
Gson().toJson(temp_maplist)).commit();
            }
         }
      }
      else {
         AllGoals.edit().putString("Goals", new
Gson().toJson(Goal_List)).commit();
      }
   }
```

```
   public void changeActivityFont (final String fontname) {
      fontName = "fonts/".concat(fontname.concat(".ttf"));
      overrideFonts(this,getWindow().getDecorView());
   }
   private void overrideFonts(final android.content.Context context, final
View v) {

      try {
         Typeface
         typeace = Typeface.createFromAsset(getAssets(), fontName);;
         if ((v instanceof ViewGroup)) {
            ViewGroup vg = (ViewGroup) v;
            for (int i = 0;
            i < vg.getChildCount();
            i++) {
               View child = vg.getChildAt(i);
               overrideFonts(context, child);
            }
         }
         else {
            if ((v instanceof TextView)) {
               ((TextView) v).setTypeface(typeace);
            }
            else {
               if ((v instanceof EditText )) {
                  ((EditText) v).setTypeface(typeace);
               }
               else {
                  if ((v instanceof Button)) {
                     ((Button) v).setTypeface(typeace);
                  }
               }
            }
         }
      }
      catch(Exception e)

      {

      };
   }


   public void clickAnimation (final View view) {
      ScaleAnimation fade_in = new ScaleAnimation(0.9f, 1f, 0.9f, 1f,
Animation.RELATIVE_TO_SELF, 0.5f, Animation.RELATIVE_TO_SELF, 0.7f);
      fade_in.setDuration(300);
      fade_in.setFillAfter(true);
      view.startAnimation(fade_in);
   }


   public class Compltd_listAdapter extends BaseAdapter {
      ArrayList<HashMap<String, Object>> data;
      public Compltd_listAdapter(ArrayList<HashMap<String, Object>> arr) {
         data = arr;
      }

      @Override
      public int getCount() {
```

```
            return data.size();
        }

        @Override
        public HashMap<String, Object> getItem(int index) {
            return data.get(index);
        }

        @Override
        public long getItemId(int index) {
            return index;
        }
        @Override
        public View getView(final int position, View v, ViewGroup container)
{
            LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
            View view = v;
            if (view == null) {
               view = inflater.inflate(R.layout.goal_list, null);
            }

            final LinearLayout bg =  view.findViewById(R.id.bg);
            final LinearLayout linear23 =  view.findViewById(R.id.linear23);
            final LinearLayout main =  view.findViewById(R.id.main);
            final ImageView imageview1 =  view.findViewById(R.id.imageview1);
            final LinearLayout text_lin =  view.findViewById(R.id.text_lin);
            final LinearLayout title_time_lin =
view.findViewById(R.id.title_time_lin);
            final LinearLayout linear2 =  view.findViewById(R.id.linear2);
            final LinearLayout linear5 =  view.findViewById(R.id.linear5);
            final TextView title = (TextView) view.findViewById(R.id.title);
            final LinearLayout linear22 =  view.findViewById(R.id.linear22);
            final TextView txt_Tag = (TextView)
view.findViewById(R.id.txt_Tag);
            final TextView time = (TextView) view.findViewById(R.id.time);
            final TextView date = (TextView) view.findViewById(R.id.date);
            final TextView txtStartDate = (TextView)
view.findViewById(R.id.txtStartDate);
            final TextView txtEndDate = (TextView)
view.findViewById(R.id.txtEndDate);
            final TextView txtGoalSub = (TextView)
view.findViewById(R.id.txtGoalSub);


date.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
ium.ttf"), Typeface.BOLD);

title.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansbo
ld.ttf"), Typeface.BOLD);

time.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
ium.ttf"), Typeface.BOLD);
            main.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFFFFFFFF));
            linear23.setVisibility(View.GONE);
            if (Selected) {

imageview1.setImageResource(R.drawable.ic_radio_button_off_black);
```

```
            if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).get("Select").toString().equals("True")) {
                imageview1.setVisibility(View.VISIBLE);

imageview1.setImageResource(R.drawable.ic_radio_button_on_black);
            }
            else {

imageview1.setImageResource(R.drawable.ic_radio_button_off_black);
            }
         }
         else {
            imageview1.setVisibility(View.GONE);
         }
         if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).containsKey("Goal")) {
            title.setText(Compltd_List.get((int)(Compltd_List.size() - 1) -
position).get("Goal").toString());
            if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).get("Goal").toString().length() > 28) {
                title.setText(Compltd_List.get((int)(Compltd_List.size() -
1) - position).get("Goal").toString().substring((int)(0),
(int)(28)).concat("..."));
            }
            else {

            }
         }
         if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).containsKey("Time")) {
            time.setText(Compltd_List.get((int)(Compltd_List.size() - 1) -
position).get("Time").toString());
         }
         if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).containsKey("Date")) {
            date.setText(Compltd_List.get((int)(Compltd_List.size() - 1) -
position).get("Date").toString());
         }
         if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).containsKey("Tag")) {
            txt_Tag.setText(Compltd_List.get((int)(Compltd_List.size() - 1)
- position).get("Tag").toString());
         }
         if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).containsKey("End")) {
            txtEndDate.setText(Compltd_List.get((int)(Compltd_List.size() -
1) - position).get("End").toString());
         }
         if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).containsKey("Start")) {
            txtStartDate.setText(Compltd_List.get((int)(Compltd_List.size()
- 1) - position).get("Start").toString());
         }
         if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).containsKey("Goalsub")) {
            if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).get("Goalsub").toString().equals("")) {
               txtGoalSub.setText("Null");
            }
            else {
```

```
txtGoalSub.setText(Compltd_List.get((int)(Compltd_List.size() - 1) -
position).get("Goalsub").toString());
            }
         }

         return view;
      }
   }

   public class Notes_listAdapter extends BaseAdapter {
      ArrayList<HashMap<String, Object>> data;
      public Notes_listAdapter(ArrayList<HashMap<String, Object>> arr) {
         data = arr;
      }

      @Override
      public int getCount() {
         return data.size();
      }

      @Override
      public HashMap<String, Object> getItem(int index) {
         return data.get(index);
      }

      @Override
      public long getItemId(int index) {
         return index;
      }
      @Override
      public View getView(final int position, View v, ViewGroup container)
{
         LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
         View view = v;
         if (view == null) {
            view = inflater.inflate(R.layout.saveddiary, null);
         }

         final LinearLayout linear1 =  view.findViewById(R.id.linear1);
         final LinearLayout linear4 =  view.findViewById(R.id.linear4);
         final LinearLayout linear2 =  view.findViewById(R.id.linear2);
         final LinearLayout linear3 =  view.findViewById(R.id.linear3);
         final ImageView imgFoodImg =  view.findViewById(R.id.imgFoodImg);
         final LinearLayout linear5 =  view.findViewById(R.id.linear5);
         final LinearLayout linear6 =  view.findViewById(R.id.linear6);
         final TextView txtFoodName = (TextView)
view.findViewById(R.id.txtFoodName);
         final TextView txtIngtd = (TextView)
view.findViewById(R.id.txtIngtd);
         final TextView txtDate = (TextView)
view.findViewById(R.id.txtDate);
         final TextView txtCarbo = (TextView)
view.findViewById(R.id.txtCarbo);
         final TextView txtPro = (TextView) view.findViewById(R.id.txtPro);
         final TextView txtKcal = (TextView)
view.findViewById(R.id.txtKcal);
         final TextView txtfat = (TextView) view.findViewById(R.id.txtfat);
         final TextView txtSugar = (TextView)
view.findViewById(R.id.txtSugar);
```

```
        final TextView txtSalt = (TextView)
view.findViewById(R.id.txtSalt);

        linear4.setVisibility(View.GONE);
        if (Food_List.get((int)(Food_List.size() - 1) -
position).containsKey("Img")) {

Glide.with(getApplicationContext()).load(Uri.parse(Food_List.get((int)(Food
_List.size() - 1) - position).get("Img").toString())).into(imgFoodImg);
        }
        if (Food_List.get((int)(Food_List.size() - 1) -
position).containsKey("Food")) {
            txtFoodName.setText(Food_List.get((int)(Food_List.size() - 1) -
position).get("Food").toString());
        }
        if (Food_List.get((int)(Food_List.size() - 1) -
position).containsKey("Ingredt")) {
            txtIngtd.setText(Food_List.get((int)(Food_List.size() - 1) -
position).get("Ingredt").toString());
        }
        if (Food_List.get((int)(Food_List.size() - 1) -
position).containsKey("kCal")) {
            txtKcal.setText(Food_List.get((int)(Food_List.size() - 1) -
position).get("kCal").toString());
        }
        if (Food_List.get((int)(Food_List.size() - 1) -
position).containsKey("Carbo")) {
            txtCarbo.setText(Food_List.get((int)(Food_List.size() - 1) -
position).get("Carbo").toString());
        }
        if (Food_List.get((int)(Food_List.size() - 1) -
position).containsKey("Protein")) {
            txtPro.setText(Food_List.get((int)(Food_List.size() - 1) -
position).get("Protein").toString());
        }
        if (Food_List.get((int)(Food_List.size() - 1) -
position).containsKey("Fat")) {
            txtfat.setText(Food_List.get((int)(Food_List.size() - 1) -
position).get("Fat").toString());
        }
        if (Food_List.get((int)(Food_List.size() - 1) -
position).containsKey("Salt")) {
            txtSalt.setText(Food_List.get((int)(Food_List.size() - 1) -
position).get("Salt").toString());
        }
        if (Food_List.get((int)(Food_List.size() - 1) -
position).containsKey("Date")) {
            txtDate.setText(Food_List.get((int)(Food_List.size() - 1) -
position).get("Date").toString());
        }
        linear1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                To_Noteview.setAction(Intent.ACTION_VIEW);
                To_Noteview.setClass(getApplicationContext(),
DiaryviewActivity.class);
                To_Noteview.putExtra("Food",
txtFoodName.getText().toString());
                To_Noteview.putExtra("Img", "");
                To_Noteview.putExtra("Ingredt",
txtIngtd.getText().toString());
```

```
                To_Noteview.putExtra("kCal", txtKcal.getText().toString());
                To_Noteview.putExtra("Protein",
txtPro.getText().toString());
                To_Noteview.putExtra("Fat", txtfat.getText().toString());
                To_Noteview.putExtra("Salt", txtSalt.getText().toString());
                startActivity(To_Noteview);
            }
        });

        return view;
    }
}

@Deprecated
public void showMessage(String msg) {
    Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
}

@Deprecated
public int getLocationX(View v) {
    int location[] = new int[2];
    v.getLocationInWindow(location);
    return location[0];
}

@Deprecated
public int getLocationY(View v) {
    int location[] = new int[2];
    v.getLocationInWindow(location);
    return location[1];
}

@Deprecated
public int getRandom(int min, int max) {
    Random random = new Random();
    return random.nextInt(max - min + 1) + min;
}

@Deprecated
public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
    ArrayList<Double> result = new ArrayList<Double>();
    SparseBooleanArray arr = _list.getCheckedItemPositions();
    for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
        if (arr.valueAt(iIdx))
        result.add((double)arr.keyAt(iIdx));
    }
    return result;
}

@Deprecated
public float getDip(int input) {
    return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
}

@Deprecated
public int getDisplayWidthPixels() {
    return getResources().getDisplayMetrics().widthPixels;
}
```

```
    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Goal Planner Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.content.Intent;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.widget.ListView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;
import java.util.Random;


public class GoalplannerActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.goalplanner);
        initialize(savedInstanceState);
        com.google.firebase.FirebaseApp.initializeApp(this);
        initializeLogic();
    }

    private void initialize(Bundle savedInstanceState) {
    }

    private void initializeLogic() {
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);
        switch (requestCode) {

            default:
            break;
```

```
        }
    }

    @Deprecated
    public void showMessage(String msg) {
        Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
    }

    @Deprecated
    public int getLocationX(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Goals List Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import android.graphics.drawable.GradientDrawable;
import android.os.Bundle;
import android.os.Parcelable;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.Animation;
import android.view.animation.ScaleAnimation;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.blogspot.atifsoftwares.animatoolib.Animatoo;
import
com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import com.sdsmdg.tastytoast.TastyToast;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;


public class GoalslistActivity extends AppCompatActivity {

   private FloatingActionButton fab;
   private double Position = 0;
   private boolean Selected = false;
   private double Selected_Length = 0;
   private boolean IsSelectAll = false;
   private String fontName = "";
```

```
   private String typeace = "";

   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> temp_maplist = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> Goal_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> Compltd_List = new
ArrayList<>();

   private LinearLayout toolbar;
   private LinearLayout main;
   private ImageView back_img;
   private TextView toolbar_txt;
   private ImageView select_all_img;
   private ListView compltd_list;
   private ListView notes_list;
   private LinearLayout linrSetGoal;
   private ImageView imageview34;
   private TextView textview45;

   private Intent To_Noteview = new Intent();
   private SharedPreferences Settings;
   private SharedPreferences AllMeals;
   private AlertDialog.Builder Delete_Dialog;
   private SharedPreferences AllGoals;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.goalslist);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      fab = (FloatingActionButton) findViewById(R.id.fab);

      toolbar =  findViewById(R.id.toolbar);
      main =  findViewById(R.id.main);
      back_img =  findViewById(R.id.back_img);
      toolbar_txt = (TextView) findViewById(R.id.toolbar_txt);
      select_all_img =  findViewById(R.id.select_all_img);
      compltd_list = (ListView) findViewById(R.id.compltd_list);
      notes_list = (ListView) findViewById(R.id.notes_list);
      linrSetGoal =  findViewById(R.id.linrSetGoal);
      imageview34 =  findViewById(R.id.imageview34);
      textview45 = (TextView) findViewById(R.id.textview45);
      Settings = getSharedPreferences("Settings", Activity.MODE_PRIVATE);
      AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);
      Delete_Dialog = new AlertDialog.Builder(this);
      AllGoals = getSharedPreferences("AllGoals", Activity.MODE_PRIVATE);

      back_img.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            finish();
         }
      });
```

```
      select_all_img.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            selectAll();
         }
      });

      compltd_list.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
         @Override
         public void onItemClick(AdapterView<?> param1, View param2, int
param3, long param4) {
            final int position = param3;
            ToastUtil.showMessage(getApplicationContext(), "Already
completed");
         }
      });

      notes_list.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
         @Override
         public void onItemClick(AdapterView<?> param1, View param2, int
param3, long param4) {
            final int position = param3;
            if (Selected) {
               if (Goal_List.get((int)(Goal_List.size() - 1) -
position).get("Select").toString().equals("True")) {
                  Goal_List.get((int)(Goal_List.size() - 1) -
position).put("Select", "False");
                  Selected_Length--;
                  IsSelectAll = false;
               }
               else {
                  Goal_List.get((int)(Goal_List.size() - 1) -
position).put("Select", "True");
                  Selected_Length++;
               }
               refresh(Notes_List);
               _Toolbar(Selected_Length);
               if (Selected_Length == 0) {
                  IsSelectAll = false;
                  select_all_img.setEnabled(true);
                  select_all_img.setVisibility(View.INVISIBLE);
                  toolbar_txt.setText("Your Goals");
                  fab(false);
                  Selected = false;

((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
               }
            }
            else {
               To_Noteview.setAction(Intent.ACTION_VIEW);
               To_Noteview.setClass(getApplicationContext(),
GoalviewActivity.class);
               To_Noteview.putExtra("Position",
String.valueOf((long)((Goal_List.size() - 1) - position)));
               startActivity(To_Noteview);
               finish();
            }
         }
```

```
      });

      notes_list.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
         @Override
         public boolean onItemLongClick(AdapterView<?> param1, View param2,
int param3, long param4) {
            final int position = param3;
            if (Selected) {
               ToastUtil.showMessage(getApplicationContext(), "Selection
Mode Is Enabled");
            }
            else {
               Selected = true;
               Goal_List.get((int)(Goal_List.size() - 1) -
position).put("Select", "True");
               refresh(Goal_List);
               fab(true);
               Selected_Length++;
               _Toolbar(Selected_Length);
               select_all_img.setVisibility(View.VISIBLE);
               select_all_img.setEnabled(true);
            }
            return true;
         }
      });

      linrSetGoal.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            startActivity(new Intent(GoalslistActivity.this,
AddgoalActivity.class)); Animatoo.animateFade(GoalslistActivity.this);
            clickAnimation(linrSetGoal);
         }
      });

      fab.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            Delete_Dialog.setTitle("Delete ?");
            Delete_Dialog.setMessage("Are You Sure To Delete
".concat(String.valueOf((long)(Selected_Length)).concat(" Selected Items ?
This Action Cannot Be Undone.")));
            Delete_Dialog.setPositiveButton("Delete", new
DialogInterface.OnClickListener() {
               @Override
               public void onClick(DialogInterface dialog, int which) {
                  Selected = false;
                  Position = Goal_List.size() - 1;
                  while(true) {
                     if (Position == -1) {
                        break;
                     }
                     else {
                        if
(Goal_List.get((int)Position).get("Select").toString().equals("True")) {
                           Goal_List.remove((int)(Position));
                        }
                        Position--;
                     }
                  }
```

```
                    refresh(Goal_List);
                    Selected_Length = 0;
                    toolbar_txt.setText("Your Goals");
                    fab(false);
                    select_all_img.setVisibility(View.INVISIBLE);
                    saveData();
                    TastyToast.makeText(getApplicationContext(), "Successfuly
Deleted ", TastyToast.LENGTH_LONG, TastyToast.SUCCESS);
                }
            });
            Delete_Dialog.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    _UnSelectAll();
                }
            });
            Delete_Dialog.create().show();
        }
    });
  }

  private void initializeLogic() {
    changeActivityFont("railway_semisolid");
    removeScrollBar(notes_list);
    getSettingsData();
    fab(false);
    select_all_img.setVisibility(View.INVISIBLE);
    select_all_img.setEnabled(false);
    if (!AllGoals.getString("Goals", "").equals("")) {
      Goal_List = new Gson().fromJson(AllGoals.getString("Goals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      notes_list.setAdapter(new Notes_listAdapter(Goal_List));
      ((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
    }
    linrSetGoal.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFF));
    if (!AllGoals.getString("Compltd", "").equals("")) {
      Compltd_List = new Gson().fromJson(AllGoals.getString("Compltd",
""), new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      compltd_list.setAdapter(new Compltd_listAdapter(Compltd_List));
      ((BaseAdapter)compltd_list.getAdapter()).notifyDataSetChanged();
    }
    if (getIntent().getStringExtra("Type").equals("All")) {
      compltd_list.setVisibility(View.GONE);
      notes_list.setVisibility(View.VISIBLE);
    }
    else {
      if (getIntent().getStringExtra("Type").equals("Completed")) {
        compltd_list.setVisibility(View.VISIBLE);
        notes_list.setVisibility(View.GONE);
      }
    }
  }

  @Override
  protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode) {
```

```
            default:
            break;
        }
    }

    @Override
    public void onBackPressed() {
        if (Selected) {
            _UnSelectAll();
        }
        else {
            finish();
        }
    }

    @Override
    public void onResume() {
        super.onResume();
        refresh(Goal_List);
        if (!AllGoals.getString("Goals", "").equals("")) {
            Goal_List = new Gson().fromJson(AllGoals.getString("Goals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
        }
    }
    public void setRadiusToView (final View view, final double radius, final
String Colour) {
        android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
gd.setColor(Color.parseColor(Colour)); gd.setCornerRadius((int)radius);
view.setBackground(gd);
    }


    public void setRadiusToView (final View view, final double value) {
        view.setElevation((float)value);
    }


    public void corners (final View view, final String _color1, final String
_color2, final double msgtr, final double _n1, final double _n2, final
double _n3, final double _n4) {
        android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();

        gd.setColor(Color.parseColor(_color1));

        gd.setStroke((int)msgtr, Color.parseColor(_color2));

        gd.setCornerRadii(new
float[]{(int)_n1,(int)_n1,(int)_n2,(int)_n2,(int)_n3,(int)_n3,(int)_n4,(int
)_n4});

        view.setBackground(gd);

        view.setElevation(4);
    }


    public void Add (final String Colour, final ImageView Imageview) {
        Imageview.getDrawable().setColorFilter(Color.parseColor(Colour),
```

```
PorterDuff.Mode.SRC_IN);
    }


    public void removeScrollBar (final View view) {
        view.setVerticalScrollBarEnabled(false);
view.setHorizontalScrollBarEnabled(false);
    }


    public void getSettingsData () {

    }


    public void textSize (final TextView TextView1, final double size) {
        int j = (int) size;
        TextView1.setTextSize(j);
    }


    public void fab (final boolean visibility) {
        if (visibility) {
            fab.show();
        }
        else {
            fab.hide();
        }
    }


    public void _Toolbar (final double _Length) {

toolbar_txt.setText(String.valueOf((long)(_Length)).concat("/".concat(Strin
g.valueOf((long)(Notes_List.size())).concat(" Selected"))));
    }


    public void _UnSelectAll () {
        Selected = false;
        Selected_Length = 0;
        toolbar_txt.setText("Your Goals");
        select_all_img.setVisibility(View.INVISIBLE);
        select_all_img.setEnabled(false);
        IsSelectAll = false;
        Position = 0;
        for(int i = 0; i < (int)(Notes_List.size()); i++) {
            if
(Notes_List.get((int)Position).get("Select").toString().equals("True")) {
                Notes_List.get((int)Position).put("Select", "False");
            }
            Position++;
        }
        refresh(Notes_List);
        fab(false);
    }


    public void refresh (final ArrayList<HashMap<String, Object>> ListMap) {
        Parcelable state =
        notes_list.onSaveInstanceState();
```

```
      notes_list.setAdapter(new Notes_listAdapter(ListMap));
      ((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
      notes_list.onRestoreInstanceState(state);
   }


   public void selectAll () {
      if (IsSelectAll) {
         _UnSelectAll();
         IsSelectAll = false;
      }
      else {
         IsSelectAll = true;
         Position = 0;
         for(int i = 0; i < (int)(Goal_List.size()); i++) {
            if
(Goal_List.get((int)Position).get("Select").toString().equals("False")) {
               Goal_List.get((int)Position).put("Select", "True");
               Selected_Length++;
            }
            Position++;
            _Toolbar(Selected_Length);
         }
         refresh(Goal_List);
      }
   }


   public void saveData () {
      temp_maplist = new Gson().fromJson(AllGoals.getString("Goals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      if (Goal_List.size() == 0) {
         if (getIntent().getStringExtra("Type").equals("All")) {
            AllGoals.edit().putString("Meals", new
Gson().toJson(Goal_List)).commit();
         }
         else {
            if (getIntent().getStringExtra("Type").equals("Completed")) {
               Position = temp_maplist.size() - 1;
               for(int i = 0; i < (int)(temp_maplist.size()); i++) {
                  if
(temp_maplist.get((int)Position).get("Completed").toString().equals("True")
) {
                     temp_maplist.remove((int)(Position));
                  }
                  Position--;
               }
               AllGoals.edit().putString("Goals", new
Gson().toJson(temp_maplist)).commit();
            }
         }
      }
      else {
         AllGoals.edit().putString("Goals", new
Gson().toJson(Goal_List)).commit();
      }
   }


   public void changeActivityFont (final String fontname) {
      fontName = "fonts/".concat(fontname.concat(".ttf"));
```

```
      overrideFonts(this,getWindow().getDecorView());
   }
   private void overrideFonts(final android.content.Context context, final
View v) {

      try {
         Typeface
         typeace = Typeface.createFromAsset(getAssets(), fontName);;
         if ((v instanceof ViewGroup)) {
            ViewGroup vg = (ViewGroup) v;
            for (int i = 0;
            i < vg.getChildCount();
            i++) {
               View child = vg.getChildAt(i);
               overrideFonts(context, child);
            }
         }
         else {
            if ((v instanceof TextView)) {
               ((TextView) v).setTypeface(typeace);
            }
            else {
               if ((v instanceof EditText )) {
                  ((EditText) v).setTypeface(typeace);
               }
               else {
                  if ((v instanceof Button)) {
                     ((Button) v).setTypeface(typeace);
                  }
               }
            }
         }
      }
      catch(Exception e)

      {

      };
   }


   public void clickAnimation (final View view) {
      ScaleAnimation fade_in = new ScaleAnimation(0.9f, 1f, 0.9f, 1f,
Animation.RELATIVE_TO_SELF, 0.5f, Animation.RELATIVE_TO_SELF, 0.7f);
      fade_in.setDuration(300);
      fade_in.setFillAfter(true);
      view.startAnimation(fade_in);
   }


   public class Compltd_listAdapter extends BaseAdapter {
      ArrayList<HashMap<String, Object>> data;
      public Compltd_listAdapter(ArrayList<HashMap<String, Object>> arr) {
         data = arr;
      }

      @Override
      public int getCount() {
         return data.size();
      }
```

```
        @Override
        public HashMap<String, Object> getItem(int index) {
            return data.get(index);
        }

        @Override
        public long getItemId(int index) {
            return index;
        }
        @Override
        public View getView(final int position, View v, ViewGroup container)
{
            LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
            View view = v;
            if (view == null) {
               view = inflater.inflate(R.layout.goal_list, null);
            }

            final LinearLayout bg =  view.findViewById(R.id.bg);
            final LinearLayout linear23 =  view.findViewById(R.id.linear23);
            final LinearLayout main =  view.findViewById(R.id.main);
            final ImageView imageview1 =  view.findViewById(R.id.imageview1);
            final LinearLayout text_lin =  view.findViewById(R.id.text_lin);
            final LinearLayout title_time_lin =
view.findViewById(R.id.title_time_lin);
            final LinearLayout linear2 =  view.findViewById(R.id.linear2);
            final LinearLayout linear5 =  view.findViewById(R.id.linear5);
            final TextView title = (TextView) view.findViewById(R.id.title);
            final LinearLayout linear22 =  view.findViewById(R.id.linear22);
            final TextView txt_Tag = (TextView)
view.findViewById(R.id.txt_Tag);
            final TextView time = (TextView) view.findViewById(R.id.time);
            final TextView date = (TextView) view.findViewById(R.id.date);
            final TextView txtStartDate = (TextView)
view.findViewById(R.id.txtStartDate);
            final TextView txtEndDate = (TextView)
view.findViewById(R.id.txtEndDate);
            final TextView txtGoalSub = (TextView)
view.findViewById(R.id.txtGoalSub);


date.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
ium.ttf"), Typeface.NORMAL);

title.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansbo
ld.ttf"), Typeface.NORMAL);

time.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
ium.ttf"), Typeface.NORMAL);
            main.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFFFFFF));
            linear23.setVisibility(View.GONE);
            if (Selected) {

imageview1.setImageResource(R.drawable.ic_radio_button_off_black);
               if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).get("Select").toString().equals("True")) {
                  imageview1.setVisibility(View.VISIBLE);
```

```
imageview1.setImageResource(R.drawable.ic_radio_button_on_black);
            }
            else {

imageview1.setImageResource(R.drawable.ic_radio_button_off_black);
            }
         }
         else {
            imageview1.setVisibility(View.GONE);
         }
         if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).containsKey("Goal")) {
            title.setText(Compltd_List.get((int)(Compltd_List.size() - 1) -
position).get("Goal").toString());
            if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).get("Goal").toString().length() > 28) {
               title.setText(Compltd_List.get((int)(Compltd_List.size() -
1) - position).get("Goal").toString().substring((int)(0),
(int)(28)).concat("..."));
            }
            else {

            }
         }
         if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).containsKey("Time")) {
            time.setText(Compltd_List.get((int)(Compltd_List.size() - 1) -
position).get("Time").toString());
         }
         if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).containsKey("Date")) {
            date.setText(Compltd_List.get((int)(Compltd_List.size() - 1) -
position).get("Date").toString());
         }
         if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).containsKey("Tag")) {
            txt_Tag.setText(Compltd_List.get((int)(Compltd_List.size() - 1)
- position).get("Tag").toString());
         }
         if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).containsKey("End")) {
            txtEndDate.setText(Compltd_List.get((int)(Compltd_List.size() -
1) - position).get("End").toString());
         }
         if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).containsKey("Start")) {
            txtStartDate.setText(Compltd_List.get((int)(Compltd_List.size()
- 1) - position).get("Start").toString());
         }
         if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).containsKey("Goalsub")) {
            if (Compltd_List.get((int)(Compltd_List.size() - 1) -
position).get("Goalsub").toString().equals("")) {
               txtGoalSub.setText("Null");
            }
            else {

txtGoalSub.setText(Compltd_List.get((int)(Compltd_List.size() - 1) -
position).get("Goalsub").toString());
            }
```

```
         }

         return view;
      }
   }

   public class Notes_listAdapter extends BaseAdapter {
      ArrayList<HashMap<String, Object>> data;
      public Notes_listAdapter(ArrayList<HashMap<String, Object>> arr) {
         data = arr;
      }

      @Override
      public int getCount() {
         return data.size();
      }

      @Override
      public HashMap<String, Object> getItem(int index) {
         return data.get(index);
      }

      @Override
      public long getItemId(int index) {
         return index;
      }
      @Override
      public View getView(final int position, View v, ViewGroup container)
{
         LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
         View view = v;
         if (view == null) {
            view = inflater.inflate(R.layout.goal_list, null);
         }

         final LinearLayout bg =  view.findViewById(R.id.bg);
         final LinearLayout linear23 =  view.findViewById(R.id.linear23);
         final LinearLayout main =  view.findViewById(R.id.main);
         final ImageView imageview1 =  view.findViewById(R.id.imageview1);
         final LinearLayout text_lin =  view.findViewById(R.id.text_lin);
         final LinearLayout title_time_lin =
view.findViewById(R.id.title_time_lin);
         final LinearLayout linear2 =  view.findViewById(R.id.linear2);
         final LinearLayout linear5 =  view.findViewById(R.id.linear5);
         final TextView title = (TextView) view.findViewById(R.id.title);
         final LinearLayout linear22 =  view.findViewById(R.id.linear22);
         final TextView txt_Tag = (TextView)
view.findViewById(R.id.txt_Tag);
         final TextView time = (TextView) view.findViewById(R.id.time);
         final TextView date = (TextView) view.findViewById(R.id.date);
         final TextView txtStartDate = (TextView)
view.findViewById(R.id.txtStartDate);
         final TextView txtEndDate = (TextView)
view.findViewById(R.id.txtEndDate);
         final TextView txtGoalSub = (TextView)
view.findViewById(R.id.txtGoalSub);


date.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
```

```
ium.ttf"), Typeface.NORMAL);

title.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansbo
ld.ttf"), Typeface.NORMAL);

time.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
ium.ttf"), Typeface.NORMAL);
        main.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFF));
        linear23.setVisibility(View.GONE);
        if (Selected) {

imageview1.setImageResource(R.drawable.ic_radio_button_off_black);
            if (Goal_List.get((int)(Goal_List.size() - 1) -
position).get("Select").toString().equals("True")) {
                imageview1.setVisibility(View.VISIBLE);

imageview1.setImageResource(R.drawable.ic_radio_button_on_black);
            }
            else {

imageview1.setImageResource(R.drawable.ic_radio_button_off_black);
            }
        }
        else {
            imageview1.setVisibility(View.GONE);
        }
        if (Goal_List.get((int)(Goal_List.size() - 1) -
position).containsKey("Goal")) {
            title.setText(Goal_List.get((int)(Goal_List.size() - 1) -
position).get("Goal").toString());
            if (Goal_List.get((int)(Goal_List.size() - 1) -
position).get("Goal").toString().length() > 28) {
                title.setText(Goal_List.get((int)(Goal_List.size() - 1) -
position).get("Goal").toString().substring((int)(0),
(int)(28)).concat("..."));
            }
            else {

            }
        }
        if (Goal_List.get((int)(Goal_List.size() - 1) -
position).containsKey("Time")) {
            time.setText(Goal_List.get((int)(Goal_List.size() - 1) -
position).get("Time").toString());
        }
        if (Goal_List.get((int)(Goal_List.size() - 1) -
position).containsKey("Date")) {
            date.setText(Goal_List.get((int)(Goal_List.size() - 1) -
position).get("Date").toString());
        }
        if (Goal_List.get((int)(Goal_List.size() - 1) -
position).containsKey("Tag")) {
            txt_Tag.setText(Goal_List.get((int)(Goal_List.size() - 1) -
position).get("Tag").toString());
        }
        if (Goal_List.get((int)(Goal_List.size() - 1) -
position).containsKey("End")) {
            txtEndDate.setText(Goal_List.get((int)(Goal_List.size() - 1) -
position).get("End").toString());
```

```
         }
         if (Goal_List.get((int)(Goal_List.size() - 1) -
position).containsKey("Start")) {
            txtStartDate.setText(Goal_List.get((int)(Goal_List.size() - 1)
- position).get("Start").toString());
         }
         if (Goal_List.get((int)(Goal_List.size() - 1) -
position).containsKey("Goalsub")) {
            if (Goal_List.get((int)(Goal_List.size() - 1) -
position).get("Goalsub").toString().equals("")) {
               txtGoalSub.setText("Null");
            }
            else {
               txtGoalSub.setText(Goal_List.get((int)(Goal_List.size() - 1)
- position).get("Goalsub").toString());
            }
         }

         return view;
      }
   }

   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }

   @Deprecated
   public int getRandom(int min, int max) {
      Random random = new Random();
      return random.nextInt(max - min + 1) + min;
   }

   @Deprecated
   public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
      ArrayList<Double> result = new ArrayList<Double>();
      SparseBooleanArray arr = _list.getCheckedItemPositions();
      for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
         if (arr.valueAt(iIdx))
         result.add((double)arr.keyAt(iIdx));
      }
      return result;
   }

   @Deprecated
```

```
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Goal View Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.animation.ObjectAnimator;
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import android.graphics.drawable.GradientDrawable;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.AccelerateInterpolator;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import com.sdsmdg.tastytoast.TastyToast;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Random;
```

```
import java.util.Timer;
import java.util.TimerTask;


public class GoalviewActivity extends AppCompatActivity {
   private Timer timer = new Timer();

   private String Title = "";
   private String Note = "";
   private HashMap<String, Object> Get_Note = new HashMap<>();
   private double Position = 0;
   private String fontName = "";
   private String typeace = "";
   private String Method = "";
   private String Ingredients = "";
   private String Calories = "";
   private String Hours = "";
   private String Minutes = "";
   private String classes = "";
   private String Serving = "";
   private String Goalsu = "";
   private String Goal = "";
   private String Start = "";
   private String End = "";
   private String Tag = "";
   private HashMap<String, Object> Get_Goals = new HashMap<>();
   private String Date = "";
   private HashMap<String, Object> AddCompltd = new HashMap<>();

   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> Goal_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> Compltd_List = new
ArrayList<>();

   private LinearLayout toolbar;
   private LinearLayout main;
   private LinearLayout check_lin;
   private ImageView back;
   private LinearLayout linear4;
   private TextView note;
   private LinearLayout linear7;
   private LinearLayout linear5;
   private LinearLayout linear6;
   private TextView title;
   private TextView textview3;
   private TextView txtToBeComp;
   private LinearLayout linear9;
   private LinearLayout linear11;
   private LinearLayout linear10;
   private ImageView imageview1;
   private LinearLayout linrBar;
   private LinearLayout linear17;
   private LinearLayout linear15;
   private TextView textview4;
   private LinearLayout linear16;
   private TextView txtGOalStart;
   private TextView txtDateEntered;
   private TextView txtEnd;
   private CheckBox importantcheckbox;
```

```
private TextView characters;

private TextToSpeech NoteSpeak;
private SharedPreferences Settings;
private SharedPreferences AllMeals;
private Intent Note_Edit = new Intent();
private SharedPreferences AllGoals;
private Calendar cal = Calendar.getInstance();
private ObjectAnimator anime = new ObjectAnimator();
private TimerTask k;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.goalview);
    initialize(savedInstanceState);
    com.google.firebase.FirebaseApp.initializeApp(this);
    initializeLogic();
}

private void initialize(Bundle savedInstanceState) {
    toolbar =  findViewById(R.id.toolbar);
    main =  findViewById(R.id.main);
    check_lin =  findViewById(R.id.check_lin);
    back =  findViewById(R.id.back);
    linear4 =  findViewById(R.id.linear4);
    note = (TextView) findViewById(R.id.note);
    linear7 =  findViewById(R.id.linear7);
    linear5 =  findViewById(R.id.linear5);
    linear6 =  findViewById(R.id.linear6);
    title = (TextView) findViewById(R.id.title);
    textview3 = (TextView) findViewById(R.id.textview3);
    txtToBeComp = (TextView) findViewById(R.id.txtToBeComp);
    linear9 =  findViewById(R.id.linear9);
    linear11 =  findViewById(R.id.linear11);
    linear10 =  findViewById(R.id.linear10);
    imageview1 =  findViewById(R.id.imageview1);
    linrBar =  findViewById(R.id.linrBar);
    linear17 =  findViewById(R.id.linear17);
    linear15 =  findViewById(R.id.linear15);
    textview4 = (TextView) findViewById(R.id.textview4);
    linear16 =  findViewById(R.id.linear16);
    txtGOalStart = (TextView) findViewById(R.id.txtGOalStart);
    txtDateEntered = (TextView) findViewById(R.id.txtDateEntered);
    txtEnd = (TextView) findViewById(R.id.txtEnd);
    importantcheckbox = (CheckBox) findViewById(R.id.importantcheckbox);
    characters = (TextView) findViewById(R.id.characters);
    NoteSpeak = new TextToSpeech(getApplicationContext(), null);
    Settings = getSharedPreferences("Settings", Activity.MODE_PRIVATE);
    AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);
    AllGoals = getSharedPreferences("AllGoals", Activity.MODE_PRIVATE);

    back.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            cal = Calendar.getInstance();
            if (importantcheckbox.isChecked()) {
                AddCompltd = new HashMap<>();
                AddCompltd.put("Goal", Goal);
                AddCompltd.put("GoalSub", Goalsu);
                AddCompltd.put("Tag", Tag);
```

```
                AddCompltd.put("Date", new SimpleDateFormat("dd
MMM").format(cal.getTime()));
                AddCompltd.put("Time", new SimpleDateFormat("hh:mm
a").format(cal.getTime()));
                AddCompltd.put("Start", Start);
                AddCompltd.put("End", End);
                Compltd_List.add(AddCompltd);
                AllGoals.edit().putString("Compltd", new
Gson().toJson(Compltd_List)).commit();
                TastyToast.makeText(getApplicationContext(),
"Congratulations You've completed your goal", TastyToast.LENGTH_LONG,
TastyToast.SUCCESS);
                finish();
            }
            else {
                finish();
            }
        }
    });

    importantcheckbox.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton param1, boolean
param2)  {
            final boolean _isChecked = param2;
            if (_isChecked) {
                anime.setTarget(linear17);
                anime.setPropertyName("translationY");
                anime.setFloatValues((float)(-300.0d), (float)(-60));
                anime.setDuration((int)(1000));
                anime.setInterpolator(new AccelerateInterpolator());
                anime.start();
            }
            else {
                anime.setTarget(linear17);
                anime.setPropertyName("translationY");
                anime.setFloatValues((float)(-60), (float)(-300));
                anime.setDuration((int)(1000));
                anime.setInterpolator(new AccelerateInterpolator());
                anime.start();
            }
        }
    });
  }

  private void initializeLogic() {
    linrBar.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b, int c, int d) {
this.setCornerRadius(a); this.setStroke(b, c); this.setColor(d); return
this; } }.getIns((int)60, (int)0, 0xFFC5CAE9, 0xFF3F51B6));
    linear17.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b, int c, int d) {
this.setCornerRadius(a); this.setStroke(b, c); this.setColor(d); return
this; } }.getIns((int)100, (int)14, 0xFFFFFFFF, 0xFF3F51B6));
    linear17.setElevation((float)10);
    linear15.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)20, 0xFFFFFFFF));
    linear16.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
```

```
this.setColor(b); return this; } }.getIns((int)20, 0xFFFFFFFF));
      ONCREATE();
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   @Override
   public void onStart() {
      super.onStart();
      _GetNote();
   }

   @Override
   public void onBackPressed() {
      back.performClick();
   }
   public void Typeface () {

   }


   public void ONCREATE () {
      Position =
Double.parseDouble(getIntent().getStringExtra("Position"));
      Typeface();
      _GetNote();
      title.setText(Goal);
      txtToBeComp.setText(Goalsu);
      txtGOalStart.setText("Goal Start: ".concat(Start));
      txtEnd.setText("Goal Ends: ".concat(End));
      txtDateEntered.setText("Date Entered: ".concat(Date));
      getSettingsData();
   }


   public void Add (final String Colour, final ImageView Imageview) {
      Imageview.getDrawable().setColorFilter(Color.parseColor(Colour),
PorterDuff.Mode.SRC_IN);
   }


   public void getSettingsData () {

   }


   public void _detectLinks (final TextView _txt_linkify) {
      _txt_linkify.setClickable(true);
      android.text.util.Linkify.addLinks(_txt_linkify,
android.text.util.Linkify.ALL);
      _txt_linkify.setLinkTextColor(Color.parseColor("#FF3770FD"));
      _txt_linkify.setLinksClickable(true);
```

```
    }


    public void textSize (final TextView TextView1, final double size) {
        int j = (int) size;
        TextView1.setTextSize(j);
    }


    public void _GetNote () {
        if (!AllGoals.getString("Goals", "").equals("")) {
            Goal_List = new Gson().fromJson(AllGoals.getString("Goals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
            Get_Goals = Goal_List.get((int)Position);
            Goal = Get_Goals.get("Goal").toString();
            Goalsu = Get_Goals.get("GoalSub").toString();
            Start = Get_Goals.get("Start").toString();
            End = Get_Goals.get("End").toString();
            Tag = Get_Goals.get("Tag").toString();
            Date = Get_Goals.get("Date").toString();
        }
    }


    public void changeActivityFont (final String fontname) {
        fontName = "fonts/".concat(fontname.concat(".ttf"));
        overrideFonts(this,getWindow().getDecorView());
    }
    private void overrideFonts(final android.content.Context context, final
View v) {

        try {
            Typeface
            typeace = Typeface.createFromAsset(getAssets(), fontName);;
            if ((v instanceof ViewGroup)) {
                ViewGroup vg = (ViewGroup) v;
                for (int i = 0;
                i < vg.getChildCount();
                i++) {
                    View child = vg.getChildAt(i);
                    overrideFonts(context, child);
                }
            }
            else {
                if ((v instanceof TextView)) {
                    ((TextView) v).setTypeface(typeace);
                }
                else {
                    if ((v instanceof EditText )) {
                        ((EditText) v).setTypeface(typeace);
                    }
                    else {
                        if ((v instanceof Button)) {
                            ((Button) v).setTypeface(typeace);
                        }
                    }
                }
            }
        }
        catch(Exception e)
```

```
        {

        };
    }


    @Deprecated
    public void showMessage(String msg) {
        Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
    }

    @Deprecated
    public int getLocationX(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Main Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.Manifest;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import android.graphics.drawable.GradientDrawable;
import android.os.Build;
import android.os.Bundle;
import android.os.Parcelable;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.view.WindowManager;
import android.view.animation.Animation;
import android.view.animation.ScaleAnimation;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.airbnb.lottie.LottieAnimationView;
import com.blogspot.atifsoftwares.animatoolib.Animatoo;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
```

```
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.GenericTypeIndicator;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Random;
import java.util.Timer;
import java.util.TimerTask;

import de.hdodenhof.circleimageview.CircleImageView;


public class MainActivity extends AppCompatActivity {
   private Timer timer = new Timer();
   private FirebaseDatabase firebase = FirebaseDatabase.getInstance();

   private String fontName = "";
   private HashMap<String, Object> user = new HashMap<>();
   private String name1 = "";
   private double show = 0;
   private String weights = "";
   private String email = "";
   private String height = "";
   private double Position = 0;
   private HashMap<String, Object> GetNotes = new HashMap<>();
   private HashMap<String, Object> GetGoals = new HashMap<>();

   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> meal_list = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> Goal_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> Important_Notes_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> Important_Goal_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> Compltd_List = new
ArrayList<>();


   private ScrollView vscroll1;
   private LinearLayout fragment2;
   private LinearLayout fragment4;
   private LinearLayout fragment3;
   private LinearLayout linrSearchBtn;
   private ImageView imageview1;
   private TextView textview1;
   private TextView txt_username;
   private LinearLayout linear8;
   private TextView txtTotalMeal;
   private TextView textview8;
   private TextView textview9;
   private LinearLayout linear17;
   private TextView txtWeekNum;
   private TextView textview15;
   private TextView textview16;
```

```
    private TextView textview18;
    private TextView title;
    private TextView time;
    private LinearLayout linrCreateNewMealPlan;
    private ListView listview2;
    private ImageView imageview38;
    private TextView textview50;
    private LottieAnimationView lottie1;
    private TextView txtDate;
    private TextView txtDay;
    private TextView txtCurrentWeight;
    private ScrollView vscroll4;
    private LinearLayout linrDiary;
    private TextView txtFatFigure;
    private ProgressBar progressbar3;
    private ProgressBar progressbar4;
    private TextView txtCalorieFigure;
    private ProgressBar progressbar2;
    private TextView txtProteinFigure;
    private LinearLayout linrCreateDiary;
    private CircleImageView circleimageview2;
    private ScrollView vscroll3;
    private LinearLayout linrGraph;
    private ProgressBar progressbar1;
    private LinearLayout linrAllGoals;
    private LinearLayout linrCompltdGoals;
    private TextView txtAllGoalsLength;
    private TextView txtCompletedLength;
    private LinearLayout linrBtnHome;
    private LinearLayout linrBtnGoal;
    private LinearLayout linrFoodDiary;
    private LinearLayout linrBtnMore;
    private LinearLayout linrSeeAll;
    private LinearLayout linrOpenRecent;
    private LinearLayout linear12;
    private LinearLayout linear15;
    private LinearLayout linear29;


    private Intent go = new Intent();
    private FirebaseAuth auth;
    private OnCompleteListener<Void> auth_updateEmailListener;
    private OnCompleteListener<Void> auth_updatePasswordListener;
    private OnCompleteListener<Void> auth_emailVerificationSentListener;
    private OnCompleteListener<Void> auth_deleteUserListener;
    private OnCompleteListener<Void> auth_updateProfileListener;
    private OnCompleteListener<AuthResult> auth_phoneAuthListener;
    private OnCompleteListener<AuthResult> auth_googleSignInListener;
    private OnCompleteListener<AuthResult> auth_create_user_listener;
    private OnCompleteListener<AuthResult> authsign_in_listener;
    private OnCompleteListener<Void> _auth_reset_password_listener;
    private Intent register = new Intent();
    private SharedPreferences name;
    private DatabaseReference sa_db = firebase.getReference("sa_db");
    private ChildEventListener sa_db_child_listener;
    private AlertDialog.Builder notify;
    private Intent weight = new Intent();
    private Intent To_Notes = new Intent();
    private SharedPreferences AllMeals;
    private Calendar cal2 = Calendar.getInstance();
    private TimerTask t;
```

```
   private SharedPreferences week;
   private SharedPreferences userdetail;
   private Intent mealtype = new Intent();
   private Intent mealsplanner = new Intent();
   private SharedPreferences meals;
   private Intent to_mealview = new Intent();
   private SharedPreferences today;
   private Intent goal = new Intent();
   private SharedPreferences AllGoals;
   private Intent diary = new Intent();
   private SharedPreferences protein;
   private SharedPreferences calories;
   private SharedPreferences fat;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_DENIED) {
         ActivityCompat.requestPermissions(this, new String[]
{Manifest.permission.READ_EXTERNAL_STORAGE}, 1000);
      }
      else {
         initializeLogic();
      }
   }

   @Override
   public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
      super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
      if (requestCode == 1000) {
         initializeLogic();
      }
   }

   private void initialize(Bundle savedInstanceState) {
      linear29 = findViewById(R.id.linear29);
      linear15 = findViewById(R.id.linear15);
      linrOpenRecent = findViewById(R.id.linrOpenRecent);
      linrSeeAll = findViewById(R.id.linrSeeAll);
      vscroll1 =  findViewById(R.id.vscroll1);
      linear12 = findViewById(R.id.linear12);
      fragment2 =  findViewById(R.id.fragment2);
      fragment4 =  findViewById(R.id.fragment4);
      fragment3 =  findViewById(R.id.fragment3);
      linrSearchBtn =  findViewById(R.id.linrSearchBtn);
      imageview1 =  findViewById(R.id.imageview1);
      textview1 = (TextView) findViewById(R.id.textview1);
      txt_username = (TextView) findViewById(R.id.txt_username);
      linear8 =  findViewById(R.id.linear8);
      txtTotalMeal = (TextView) findViewById(R.id.txtTotalMeal);
      textview8 = (TextView) findViewById(R.id.textview8);
      linrCompltdGoals = findViewById(R.id.linrCompltdGoals);
      textview9 = (TextView) findViewById(R.id.textview9);
      linear17 =  findViewById(R.id.linear17);
```

```
        txtWeekNum = (TextView) findViewById(R.id.txtWeekNum);
        textview15 = (TextView) findViewById(R.id.textview15);
        textview16 = (TextView) findViewById(R.id.textview16);
        textview18 = (TextView) findViewById(R.id.textview18);
        title = (TextView) findViewById(R.id.title);
        time = (TextView) findViewById(R.id.time);
        linrCreateNewMealPlan =  findViewById(R.id.linrCreateNewMealPlan);
        listview2 = (ListView) findViewById(R.id.listview2);
        imageview38 =  findViewById(R.id.imageview38);
        textview50 = (TextView) findViewById(R.id.textview50);
        lottie1 = (LottieAnimationView) findViewById(R.id.lottie1);
        txtDate = (TextView) findViewById(R.id.txtDate);
        txtDay = (TextView) findViewById(R.id.txtDay);
        txtCurrentWeight = (TextView) findViewById(R.id.txtCurrentWeight);
        vscroll4 =  findViewById(R.id.vscroll4);
        linrDiary =  findViewById(R.id.linrDiary);
        txtFatFigure = (TextView) findViewById(R.id.txtFatFigure);
        progressbar3 = (ProgressBar) findViewById(R.id.progressbar3);
        progressbar4 = (ProgressBar) findViewById(R.id.progressbar4);
        txtCalorieFigure = (TextView) findViewById(R.id.txtCalorieFigure);
        progressbar2 = (ProgressBar) findViewById(R.id.progressbar2);
        txtProteinFigure = (TextView) findViewById(R.id.txtProteinFigure);
        linrCreateDiary =  findViewById(R.id.linrCreateDiary);
        circleimageview2 = (CircleImageView)
findViewById(R.id.circleimageview2);
        vscroll3 =  findViewById(R.id.vscroll3);
        linrGraph =  findViewById(R.id.linrGraph);
        progressbar1 = (ProgressBar) findViewById(R.id.progressbar1);
        linrAllGoals =  findViewById(R.id.linrAllGoals);
        txtAllGoalsLength = (TextView) findViewById(R.id.txtAllGoalsLength);
        txtCompletedLength = (TextView)
findViewById(R.id.txtCompletedLength);
        linrBtnHome =  findViewById(R.id.linrBtnHome);
        linrBtnGoal =  findViewById(R.id.linrBtnGoal);
        linrFoodDiary =  findViewById(R.id.linrFoodDiary);
        linrBtnMore =  findViewById(R.id.linrBtnMore);
        auth = FirebaseAuth.getInstance();
        name = getSharedPreferences("name", Activity.MODE_PRIVATE);
        notify = new AlertDialog.Builder(this);
        AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);
        week = getSharedPreferences("week", Activity.MODE_PRIVATE);
        userdetail = getSharedPreferences("userdetail",
Activity.MODE_PRIVATE);
        meals = getSharedPreferences("meals", Activity.MODE_PRIVATE);
        today = getSharedPreferences("today", Activity.MODE_PRIVATE);
        AllGoals = getSharedPreferences("AllGoals", Activity.MODE_PRIVATE);
        protein = getSharedPreferences("protein", Activity.MODE_PRIVATE);
        calories = getSharedPreferences("calories", Activity.MODE_PRIVATE);
        fat = getSharedPreferences("fat", Activity.MODE_PRIVATE);

        linrSeeAll.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                To_Notes.setAction(Intent.ACTION_VIEW);
                To_Notes.setClass(getApplicationContext(),
MealsListActivity.class);
                To_Notes.putExtra("Data", new Gson().toJson(Notes_List));
                To_Notes.putExtra("Type", "All");
                startActivity(To_Notes);
            }
        });
```

```
      linrOpenRecent.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            To_Notes.setAction(Intent.ACTION_VIEW);
            To_Notes.setClass(getApplicationContext(),
MealsListActivity.class);
            To_Notes.putExtra("Data", new Gson().toJson(Notes_List));
            To_Notes.putExtra("Type", "All");
            startActivity(To_Notes);
            _fade_translate();
         }
      });

      linrSearchBtn.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            startActivity(new Intent(MainActivity.this,
SearchscanmealActivity.class)); Animatoo.animateFade(MainActivity.this);
            clickAnimation(linrSearchBtn);
         }
      });

      imageview1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            startActivity(new Intent(MainActivity.this,
ProfileActivity.class)); Animatoo.animateFade(MainActivity.this);
            clickAnimation(imageview1);
         }
      });

      linear8.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            startActivity(new Intent(MainActivity.this,
MealsActivity.class)); Animatoo.animateFade(MainActivity.this);
            clickAnimation(linear8);
         }
      });

      linear12.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            linrBtnGoal.performClick();
         }
      });

      linear15.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            weight.setClass(getApplicationContext(),
WeightLogsActivity.class);
            startActivity(weight);
         }
      });

      linear17.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            startActivity(new Intent(MainActivity.this,
```

```
MainrecipeActivity.class)); Animatoo.animateFade(MainActivity.this);
            clickAnimation(linear17);
         }
      });

      linear29.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            mealsplanner.setAction(Intent.ACTION_VIEW);
            mealsplanner.setClass(getApplicationContext(),
MealPlannerListActivity.class);
            startActivity(mealsplanner);
         }
      });

      linrCreateNewMealPlan.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            To_Notes.setAction(Intent.ACTION_VIEW);
            To_Notes.setClass(getApplicationContext(),
DailymealplanActivity.class);
            To_Notes.putExtra("Data", new Gson().toJson(Notes_List));
            To_Notes.putExtra("Date", txtDate.getText().toString());
            startActivity(To_Notes);
            clickAnimation(linrCreateNewMealPlan);
            today.edit().putString("Today", "false").commit();
         }
      });

      //Open meal list on another activity
      listview2.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
         @Override
         public void onItemClick(AdapterView<?> param1, View param2, int
param3, long param4) {
            final int position = param3;
            to_mealview.setAction(Intent.ACTION_VIEW);
            to_mealview.setClass(getApplicationContext(),
PlannedMealsViewActivity.class);
            to_mealview.putExtra("Position",
String.valueOf((long)((meal_list.size() - 1) - position)));
            startActivity(to_mealview);
         }
      });

      linrCreateDiary.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            clickAnimation(linrCreateDiary);
            diary.setClass(getApplicationContext(),
FoodDiaryListActivity.class);
            diary.putExtra("Type", "All");
            startActivity(diary);
         }
      });

      circleimageview2.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            startActivity(new Intent(MainActivity.this,
SearchscanmealActivity.class)); Animatoo.animateFade(MainActivity.this);
```

```
            clickAnimation(circleimageview2);
        }
    });

    linrAllGoals.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            clickAnimation(linrAllGoals);
            goal.setAction(Intent.ACTION_VIEW);
            goal.setClass(getApplicationContext(),
GoalslistActivity.class);
            goal.putExtra("Type", "All");
            startActivity(goal);
        }
    });

    linrCompltdGoals.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            clickAnimation(linear8);
            goal.setAction(Intent.ACTION_VIEW);
            goal.setClass(getApplicationContext(),
GoalslistActivity.class);
            goal.putExtra("Type", "Completed");
            startActivity(goal);
        }
    });


    //this linear acts as a button of a custom bottom bar navigation
    //onClick returns the user to homepage fragment
    linrBtnHome.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            clickAnimation(linrBtnHome);
            linrBtnHome.setBackground(new GradientDrawable() {
                public GradientDrawable getIns(int a, int b) {
                    this.setCornerRadius(a);
                    this.setColor(b);
                    return this;
                }
            }.getIns((int)50, 0xFFF4A522));
            linrBtnMore.setBackground(new GradientDrawable() {
                public GradientDrawable getIns(int a, int b) {
                    this.setCornerRadius(a);
                    this.setColor(b);
                    return this;
                }
            }.getIns((int)50, Color.TRANSPARENT));
            linrBtnGoal.setBackground(new GradientDrawable() {
                public GradientDrawable getIns(int a, int b) {
                    this.setCornerRadius(a);
                    this.setColor(b);
                    return this;
                }
            }.getIns((int)50, Color.TRANSPARENT));
            linrFoodDiary.setBackground(new GradientDrawable() {
                public GradientDrawable getIns(int a, int b) {
                    this.setCornerRadius(a);
                    this.setColor(b);
                    return this;
```

```
            }
        }.getIns((int)50, Color.TRANSPARENT));
        vscroll1.setVisibility(View.VISIBLE);
        fragment2.setVisibility(View.GONE);
        fragment3.setVisibility(View.GONE);
        fragment4.setVisibility(View.GONE);
    }
});

//onClick returns the user to goals fragment
linrBtnGoal.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        clickAnimation(linrBtnGoal);
        linrBtnGoal.setBackground(new GradientDrawable() {
            public GradientDrawable getIns(int a, int b) {
                this.setCornerRadius(a);
                this.setColor(b);
                return this;
            }
        }.getIns((int)50, 0xFFF4A522));
        linrBtnHome.setBackground(new GradientDrawable() {
            public GradientDrawable getIns(int a, int b) {
                this.setCornerRadius(a);
                this.setColor(b);
                return this; }
        }.getIns((int)50, Color.TRANSPARENT));
        linrBtnMore.setBackground(new GradientDrawable() {
            public GradientDrawable getIns(int a, int b) {
                this.setCornerRadius(a);
                this.setColor(b);
                return this;
            }
        }.getIns((int)50, Color.TRANSPARENT));
        linrFoodDiary.setBackground(new GradientDrawable() {
            public GradientDrawable getIns(int a, int b) {
                this.setCornerRadius(a);
                this.setColor(b);
                return this;
            }
        }.getIns((int)50, Color.TRANSPARENT));
        vscroll1.setVisibility(View.GONE);
        fragment2.setVisibility(View.VISIBLE);
        fragment3.setVisibility(View.GONE);
        fragment4.setVisibility(View.GONE);
    }
});

//onClick returns the user to food diary fragment
linrFoodDiary.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        clickAnimation(linrFoodDiary);
        linrBtnGoal.setBackground(new GradientDrawable() {
            public GradientDrawable getIns(int a, int b) {
                this.setCornerRadius(a);
                this.setColor(b);
                return this;
            }
        }.getIns((int)50, Color.TRANSPARENT));
        linrBtnHome.setBackground(new GradientDrawable() {
```

```
            public GradientDrawable getIns(int a, int b) {
                this.setCornerRadius(a);
                this.setColor(b);
                return this;
            }
        }.getIns((int)50, Color.TRANSPARENT));
        linrBtnMore.setBackground(new GradientDrawable() {
            public GradientDrawable getIns(int a, int b) {
                this.setCornerRadius(a);
                this.setColor(b);
                return this;
            }
        }.getIns((int)50, Color.TRANSPARENT));
        linrFoodDiary.setBackground(new GradientDrawable() {
            public GradientDrawable getIns(int a, int b) {
                this.setCornerRadius(a);
                this.setColor(b);
                return this;
            }
        }.getIns((int)50, 0xFFF4A522));
        vscroll1.setVisibility(View.GONE);
        fragment2.setVisibility(View.GONE);
        fragment3.setVisibility(View.GONE);
        fragment4.setVisibility(View.VISIBLE);
    }
});

//onClick returns the user to food diary fragment
linrBtnMore.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        clickAnimation(linrBtnMore);
        linrBtnHome.setBackground(new GradientDrawable() {
            public GradientDrawable getIns(int a, int b) {
                this.setCornerRadius(a);
                this.setColor(b);
                return this;
            }
        }.getIns((int)50, Color.TRANSPARENT));
        linrBtnMore.setBackground(new GradientDrawable() {
            public GradientDrawable getIns(int a, int b) {
                this.setCornerRadius(a);
                this.setColor(b);
                return this;
            }
        }.getIns((int)50, 0xFFF4A522));
        linrBtnGoal.setBackground(new GradientDrawable() {
            public GradientDrawable getIns(int a, int b) {
                this.setCornerRadius(a);
                this.setColor(b);
                return this;
            }
        }.getIns((int)50, Color.TRANSPARENT));
        linrFoodDiary.setBackground(new GradientDrawable() {
            public GradientDrawable getIns(int a, int b) {
                this.setCornerRadius(a);
                this.setColor(b);
                return this;
            }
        }.getIns((int)50, Color.TRANSPARENT));
        vscroll1.setVisibility(View.GONE);
```

```
            fragment2.setVisibility(View.GONE);
            fragment3.setVisibility(View.VISIBLE);
            fragment4.setVisibility(View.GONE);
         }
      });

      //onChildAdded of the firebase database, we'll get the user details
from our firebase
      sa_db_child_listener = new ChildEventListener() {
         @Override
         public void onChildAdded(DataSnapshot param1, String param2) {
            GenericTypeIndicator<HashMap<String, Object>> ind = new
GenericTypeIndicator<HashMap<String, Object>>() {};
            final String childKey = param1.getKey();
            final HashMap<String, Object> childValue =
param1.getValue(ind);
            try{
               //check if the user is not null
               if ((FirebaseAuth.getInstance().getCurrentUser() != null)) {
                  //Check if the current logged user equals the requesting
UId on the firebase database
                  //Then the user details gotten are then saved to
different appropriate strings
                  if
(FirebaseAuth.getInstance().getCurrentUser().getUid().equals(childValue.get
("uid").toString())) {
                     name1 = childValue.get("username").toString();
                     weights = childValue.get("weight").toString();
                     email =
FirebaseAuth.getInstance().getCurrentUser().getEmail();
                     height = childValue.get("height").toString();
                     name.edit().putString("name", name1).commit();
                     userdetail.edit().putString("weight",
weights).commit();
                     userdetail.edit().putString("email", email).commit();
                     userdetail.edit().putString("height",
height).commit();
                  }
               }
            }
            catch(Exception e){
               //do something if error occurs
               //To know error use _e.toString()

            }
         }

         @Override
         public void onChildChanged(@NonNull DataSnapshot param1, String
param2) {
            GenericTypeIndicator<HashMap<String, Object>> ind = new
GenericTypeIndicator<HashMap<String, Object>>() {};
            final String childKey = param1.getKey();
            final HashMap<String, Object> childValue =
param1.getValue(ind);

         }

         @Override
         public void onChildMoved(@NonNull DataSnapshot param1, String
param2) {
```

```
        }

        @Override
        public void onChildRemoved(@NonNull DataSnapshot param1) {
            GenericTypeIndicator<HashMap<String, Object>> ind = new
GenericTypeIndicator<HashMap<String, Object>>() {};
            final String childKey = param1.getKey();
            final HashMap<String, Object> childValue =
param1.getValue(ind);

        }

        @Override
        public void onCancelled(@NonNull DatabaseError param1) {
            final int _errorCode = param1.getCode();
            final String errorMessage = param1.getMessage();

        }
    };
    sa_db.addChildEventListener(sa_db_child_listener);

    auth_updateEmailListener = new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

        }
    };

    auth_updatePasswordListener = new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

        }
    };

    auth_emailVerificationSentListener = new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

        }
    };

    auth_deleteUserListener = new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

        }
    };
```

```
      auth_phoneAuthListener = new OnCompleteListener<AuthResult>() {
         @Override
         public void onComplete(@NonNull Task<AuthResult> task){
            final boolean success = task.isSuccessful();
            final String errorMessage = task.getException() != null ?
task.getException().getMessage() : "";

         }
      };

      auth_updateProfileListener = new OnCompleteListener<Void>() {
         @Override
         public void onComplete(@NonNull Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };

      auth_googleSignInListener = new OnCompleteListener<AuthResult>() {
         @Override
         public void onComplete(@NonNull Task<AuthResult> task){
            final boolean success = task.isSuccessful();
            final String errorMessage = task.getException() != null ?
task.getException().getMessage() : "";

         }
      };

      auth_create_user_listener = new OnCompleteListener<AuthResult>() {
         @Override
         public void onComplete(@NonNull Task<AuthResult> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };

      authsign_in_listener = new OnCompleteListener<AuthResult>() {
         @Override
         public void onComplete(@NonNull Task<AuthResult> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };

      _auth_reset_password_listener = new OnCompleteListener<Void>() {
         @Override
         public void onComplete(@NonNull Task<Void> param1) {
            final boolean success = param1.isSuccessful();

         }
      };
   }

   @SuppressLint("UseCompatLoadingForDrawables")
```

```
   private void initializeLogic() {
      if (Build.VERSION.SDK_INT > Build.VERSION_CODES.KITKAT) {
         Window w =MainActivity.this.getWindow();
         w.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);

w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
w.setStatusBarColor(0xFFE0E0E0);
      }
      progressbar1.getProgressDrawable().setColorFilter(Color.WHITE,
PorterDuff.Mode.SRC_IN);
      progressbar2.setProgressDrawable(getDrawable(R.drawable.progress));
      progressbar3.setProgressDrawable(getDrawable(R.drawable.progress));
      progressbar4.setProgressDrawable(getDrawable(R.drawable.progress));
      ActivityFont("ralewaysemibold");
      _cal();
      UI();
      removeScrollBar(vscroll1);
      removeScrollBar(vscroll3);
      removeScrollBar(vscroll4);
      show = 0;

      //Checks if user is logged in, if not such user will be redirected to
registration page
      if ((FirebaseAuth.getInstance().getCurrentUser() != null)) {

      }
      else {
         register.setClass(getApplicationContext(),
CreateaccountActivity.class);
         startActivity(register);
         finish();
      }
      ONCREATE_LIST();

      //Get username to a string then send a random greeting message to the
user
      txt_username.setText(name.getString("name", ""));
      if (ToastUtil.getRandom((int)(1), (int)(3)) == 1) {
         textview1.setText("Hi, ");
      }
      else {
         if (ToastUtil.getRandom((int)(1), (int)(3)) == 2) {
            textview1.setText("Hello, ");
         }
         else {
            if (ToastUtil.getRandom((int)(1), (int)(3)) == 3) {
               textview1.setText("Eat Healthiest, ");
            }
         }
      }

      //Get the number of week from the saved sharedPreferences of week
saved from the weightActivity
      if (week.getString("week", "").equals("")) {
         txtWeekNum.setText("0");
      }
      else {
         txtWeekNum.setText(week.getString("week", ""));
      }
      //Get the current weight of the user from the saved sharedPreferences
of weight saved from the weightActivity
```

```
      txtCurrentWeight.setText("Current weight:
".concat(week.getString("week", "")));

      //Get the list saved meals by the user
      //check if the key "Meals" is contains any saved value from saved
meal sharedPreferences
      if (!meals.getString("Meals", "").equals("")) {
         //Create a new ArrayList HashMap and fetch the saved list of meal
from saved sharedPreferences of "Meals" into Gson
         meal_list = new Gson().fromJson(meals.getString("Meals", ""), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
         //Load the data into a Listview
         listview2.setAdapter(new Listview2Adapter(meal_list));
         ((BaseAdapter)listview2.getAdapter()).notifyDataSetChanged();
      }
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   @Override
   public void onStart() {
      super.onStart();
      ONCREATE_LIST();
   }

   @Override
   public void onResume() {
      super.onResume();
      //Refresh all the data such as meals, goals, completed goals so as to
get any changes made after the user adds or delete any
      //list item in their respective listView
      //NOTE: Always check if the key has any saved value to avoid crashing
of the app
      refresh(meal_list);
      if (!meals.getString("Meals", "").equals("")) {
         meal_list = new Gson().fromJson(meals.getString("Meals", ""), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      }
      if (!AllGoals.getString("Goals", "").equals("")) {
         Goal_List = new Gson().fromJson(AllGoals.getString("Goals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      }
      if (!AllGoals.getString("Compltd", "").equals("")) {
         Compltd_List = new Gson().fromJson(AllGoals.getString("Compltd",
""), new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      }
      if (protein.getString("protein", "").equals("")) {

      }
      else {
         //If the value of protein is found, its then passed to a text and
progressBar
```

```
         txtProteinFigure.setText(protein.getString("protein", ""));

progressbar2.setProgress((int)Double.parseDouble(txtProteinFigure.getText()
.toString()));

progressbar2.setMax((int)Double.parseDouble(txtProteinFigure.getText().toSt
ring()) * 2);
      }
      if (calories.getString("calories", "").equals("")) {

      }
      else {
         txtCalorieFigure.setText(calories.getString("calories", ""));

progressbar4.setProgress((int)Double.parseDouble(txtCalorieFigure.getText()
.toString()));

progressbar4.setMax((int)Double.parseDouble(txtCalorieFigure.getText().toSt
ring()) * 2);
      }
   }

   //this method changes the whole text font in this activity
   public void ActivityFont (final String fontname) {
      fontName = "fonts/".concat(fontname.concat(".ttf"));
      overrideFonts(this,getWindow().getDecorView());
   }
   private void overrideFonts(final android.content.Context context, final
View v) {

      try {
          Typeface
          typeace = Typeface.createFromAsset(getAssets(), fontName);;
          if ((v instanceof ViewGroup)) {
              ViewGroup vg = (ViewGroup) v;
              for (int i = 0;
              i < vg.getChildCount();
              i++) {
                  View child = vg.getChildAt(i);
                  overrideFonts(context, child);
                }
            }
          else {
              if ((v instanceof TextView)) {
                  ((TextView) v).setTypeface(typeace);
                }
              else {
                  if ((v instanceof EditText )) {
                      ((EditText) v).setTypeface(typeace);
                    }
                  else {
                      if ((v instanceof Button)) {
                          ((Button) v).setTypeface(typeace);
                        }
                    }
                }
            }
      }
      catch(Exception e){
      };
   }
```

```
   public void _fade_translate () {
      overridePendingTransition(android.R.anim.fade_in,
android.R.anim.fade_out);
   }


   public void clickAnimation (final View view) {
      ScaleAnimation fade_in = new ScaleAnimation(0.9f, 1f, 0.9f, 1f,
Animation.RELATIVE_TO_SELF, 0.5f, Animation.RELATIVE_TO_SELF, 0.7f);
      fade_in.setDuration(300);
      fade_in.setFillAfter(true);
      view.startAnimation(fade_in);
   }


   public void ONCREATE_LIST () {
      if (!AllMeals.getString("Meals", "").equals("")) {
         Notes_List = new Gson().fromJson(AllMeals.getString("Meals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      }
      if (Notes_List.size() == 0) {
         txtTotalMeal.setText("Meals : 0");
      }
      else {
         GetRecentDataFrom(Notes_List.size() - 1);
         txtTotalMeal.setText("Meals :
".concat(String.valueOf((long)(Notes_List.size()))));
      }
      if (week.getString("week", "").equals("")) {

      }
      else {
         txtWeekNum.setText(week.getString("week", ""));
         txtCurrentWeight.setText(userdetail.getString("weight", ""));
      }
   }

   //Get the last position data of meals list to texts
   public void GetRecentDataFrom (final double position) {
      if (Notes_List.get((int)position).containsKey("Title")) {
         if (Notes_List.get((int)position).get("Title").toString().length()
> 24) {

title.setText(Notes_List.get((int)position).get("Title").toString().substri
ng((int)(0), (int)(24)).concat("..."));
         }
         else {

title.setText(Notes_List.get((int)position).get("Title").toString());
         }
      }
      if (Notes_List.get((int)position).containsKey("Date")) {
         time.setText("Last Updated :
".concat(Notes_List.get((int)position).get("Date").toString()));
      }
   }

   //Get the current date and day from calender component
   public void _cal () {
```

```
      imageview38.setVisibility(View.GONE);
      textview50.setVisibility(View.GONE);
      if (new
SimpleDateFormat("dd").format(cal2.getTime()).startsWith("0")) {
         txtDate.setText(new
SimpleDateFormat("dd").format(cal2.getTime()).replace("0", ""));
      }
      else {
         txtDate.setText(new
SimpleDateFormat("dd").format(cal2.getTime()));
      }
      cal2 = Calendar.getInstance();
      txtDay.setText(new SimpleDateFormat("EEEE").format(cal2.getTime()));
      //If meal list is empty, then an imageview and textview becomes
visible to inform the user
      if (meal_list.size() == 0) {
         imageview38.setVisibility(View.VISIBLE);
         textview50.setVisibility(View.VISIBLE);
      }
      else {
         imageview38.setVisibility(View.GONE);
         textview50.setVisibility(View.GONE);
      }

      //Do not remove try catch block here because some users may not enter
some data while the code checks
      //for such data which is empty may cause crash on the app
      try{
         /**Get All Goals entered by the user
         //Note: the method of getting these data is same method which
involves: sharedPreferences, Gson, ArrayList <HashMap>
         **/
         if (!AllGoals.getString("Goals", "").equals("")) {
            Goal_List = new Gson().fromJson(AllGoals.getString("Goals",
""), new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());

txtAllGoalsLength.setText(String.valueOf((long)(Goal_List.size())));
         }
         if (!AllGoals.getString("Compltd", "").equals("")) {
            Compltd_List = new
Gson().fromJson(AllGoals.getString("Compltd", ""), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());

txtCompletedLength.setText(String.valueOf((long)(Compltd_List.size())));

progressbar1.setProgress((int)Double.parseDouble(String.valueOf((long)(Comp
ltd_List.size()))));
         }
         if (protein.getString("protein", "").equals("")) {

         }
         else {
            txtProteinFigure.setText(protein.getString("protein", ""));

progressbar2.setProgress((int)Double.parseDouble(txtProteinFigure.getText()
.toString()));

progressbar2.setMax((int)Double.parseDouble(txtProteinFigure.getText().toSt
ring()) * 2);
         }
         if (calories.getString("calories", "").equals("")) {
```

```
         }
         else {
            txtCalorieFigure.setText(calories.getString("calories", ""));

progressbar4.setProgress((int)Double.parseDouble(txtCalorieFigure.getText()
.toString()));

progressbar4.setMax((int)Double.parseDouble(txtCalorieFigure.getText().toSt
ring()) * 2);
         }
         if (fat.getString("fat", "").equals("")) {

         }
         else {
            txtCalorieFigure.setText(fat.getString("fat", ""));

progressbar3.setProgress((int)Double.parseDouble(txtFatFigure.getText().toS
tring()));

progressbar3.setMax((int)Double.parseDouble(txtFatFigure.getText().toString
()) * 2);
         }
      }catch(Exception e){

      }
   }


   public void UI () {
      lottie1.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) {
         this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)100, 0xFFFFFFFF));
      textview1.setTypeface(Typeface.createFromAsset(getAssets(),
"fonts/railway_semisolid.ttf"), Typeface.NORMAL);
      txt_username.setTypeface(Typeface.createFromAsset(getAssets(),
"fonts/railway_semisolid.ttf"), Typeface.NORMAL);
      textview8.setTypeface(Typeface.createFromAsset(getAssets(),
"fonts/railway_semisolid.ttf"), Typeface.NORMAL);
      textview9.setTypeface(Typeface.createFromAsset(getAssets(),
"fonts/railway_semisolid.ttf"), Typeface.NORMAL);
      textview15.setTypeface(Typeface.createFromAsset(getAssets(),
"fonts/railway_semisolid.ttf"), Typeface.NORMAL);
      textview16.setTypeface(Typeface.createFromAsset(getAssets(),
"fonts/railway_semisolid.ttf"), Typeface.NORMAL);
      textview18.setTypeface(Typeface.createFromAsset(getAssets(),
"fonts/railway_semisolid.ttf"), Typeface.NORMAL);
      title.setTypeface(Typeface.createFromAsset(getAssets(),
"fonts/railway_semisolid.ttf"), Typeface.NORMAL);
      time.setTypeface(Typeface.createFromAsset(getAssets(),
"fonts/railway_semisolid.ttf"), Typeface.NORMAL);
      linrGraph.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) {
         this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)30, 0xFF3F51B6));
      linrCreateNewMealPlan.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) {
         this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)15, 0xFFFFFFFF));
      linrCreateDiary.setBackground(new GradientDrawable() { public
```

```
GradientDrawable getIns(int a, int b) {
        this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)15, 0xFFFFFFFF));
      linrDiary.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) {
        this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)30, 0xFFFFFFFF));
   }


   public void removeScrollBar (final View view) {
      view.setVerticalScrollBarEnabled(false);
      view.setOverScrollMode(ScrollView.OVER_SCROLL_NEVER);

   }


   public void refresh (final ArrayList<HashMap<String, Object>> ListMap) {
      Parcelable state =
      listview2.onSaveInstanceState();
      listview2.setAdapter(new Listview2Adapter(ListMap));
      ((BaseAdapter)listview2.getAdapter()).notifyDataSetChanged();
      listview2.onRestoreInstanceState(state);
   }


   public void GetImportantNotes () {
      Important_Goal_List.clear();
      Position = 0;
      for(int repeat10 = 0; repeat10 < (int)(Goal_List.size()); repeat10++)
{
         if
(Goal_List.get((int)Position).get("Important").toString().equals("True")) {
            GetGoals = Goal_List.get((int)Position);
            Important_Goal_List.add(GetGoals);
         }
         Position++;
      }
   }

   //Listview adapter is instantiated here
   /**Custom ListView Adapter**/
   public class Listview2Adapter extends BaseAdapter {
      ArrayList<HashMap<String, Object>> data;
      public Listview2Adapter(ArrayList<HashMap<String, Object>> arr) {
         data = arr;
      }

      @Override
      public int getCount() {
         return data.size();
      }

      @Override
      public HashMap<String, Object> getItem(int index) {
         return data.get(index);
      }

      @Override
      public long getItemId(int index) {
         return index;
```

```
      }
      @Override
      public View getView(final int position, View v, ViewGroup container)
{
         LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
         View view = v;
         if (view == null) {
            view = inflater.inflate(R.layout.mealslist, null);
         }

         final LinearLayout bg =  view.findViewById(R.id.bg);
         final LinearLayout linear4 =  view.findViewById(R.id.linear4);
         final LinearLayout linear3 =  view.findViewById(R.id.linear3);
         final LinearLayout main =  view.findViewById(R.id.main);
         final ImageView imageview1 =  view.findViewById(R.id.imageview1);
         final ImageView imgReaction =
view.findViewById(R.id.imgReaction);
         final LinearLayout text_lin =  view.findViewById(R.id.text_lin);
         final LinearLayout title_time_lin =
view.findViewById(R.id.title_time_lin);
         final LinearLayout linear2 =  view.findViewById(R.id.linear2);
         final LinearLayout linear5 =  view.findViewById(R.id.linear5);
         final TextView title = (TextView) view.findViewById(R.id.title);
         final LinearLayout linear7 =  view.findViewById(R.id.linear7);
         final TextView txtCalories = (TextView)
view.findViewById(R.id.txtCalories);
         final TextView txtTotalkCal = (TextView)
view.findViewById(R.id.txtTotalkCal);
         final TextView time = (TextView) view.findViewById(R.id.time);
         final TextView date = (TextView) view.findViewById(R.id.date);
         final TextView txtBreakfast = (TextView)
view.findViewById(R.id.txtBreakfast);
         final TextView txtLunch = (TextView)
view.findViewById(R.id.txtLunch);
         final TextView txtSnack = (TextView)
view.findViewById(R.id.txtSnack);
         final TextView txtDinner = (TextView)
view.findViewById(R.id.txtDinner);

         linear4.setVisibility(View.GONE);
         linear3.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) {
            this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)100, 0xFFFFFFFF));
         main.setElevation((float)0);
         linear3.setElevation((float)10);
         main.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) {
            this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)15, 0xFFFFFFFF));
         imageview1.setVisibility(View.GONE);

         if (meal_list.get(meal_list.size() - 1 -
position).containsKey("Time")) {
            time.setText(meal_list.get(meal_list.size() - 1 -
position).get("Time").toString());
         }
         if (meal_list.get(meal_list.size() - 1 -
position).containsKey("Date")) {
```

```
            date.setText(meal_list.get(meal_list.size() - 1 -
position).get("Date").toString());
         }
         if (meal_list.get(meal_list.size() - 1 -
position).containsKey("Breakfast")) {
            txtBreakfast.setText(meal_list.get(meal_list.size() - 1 -
position).get("Breakfast").toString());
         }
         if (meal_list.get(meal_list.size() - 1 -
position).containsKey("Lunch")) {
            txtLunch.setText(meal_list.get(meal_list.size() - 1 -
position).get("Lunch").toString());
         }
         if (meal_list.get(meal_list.size() - 1 -
position).containsKey("Snack")) {
            txtSnack.setText(meal_list.get(meal_list.size() - 1 -
position).get("Snack").toString());
         }
         if (meal_list.get(meal_list.size() - 1 -
position).containsKey("Dinner")) {
            txtDinner.setText(meal_list.get(meal_list.size() - 1 -
position).get("Dinner").toString());
         }
         if (meal_list.get(meal_list.size() - 1 -
position).containsKey("Totalkcal")) {
            txtTotalkCal.setText(meal_list.get(meal_list.size() - 1 -
position).get("Totalkcal").toString());
         }
         if (((meal_list.get(meal_list.size() - 1 -
position).get("Dinner").toString().equals("No Dinner Yet")
               || meal_list.get(meal_list.size() - 1 -
position).get("Lunch").toString().equals("What do you eat for lunch?"))
               || meal_list.get(meal_list.size() - 1 -
position).get("Breakfast").toString().equals("Breakfast Not Entered"))
               || meal_list.get(meal_list.size() - 1 -
position).get("Snack").toString().equals("No snack is added")) {
            title.setText("Incomplete Meal:\nA meal is missing");
            imgReaction.setImageResource(R.drawable.emoticonsad);
         }
         else {
            title.setText("Complete Meal ");
            imgReaction.setImageResource(R.drawable.emoticoncool);
         }

         return view;
      }
   }

   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }
```

```
    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Main Recipe Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.drawable.GradientDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.util.SparseBooleanArray;
```

```
import android.util.TypedValue;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.bumptech.glide.Glide;
import com.google.gson.Gson;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;


public class MainrecipeActivity extends AppCompatActivity {

   private HashMap<String, Object> header = new HashMap<>();
   private HashMap<String, Object> param = new HashMap<>();
   private String st = "";
   private String nutriments = "";
   private String name = "";
   private String img = "";
   private String text = "";
   private String calories = "";
   private String quantity = "";
   private String fat = "";
   private String carbo = "";
   private String sugar = "";
   private String fiber = "";
   private String protein = "";
   private String cholesterol = "";
   private String ingredients = "";
   private HashMap<String, Object> recipe = new HashMap<>();

   private ArrayList<HashMap<String, Object>> listmap = new ArrayList<>();
   private ArrayList<HashMap<String, Object>> listmap2 = new ArrayList<>();
   private ArrayList<String> map = new ArrayList<>();

   private LinearLayout headsearch;
   private LinearLayout linrLoading;
   private ListView listview1;
   private LinearLayout linear4;
   private LinearLayout linear3;
   private TextView textview3;
   private EditText edittext2;
   private ImageView imageview1;
   private ProgressBar progressbar1;
   private TextView textview4;

   private RequestNetwork webapi;
```

```
   private RequestNetwork.RequestListener _webapi_request_listener;
   private Intent i = new Intent();
   private SharedPreferences holdrec;
   private AlertDialog.Builder d1;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.mainrecipe);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      headsearch =  findViewById(R.id.headsearch);
      linrLoading =  findViewById(R.id.linrLoading);
      listview1 = (ListView) findViewById(R.id.listview1);
      linear4 =  findViewById(R.id.linear4);
      linear3 =  findViewById(R.id.linear3);
      textview3 = (TextView) findViewById(R.id.textview3);
      edittext2 = (EditText) findViewById(R.id.edittext2);
      imageview1 =  findViewById(R.id.imageview1);
      progressbar1 = (ProgressBar) findViewById(R.id.progressbar1);
      textview4 = (TextView) findViewById(R.id.textview4);
      webapi = new RequestNetwork(this);
      holdrec = getSharedPreferences("holdrec", Activity.MODE_PRIVATE);
      d1 = new AlertDialog.Builder(this);

      listview1.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
         @Override
         public void onItemClick(AdapterView<?> param1, View param2, int
param3, long param4) {
            final int position = param3;
            try{
               i.putExtra("name",
listmap.get((int)position).get("name").toString());
               i.putExtra("img",
listmap.get((int)position).get("img").toString());
               i.putExtra("ingredients",
listmap.get((int)position).get("ingredients").toString());
               i.putExtra("calories",
listmap.get((int)position).get("calories").toString());
               i.putExtra("fat",
listmap.get((int)position).get("fat").toString());
               i.putExtra("carbo",
listmap.get((int)position).get("carbo").toString());
               i.putExtra("protein",
listmap.get((int)position).get("protein").toString());
               i.putExtra("fiber",
listmap.get((int)position).get("fiber").toString());
               i.putExtra("sugar",
listmap.get((int)position).get("sugar").toString());
               i.putExtra("position", String.valueOf((long)(position)));
               i.putExtra("cholesterol",
listmap.get((int)position).get("cholesterol").toString());
               i.putExtra("text",
listmap.get((int)position).get("text").toString());
               i.setClass(getApplicationContext(),
ViewrecipeActivity.class);
```

```
                startActivity(i);
            }catch(Exception e){
                ToastUtil.showMessage(getApplicationContext(),
e.toString());
            }
         }
      });

      imageview1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            if (edittext2.getText().toString().equals("")) {
               ToastUtil.showMessage(getApplicationContext(), "Empty");
            }
            else {
               i.putExtra("search",
"https://api.edamam.com/api/recipes/v2?type=public&q=".concat(edittext2.get
Text().toString()).concat("&app_id=c74a6084&app_key=f306e4b68fcc0154b28a3b5
a97e48ebd"));
               i.setClass(getApplicationContext(),
RecipesearchedActivity.class);
               startActivity(i);
            }
         }
      });


      //This webapi request listener works with RequestNetwork and
RequestNetworkController java files
      //to make request
      //to the site api.
      //onResponse needs some parameters such as tag and response strings
      _webapi_request_listener = new RequestNetwork.RequestListener() {
         @Override
         public void onResponse(String param1, String param2,
HashMap<String, Object> param3) {
            final String tag = param1;
            final String response = param2;
            final HashMap<String, Object> _responseHeaders = param3;
            linrLoading.setVisibility(View.GONE);
            if (tag.equals("a")) {
               try {


                  //Our response from
https://api.edamam.com/api/recipes/v2?type=public&q= api begins with object
//                NOTE: I make request for different food name randomly in
onCreate such as chicken, potato etc..
//                Since we can't get list of foods from edamam.com
                  //Create an object org.json.JSONObject for the response
                  //from the json response, the detail needed are stored in
an array "products"
                  //Loop the array the to get individual objects in the
products array
                  //make sure you don't remove the block if(...).has(..)
because this checks if
                  // the object value we're requesting is made available
                  org.json.JSONObject obj = new
org.json.JSONObject(response);

                  //Loop "hits" array to get the value in the respective
```

```
value to a listView or
                    // RecyclerView
                    org.json.JSONArray array = obj.getJSONArray("hits");
                    for(int i=0;i<array.length();i++){ org.json.JSONObject
object = array.getJSONObject(i);

                        //Creates strings to save the value from the object
                        if (object.getJSONObject("recipe").has("label")) {
                            name =
object.getJSONObject("recipe").getString("label").toString(); }

                        if (object.getJSONObject("recipe").has("image")) {
                            img =
object.getJSONObject("recipe").getString("image").toString(); }

                        if (object.getJSONObject("recipe").has("calories")) {
                            calories =
object.getJSONObject("recipe").getString("calories").toString(); }

//                      Below we're requesting from object>object>object to
get the value of the object which is not as direct as above codes

                        if
(object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObje
ct("FAT").has("quantity")) {
                            fat =
object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObjec
t("FAT").getString("quantity").toString(); }

                        if
(object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObje
ct("CHOCDF").has("quantity")) {
                            carbo =
object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObjec
t("CHOCDF").getString("quantity").toString(); }

                        if
(object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObje
ct("SUGAR").has("quantity")) {
                            sugar =
object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObjec
t("SUGAR").getString("quantity").toString(); }

                        if
(object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObje
ct("FIBTG").has("quantity")) {
                            fiber =
object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObjec
t("FIBTG").getString("quantity").toString(); }

                        if
(object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObje
ct("PROCNT").has("quantity")) {
                            protein =
object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObjec
t("PROCNT").getString("quantity").toString(); }

                        if
(object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObje
ct("CHOLE").has("quantity")) {
                            cholesterol =
```

```
object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObjec
t("CHOLE").getString("quantity").toString(); }

//                    To get the ingredients of a specific meal:
                      /*It's a bit complex here, because we'll be making
request from an array in an objects
                      * and same time it's a same process as getting "hits"
array from the object "response"
                      * Please do check the
https://developer.edamam.com/edamam-docs-recipe-api docs for Searching
Recipe Api and see the
                      * json structure to understand this
                      * more better*/

//                    Creates an array for Ingredients
                      org.json.JSONArray arrayIng =
object.getJSONObject("recipe").getJSONArray("ingredients");

                      //Loop "ingredients" array to get the value in the
respective value to an ArrayList or

                      for(int j=0;j<arrayIng.length();j++){

                         //Creates ingredients string to save the value from
the object created below
                         //After looping the text is given in a string
                         org.json.JSONObject objects =
arrayIng.getJSONObject(j);
                         if (objects.has("text")){
                            ingredients = objects.getString("text");
                         }

//                       Creates a new HashMap to fetch the ingredient text
                         recipe = new HashMap<>();
                         recipe.put("ingredients", ingredients);
                         listmap2.add(recipe);
                      }
                      holdrec.edit().putString("recipe", new
Gson().toJson(listmap2)).commit();
                      param = new HashMap<>();
                      param.put("name", name);
                      param.put("img", img);
                      param.put("calories", calories);
                      param.put("fat", fat);
                      param.put("carbo", carbo);
                      param.put("sugar", sugar);
                      param.put("fiber", fiber);
                      param.put("protein", protein);
                      param.put("cholesterol", cholesterol);
                      param.put("ingredients", new Gson().toJson(listmap2));
                      param.put("text", ingredients);
                      listmap.add(param);
                   }
                   listview1.setAdapter(new Listview1Adapter(listmap));

((BaseAdapter)listview1.getAdapter()).notifyDataSetChanged();
                } catch(final org.json.JSONException ex) {
                   ToastUtil.showMessage(getApplicationContext(),
ex.toString());
                }
             }
```

```
        }

        @Override
        public void onErrorResponse(String param1, String param2) {
            final String tag = param1;
            final String message = param2;
            linrLoading.setVisibility(View.GONE);
            d1.setTitle("Failed !!");
            d1.setMessage("Failed To Connect To The Server . Please Try
Again.");
            d1.setPositiveButton("Retry", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    finish();
                }
            });
            d1.setNegativeButton("Exit", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    finish();
                }
            });
            d1.setCancelable(false);
            d1.create().show();
        }
    };
  }

  private void initializeLogic() {

//    Fetch different search request at random to display in a listView
since there's no
//    Get list of food as openfoodfacts
    if (ToastUtil.getRandom((int)(0), (int)(5)) == 0) {

        //Search request for salad
        webapi.startRequestNetwork(RequestNetworkController.GET,
"https://api.edamam.com/api/recipes/v2?type=public&q=salad&app_id=c74a6084&
app_key=f306e4b68fcc0154b28a3b5a97e48ebd", "a", _webapi_request_listener);
    }
    else {
        if (ToastUtil.getRandom((int)(0), (int)(5)) == 1) {
//          Search request for chicken
            webapi.startRequestNetwork(RequestNetworkController.GET,
"https://api.edamam.com/api/recipes/v2?type=public&q=chicken&app_id=c74a608
4&app_key=f306e4b68fcc0154b28a3b5a97e48ebd", "a",
_webapi_request_listener);
        }
        else {
            if (ToastUtil.getRandom((int)(0), (int)(5)) == 2) {

//          Search request for potato
                webapi.startRequestNetwork(RequestNetworkController.GET,
"https://api.edamam.com/api/recipes/v2?type=public&q=potato&app_id=c74a6084
&app_key=f306e4b68fcc0154b28a3b5a97e48ebd", "a", _webapi_request_listener);
            }
            else {
                //          Search request for spaghetti
                if (ToastUtil.getRandom((int)(0), (int)(5)) == 3) {
```

```
                webapi.startRequestNetwork(RequestNetworkController.GET,
"https://api.edamam.com/api/recipes/v2?type=public&q=spaghetti&app_id=c74a6
084&app_key=f306e4b68fcc0154b28a3b5a97e48ebd", "a",
_webapi_request_listener);
              }
              else {
                 //          Search request for rice
                 if (ToastUtil.getRandom((int)(0), (int)(5)) == 4) {

webapi.startRequestNetwork(RequestNetworkController.GET,
"https://api.edamam.com/api/recipes/v2?type=public&q=rice&app_id=c74a6084&a
pp_key=f306e4b68fcc0154b28a3b5a97e48ebd", "a", _webapi_request_listener);
                 }
                 else {
                    //          Search request for fish
                    if (ToastUtil.getRandom((int)(0), (int)(5)) == 5) {

webapi.startRequestNetwork(RequestNetworkController.GET,
"https://api.edamam.com/api/recipes/v2?type=public&q=fish&app_id=c74a6084&a
pp_key=f306e4b68fcc0154b28a3b5a97e48ebd", "a", _webapi_request_listener);
                    }
                 }
              }
           }
        }
     }

     @Override
     protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);
        switch (requestCode) {

           default:
           break;
        }
     }

     public class Listview1Adapter extends BaseAdapter {
        ArrayList<HashMap<String, Object>> data;
        public Listview1Adapter(ArrayList<HashMap<String, Object>> arr) {
           data = arr;
        }

        @Override
        public int getCount() {
           return data.size();
        }

        @Override
        public HashMap<String, Object> getItem(int index) {
           return data.get(index);
        }

        @Override
        public long getItemId(int index) {
           return index;
        }
        @Override
        public View getView(final int position, View v, ViewGroup container)
```

```
{
        LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
        View view = v;
        if (view == null) {
           view = inflater.inflate(R.layout.txt, null);
        }

        final TextView textview1 = (TextView)
view.findViewById(R.id.textview1);
        final androidx.cardview.widget.CardView cardview1 =
(androidx.cardview.widget.CardView) view.findViewById(R.id.cardview1);
        final TextView textview2 = (TextView)
view.findViewById(R.id.textview2);
        final androidx.cardview.widget.CardView cardview2 =
(androidx.cardview.widget.CardView) view.findViewById(R.id.cardview2);
        final LinearLayout linear19 =  view.findViewById(R.id.linear19);
        final LinearLayout linear6 =  view.findViewById(R.id.linear6);
        final LinearLayout linear2 =  view.findViewById(R.id.linear2);
        final ImageView imageview1 =  view.findViewById(R.id.imageview1);
        final ImageView imageview2 =  view.findViewById(R.id.imageview2);
        final LinearLayout imageview3 =
view.findViewById(R.id.imageview3);
        final LinearLayout linear3 =  view.findViewById(R.id.linear3);
        final LinearLayout linear4 =  view.findViewById(R.id.linear4);
        final LinearLayout linear5 =  view.findViewById(R.id.linear5);
        final TextView txtBigBannerHead = (TextView)
view.findViewById(R.id.txtBigBannerHead);
        final LinearLayout linear7 =  view.findViewById(R.id.linear7);
        final TextView txtBigDescrip = (TextView)
view.findViewById(R.id.txtBigDescrip);
        final Button btnAdd = (Button) view.findViewById(R.id.btnAdd);
        final LinearLayout linrBack =  view.findViewById(R.id.linrBack);
        final ImageView imgDisplay =  view.findViewById(R.id.imgDisplay);
        final LinearLayout linear18 =  view.findViewById(R.id.linear18);
        final TextView txtMeal = (TextView)
view.findViewById(R.id.txtMeal);
        final LinearLayout linear17 =  view.findViewById(R.id.linear17);
        final ImageView imageview8 =  view.findViewById(R.id.imageview8);
        final TextView txtCalories = (TextView)
view.findViewById(R.id.txtCalories);
        final ImageView imageview9 =  view.findViewById(R.id.imageview9);
        final TextView txtIngredientsQty = (TextView)
view.findViewById(R.id.txtIngredientsQty);
        final TextView txtIngredients = (TextView)
view.findViewById(R.id.txtIngredients);
        final TextView txtMealType = (TextView)
view.findViewById(R.id.txtMealType);
        final TextView txtFatFig = (TextView)
view.findViewById(R.id.txtFatFig);
        final TextView txtCarboFig = (TextView)
view.findViewById(R.id.txtCarboFig);
        final TextView txtProteinFig = (TextView)
view.findViewById(R.id.txtProteinFig);
        final TextView txtSugar = (TextView)
view.findViewById(R.id.txtSugar);
        final TextView txtFiber = (TextView)
view.findViewById(R.id.txtFiber);
        final TextView txtCholestrol = (TextView)
view.findViewById(R.id.txtCholestrol);
```

```
            textview1.setVisibility(View.GONE);
            textview2.setVisibility(View.GONE);
            cardview1.setVisibility(View.GONE);

Glide.with(getApplicationContext()).load(Uri.parse(listmap.get((int)positio
n).get("img").toString())).into(imgDisplay);

txtMeal.setText(listmap.get((int)position).get("name").toString());
            if (listmap.get((int)position).get("calories").toString().length()
> 4) {
                txtCalories.setText("Calories:
".concat(listmap.get((int)position).get("calories").toString().substring((i
nt)(0), (int)(4)).concat("")));
            }

txtFatFig.setText(listmap.get((int)position).get("fat").toString());

txtCarboFig.setText(listmap.get((int)position).get("carbo").toString());

txtSugar.setText(listmap.get((int)position).get("sugar").toString());

txtProteinFig.setText(listmap.get((int)position).get("protein").toString())
;

txtFatFig.setText(listmap.get((int)position).get("fiber").toString());

txtBigDescrip.setText(listmap.get((int)position).get("cholesterol").toStrin
g());
            if (position == 0) {
                cardview1.setVisibility(View.VISIBLE);
                cardview2.setVisibility(View.GONE);

Glide.with(getApplicationContext()).load(Uri.parse(listmap.get((int)positio
n).get("img").toString())).into(imageview1);

txtBigBannerHead.setText(listmap.get((int)position).get("name").toString())
;

txtBigDescrip.setText(listmap.get((int)position).get("text").toString());
                textview1.setVisibility(View.VISIBLE);
            }
            cardview1.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)30, 0xFFFFFFFF));
            cardview2.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)20, 0xFFFFFFFF));
            btnAdd.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b, int c, int d) {
this.setCornerRadius(a); this.setStroke(b, c); this.setColor(d); return
this; } }.getIns((int)30, (int)2, 0xFFFAFAFA, 0xFF000000));
            linear19.setVisibility(View.GONE);
            if (position == 0) {
                textview2.setVisibility(View.VISIBLE);
            }

            return view;
        }
    }
```

```
    @Deprecated
    public void showMessage(String msg) {
        Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
    }

    @Deprecated
    public int getLocationX(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Meal Edit Activity

```java
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import android.graphics.drawable.GradientDrawable;
import android.os.Build;
import android.os.Bundle;
import android.os.Parcelable;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.view.WindowManager;
import android.view.animation.Animation;
import android.view.animation.ScaleAnimation;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.CalendarView;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.blogspot.atifsoftwares.animatoolib.Animatoo;
import com.google.android.material.button.MaterialButton;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import com.sdsmdg.tastytoast.TastyToast;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Random;


public class MealEditActivity extends AppCompatActivity {

   private double Position = 0;
   private boolean Selected = false;
   private double Selected_Length = 0;
   private boolean IsSelectAll = false;
```

```
   private String fontName = "";
   private String typeace = "";
   private String data = "";
   private String Title = "";
   private String Calories = "";
   private HashMap<String, Object> Get_Notes = new HashMap<>();
   private String Get_Note = "";
   private HashMap<String, Object> addmeal = new HashMap<>();
   private String Date = "";
   private String Time = "";
   private boolean breakfast = false;
   private boolean lunch = false;
   private boolean snack = false;
   private boolean dinner = false;
   private double show = 0;
   private String BreakfastServing = "";
   private String LunchServing = "";
   private String SnackServing = "";
   private String DinnerSnack = "";
   private boolean empty = false;

   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> temp_maplist = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> meal_list = new
ArrayList<>();

   private LinearLayout toolbar;
   private LinearLayout main;
   private ImageView select_all_img;
   private LinearLayout lirnrCalenderBody;
   private TextView textview26;
   private TextView textview27;
   private CalendarView calendarview1;
   private LinearLayout linear42;
   private TextView textview25;
   private Button button1;
   private LinearLayout linrMainMeal;
   private LinearLayout linrAddMeal;
   private ScrollView vscroll1;
   private LinearLayout linear19;
   private LinearLayout linear41;
   private LinearLayout linrBreakfast;
   private LinearLayout linrLunch;
   private LinearLayout linrSnack;
   private LinearLayout linrDinner;
   private MaterialButton btnSave;
   private ImageView imageview13;
   private TextView textview23;
   private TextView txtDate;
   private LinearLayout linear9;
   private LinearLayout linear11;
   private ImageView imgBreakfast;
   private TextView textview7;
   private ImageView imageview12;
   private LinearLayout linear13;
   private LinearLayout linear12;
   private ImageView imageview3;
   private TextView txtBreakfastTitle;
   private LinearLayout linear37;
```

```
    private TextView txtBreakfastClass;
    private TextView textview28;
    private TextView txtKCal;
    private LinearLayout linear31;
    private LinearLayout linear32;
    private ImageView imgLunch;
    private TextView textview16;
    private ImageView imageview14;
    private LinearLayout linear33;
    private LinearLayout linear34;
    private ImageView imageview9;
    private TextView txtLunchTitle;
    private LinearLayout linear38;
    private TextView txtLunchClass;
    private TextView textview29;
    private TextView txtLunchKcal;
    private LinearLayout linear26;
    private LinearLayout linear27;
    private ImageView imgSnack;
    private TextView textview13;
    private ImageView imageview15;
    private LinearLayout linear28;
    private LinearLayout linear29;
    private ImageView imageview7;
    private TextView txtSnackTitle;
    private LinearLayout linear39;
    private TextView txtSnckClass;
    private TextView textview30;
    private TextView txtSnackKcal;
    private LinearLayout linear21;
    private LinearLayout linear22;
    private ImageView imgDinner;
    private TextView textview10;
    private ImageView imageview16;
    private LinearLayout linear23;
    private LinearLayout linear24;
    private ImageView imageview5;
    private TextView txtDinnerTitle;
    private LinearLayout linear40;
    private TextView txtDinnerClass;
    private TextView textview31;
    private TextView txtDinnerKcal;
    private LinearLayout linear36;
    private LinearLayout linrClose;
    private TextView txtMealReceiver;
    private LinearLayout linrSearch;
    private ListView notes_list;
    private TextView textview22;
    private ImageView imageview10;
    private ImageView imageview11;
    private TextView textview21;

    private Intent To_Noteview = new Intent();
    private SharedPreferences Settings;
    private SharedPreferences AllMeals;
    private AlertDialog.Builder Delete_Dialog;
    private SharedPreferences meals;
    private Calendar Cal = Calendar.getInstance();
    private AlertDialog.Builder save;

    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.meal_edit);
    initialize(savedInstanceState);
    com.google.firebase.FirebaseApp.initializeApp(this);
    initializeLogic();
}

private void initialize(Bundle savedInstanceState) {
    toolbar =  findViewById(R.id.toolbar);
    main =  findViewById(R.id.main);
    select_all_img =  findViewById(R.id.select_all_img);
    lirnrCalenderBody =  findViewById(R.id.lirnrCalenderBody);
    textview26 = (TextView) findViewById(R.id.textview26);
    textview27 = (TextView) findViewById(R.id.textview27);
    calendarview1 = (CalendarView) findViewById(R.id.calendarview1);
    linear42 =  findViewById(R.id.linear42);
    textview25 = (TextView) findViewById(R.id.textview25);
    button1 = (Button) findViewById(R.id.button1);
    linrMainMeal =  findViewById(R.id.linrMainMeal);
    linrAddMeal =  findViewById(R.id.linrAddMeal);
    vscroll1 =  findViewById(R.id.vscroll1);
    linear19 =  findViewById(R.id.linear19);
    linear41 =  findViewById(R.id.linear41);
    linrBreakfast =  findViewById(R.id.linrBreakfast);
    linrLunch =  findViewById(R.id.linrLunch);
    linrSnack =  findViewById(R.id.linrSnack);
    linrDinner =  findViewById(R.id.linrDinner);
    btnSave = (MaterialButton) findViewById(R.id.btnSave);
    imageview13 =  findViewById(R.id.imageview13);
    textview23 = (TextView) findViewById(R.id.textview23);
    txtDate = (TextView) findViewById(R.id.txtDate);
    linear9 =  findViewById(R.id.linear9);
    linear11 =  findViewById(R.id.linear11);
    imgBreakfast =  findViewById(R.id.imgBreakfast);
    textview7 = (TextView) findViewById(R.id.textview7);
    imageview12 =  findViewById(R.id.imageview12);
    linear13 =  findViewById(R.id.linear13);
    linear12 =  findViewById(R.id.linear12);
    imageview3 =  findViewById(R.id.imageview3);
    txtBreakfastTitle = (TextView) findViewById(R.id.txtBreakfastTitle);
    linear37 =  findViewById(R.id.linear37);
    txtBreakfastClass = (TextView) findViewById(R.id.txtBreakfastClass);
    textview28 = (TextView) findViewById(R.id.textview28);
    txtKCal = (TextView) findViewById(R.id.txtKCal);
    linear31 =  findViewById(R.id.linear31);
    linear32 =  findViewById(R.id.linear32);
    imgLunch =  findViewById(R.id.imgLunch);
    textview16 = (TextView) findViewById(R.id.textview16);
    imageview14 =  findViewById(R.id.imageview14);
    linear33 =  findViewById(R.id.linear33);
    linear34 =  findViewById(R.id.linear34);
    imageview9 =  findViewById(R.id.imageview9);
    txtLunchTitle = (TextView) findViewById(R.id.txtLunchTitle);
    linear38 =  findViewById(R.id.linear38);
    txtLunchClass = (TextView) findViewById(R.id.txtLunchClass);
    textview29 = (TextView) findViewById(R.id.textview29);
    txtLunchKcal = (TextView) findViewById(R.id.txtLunchKcal);
    linear26 =  findViewById(R.id.linear26);
    linear27 =  findViewById(R.id.linear27);
    imgSnack =  findViewById(R.id.imgSnack);
```

```
      textview13 = (TextView) findViewById(R.id.textview13);
      imageview15 =  findViewById(R.id.imageview15);
      linear28 =  findViewById(R.id.linear28);
      linear29 =  findViewById(R.id.linear29);
      imageview7 =  findViewById(R.id.imageview7);
      txtSnackTitle = (TextView) findViewById(R.id.txtSnackTitle);
      linear39 =  findViewById(R.id.linear39);
      txtSnckClass = (TextView) findViewById(R.id.txtSnckClass);
      textview30 = (TextView) findViewById(R.id.textview30);
      txtSnackKcal = (TextView) findViewById(R.id.txtSnackKcal);
      linear21 =  findViewById(R.id.linear21);
      linear22 =  findViewById(R.id.linear22);
      imgDinner =  findViewById(R.id.imgDinner);
      textview10 = (TextView) findViewById(R.id.textview10);
      imageview16 =  findViewById(R.id.imageview16);
      linear23 =  findViewById(R.id.linear23);
      linear24 =  findViewById(R.id.linear24);
      imageview5 =  findViewById(R.id.imageview5);
      txtDinnerTitle = (TextView) findViewById(R.id.txtDinnerTitle);
      linear40 =  findViewById(R.id.linear40);
      txtDinnerClass = (TextView) findViewById(R.id.txtDinnerClass);
      textview31 = (TextView) findViewById(R.id.textview31);
      txtDinnerKcal = (TextView) findViewById(R.id.txtDinnerKcal);
      linear36 =  findViewById(R.id.linear36);
      linrClose =  findViewById(R.id.linrClose);
      txtMealReceiver = (TextView) findViewById(R.id.txtMealReceiver);
      linrSearch =  findViewById(R.id.linrSearch);
      notes_list = (ListView) findViewById(R.id.notes_list);
      textview22 = (TextView) findViewById(R.id.textview22);
      imageview10 =  findViewById(R.id.imageview10);
      imageview11 =  findViewById(R.id.imageview11);
      textview21 = (TextView) findViewById(R.id.textview21);
      Settings = getSharedPreferences("Settings", Activity.MODE_PRIVATE);
      AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);
      Delete_Dialog = new AlertDialog.Builder(this);
      meals = getSharedPreferences("meals", Activity.MODE_PRIVATE);
      save = new AlertDialog.Builder(this);

      calendarview1.setOnDateChangeListener(new
CalendarView.OnDateChangeListener() {
         @Override
         public void onSelectedDayChange(CalendarView param1, int param2,
int param3, int param4) {
            final int year = param2;
            final int month = param3;
            final int day = param4;
            if (day == Double.parseDouble(new
SimpleDateFormat("dd").format(Cal.getTime()))) {
               txtDate.setText("Today");
            }
            else {
               txtDate.setText(String.valueOf((long)(day)));
            }
         }
      });

      button1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            show++;
            if (show == 1) {
```

```
                calendarview1.setVisibility(View.VISIBLE);
                button1.setText("Minimize");
            }
            else {
                if (show == 2) {
                    calendarview1.setVisibility(View.GONE);
                    button1.setText("Open");
                    show = 0;
                }
            }
        }
    });

    linrBreakfast.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            breakfast = true;
            txtMealReceiver.setText("Add Breakfast");
            linrMainMeal.setVisibility(View.GONE);
            linrAddMeal.setVisibility(View.VISIBLE);

            imgBreakfast.setImageResource(R.drawable.circle_slice);
        }
    });

    linrLunch.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            txtMealReceiver.setText("Add Lunch");
            linrMainMeal.setVisibility(View.GONE);
            linrAddMeal.setVisibility(View.VISIBLE);

            imgLunch.setImageResource(R.drawable.circle_slice);
            lunch = true;
        }
    });

    linrSnack.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            txtMealReceiver.setText("Add Snack");
            linrMainMeal.setVisibility(View.GONE);
            linrAddMeal.setVisibility(View.VISIBLE);

            imgSnack.setImageResource(R.drawable.circle_slice);
            snack = true;
        }
    });

    linrDinner.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            txtMealReceiver.setText("Add Dinner");
            linrMainMeal.setVisibility(View.GONE);
            linrAddMeal.setVisibility(View.VISIBLE);

            imgDinner.setImageResource(R.drawable.circle_slice);
            dinner = true;
        }
    });
```

```
        btnSave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Cal = Calendar.getInstance();
                if (empty) {
                    if (linrMainMeal.getVisibility() == View.VISIBLE) {
                        meal_list.get((int)Position).put("Breakfast",
txtBreakfastTitle.getText().toString());
                        meal_list.get((int)Position).put("Breakfastclass",
txtBreakfastClass.getText().toString());
                        meal_list.get((int)Position).put("Breakfastkcal",
txtKCal.getText().toString());
                        meal_list.get((int)Position).put("Lunch",
txtLunchTitle.getText().toString());
                        meal_list.get((int)Position).put("Lunchclass",
txtLunchClass.getText().toString());
                        meal_list.get((int)Position).put("Lunchkkcal",
txtLunchKcal.getText().toString());
                        meal_list.get((int)Position).put("Snack",
txtSnackTitle.getText().toString());
                        meal_list.get((int)Position).put("Snackclass",
txtSnckClass.getText().toString());
                        meal_list.get((int)Position).put("Snackkcal",
txtSnackKcal.getText().toString());
                        meal_list.get((int)Position).put("Dinner",
txtDinnerTitle.getText().toString());
                        meal_list.get((int)Position).put("Dinnerclass",
txtDinnerClass.getText().toString());
                        meal_list.get((int)Position).put("Dinnerkcal",
txtDinnerKcal.getText().toString());
                        meal_list.get((int)Position).put("Time", new
SimpleDateFormat("hh:mm a").format(Cal.getTime()));
                        TastyToast.makeText(getApplicationContext(), "Your Meal
Have Been Successfuly Edited ", TastyToast.LENGTH_LONG,
TastyToast.SUCCESS);
                        meals.edit().putString("Meals", new
Gson().toJson(meal_list)).commit();
                        finish();
                    }
                    else {
                        finish();
                    }
                }
                else {
                    ToastUtil.showMessage(getApplicationContext(), "Can't save
empty meal plan");
                }
            }
        });

        linrClose.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                linrMainMeal.setVisibility(View.VISIBLE);
                linrAddMeal.setVisibility(View.GONE);
                clickAnimation(linrClose);
            }
        });

        linrSearch.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
         public void onClick(View view) {
            startActivity(new Intent(MealEditActivity.this,
SearchscanmealActivity.class));
Animatoo.animateFade(MealEditActivity.this);
         }
      });

      notes_list.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
         @Override
         public void onItemClick(AdapterView<?> param1, View param2, int
param3, long param4) {
            final int position = param3;
            empty = true;
            if (Selected) {
               if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Select").toString().equals("True")) {
                  Notes_List.get((int)(Notes_List.size() - 1) -
position).put("Select", "False");
                  Selected_Length--;
                  IsSelectAll = false;
               }
               else {
                  Notes_List.get((int)(Notes_List.size() - 1) -
position).put("Select", "True");
                  Selected_Length++;
               }
               refresh(Notes_List);
               _Toolbar(Selected_Length);
               if (Selected_Length == 0) {
                  IsSelectAll = false;


                  Selected = false;

((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
               }
            }
            else {
               Get_Notes = Notes_List.get((int)(Notes_List.size() - 1) -
position);
               linrMainMeal.setVisibility(View.VISIBLE);
               linrAddMeal.setVisibility(View.GONE);

               if (breakfast) {

txtBreakfastTitle.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString());

txtBreakfastClass.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Class").toString());
                  txtKCal.setText(" | kCal:
".concat(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Calories").toString()));
                  BreakfastServing = Notes_List.get((int)(Notes_List.size()
- 1) - position).get("Serving").toString();
                  breakfast = false;
               }
               else {
                  if (lunch) {
```

```
txtLunchTitle.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString());

txtLunchClass.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Class").toString());
                    txtLunchKcal.setText("  | kCal:
".concat(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Calories").toString()));
                    lunch = false;
                    SnackServing = Notes_List.get((int)(Notes_List.size()
- 1) - position).get("Serving").toString();
                }
                else {
                    if (snack) {

txtSnackTitle.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString());

txtSnckClass.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Class").toString());
                        txtSnackKcal.setText(" |
kCal:".concat(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Calories").toString()));
                        snack = false;
                        SnackServing =
Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Serving").toString();
                    }
                    else {
                        if (dinner) {

txtDinnerTitle.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString());

txtDinnerClass.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Class").toString());
                            txtDinnerKcal.setText(" |
kCal:".concat(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Calories").toString()));
                            dinner = false;
                            DinnerSnack =
Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Serving").toString();
                        }
                    }
                }
            }
        }
        if (Notes_List.size() == 0) {

com.google.android.material.snackbar.Snackbar.make(linrAddMeal, "No meal is
found to add",
com.google.android.material.snackbar.Snackbar.LENGTH_SHORT).setAction("Crea
te Meal", new View.OnClickListener(){
                @Override
                public void onClick(View view) {
                    startActivity(new Intent(MealEditActivity.this,
AddMealActivity.class)); Animatoo.animateFade(MealEditActivity.this);
                }
            }).show();
        }
```

```
            else {

            }
         }
      });
   }

   private void initializeLogic() {
      breakfast = false;
      lunch = false;
      snack = false;
      dinner = false;
      show = 0;
      calendarview1.setVisibility(View.GONE);
      linrClose.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)60, 0xFFFFFFFF));
      notes_list.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)30, 0xFFFFFFFF));
      linrSearch.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)30, 0xFFFFFFFF));
      linrBreakfast.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFFFFFFFF));
      linrLunch.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFFFFFFFF));
      linrSnack.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFFFFFFFF));
      linrDinner.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFFFFFFFF));
      btnSave.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFF3F51B6));
      changeActivityFont("railway_semisolid");
      removeScrollBar(notes_list);
      getSettingsData();
      ONCREATE();


      if (Build.VERSION.SDK_INT > Build.VERSION_CODES.KITKAT) {
         Window w =MealEditActivity.this.getWindow();
         w.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);

w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
w.setStatusBarColor(0xFFFFFFFF);
      }
      empty = false;
      if (!AllMeals.getString("Meals", "").equals("")) {
         Notes_List = new Gson().fromJson(AllMeals.getString("Meals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
         notes_list.setAdapter(new Notes_listAdapter(Notes_List));
         ((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
      }
   }

   @Override
```

```
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   @Override
   public void onBackPressed() {
      if (Selected) {
         _UnSelectAll();
      }
      else {
         save.setTitle("Warning!!!");
         save.setMessage("Do you want to exit without saving");
         save.setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
               finish();
            }
         });
         save.setNegativeButton("No", new DialogInterface.OnClickListener()
{
            @Override
            public void onClick(DialogInterface dialog, int which) {

            }
         });
         save.create().show();
      }
   }

   @Override
   public void onResume() {
      super.onResume();

   }
   public void setRadiusToView (final View view, final double radius, final
String Colour) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
gd.setColor(Color.parseColor(Colour)); gd.setCornerRadius((int)radius);
view.setBackground(gd);
   }


   public void setRadiusToView (final View view, final double value) {
      view.setElevation((float)value);
   }


   public void corners (final View view, final String _color1, final String
_color2, final double msgtr, final double _n1, final double _n2, final
double _n3, final double _n4) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
```

```
      gd.setColor(Color.parseColor(_color1));

      gd.setStroke((int)msgtr, Color.parseColor(_color2));

      gd.setCornerRadii(new
float[]{(int)_n1,(int)_n1,(int)_n2,(int)_n2,(int)_n3,(int)_n3,(int)_n4,(int
)_n4});

      view.setBackground(gd);

      view.setElevation(4);
   }


   public void Add (final String Colour, final ImageView Imageview) {
      Imageview.getDrawable().setColorFilter(Color.parseColor(Colour),
PorterDuff.Mode.SRC_IN);
   }


   public void removeScrollBar (final View view) {
      view.setVerticalScrollBarEnabled(false);
view.setHorizontalScrollBarEnabled(false);
   }


   public void getSettingsData () {

   }


   public void textSize (final TextView TextView1, final double size) {
      int j = (int) size;
      TextView1.setTextSize(j);
   }


   public void _Toolbar (final double _Length) {

   }


   public void _UnSelectAll () {
      Selected = false;
      Selected_Length = 0;


      IsSelectAll = false;
      Position = 0;
      for(int i = 0; i < (int)(Notes_List.size()); i++) {
         if
(Notes_List.get((int)Position).get("Select").toString().equals("True")) {
            Notes_List.get((int)Position).put("Select", "False");
         }
         Position++;
      }
      refresh(Notes_List);
   }


   public void refresh (final ArrayList<HashMap<String, Object>> ListMap) {
```

```
      Parcelable state =
      notes_list.onSaveInstanceState();
      notes_list.setAdapter(new Notes_listAdapter(ListMap));
      ((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
      notes_list.onRestoreInstanceState(state);
   }


   public void selectAll () {
      if (IsSelectAll) {
         _UnSelectAll();
         IsSelectAll = false;
      }
      else {
         IsSelectAll = true;
         Position = 0;
         for(int i = 0; i < (int)(Notes_List.size()); i++) {
            if
(Notes_List.get((int)Position).get("Select").toString().equals("False")) {
               Notes_List.get((int)Position).put("Select", "True");
               Selected_Length++;
            }
            Position++;
            _Toolbar(Selected_Length);
         }
         refresh(Notes_List);
      }
   }


   public void saveData () {
      temp_maplist = new Gson().fromJson(AllMeals.getString("Meals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      if (Notes_List.size() == 0) {
         if (getIntent().getStringExtra("Type").equals("All")) {
            AllMeals.edit().putString("Meals", new
Gson().toJson(Notes_List)).commit();
         }
         else {
            if (getIntent().getStringExtra("Type").equals("Important")) {
               Position = temp_maplist.size() - 1;
               for(int i = 0; i < (int)(temp_maplist.size()); i++) {
                  if
(temp_maplist.get((int)Position).get("Important").toString().equals("True")
) {
                     temp_maplist.remove((int)(Position));
                  }
                  Position--;
               }
               AllMeals.edit().putString("Meals", new
Gson().toJson(temp_maplist)).commit();
            }
         }
      }
      else {
         AllMeals.edit().putString("Meals", new
Gson().toJson(Notes_List)).commit();
      }
   }
```

```
   public void changeActivityFont (final String fontname) {
      fontName = "fonts/".concat(fontname.concat(".ttf"));
      overrideFonts(this,getWindow().getDecorView());
   }
   private void overrideFonts(final android.content.Context context, final
View v) {

      try {
         Typeface
         typeace = Typeface.createFromAsset(getAssets(), fontName);;
         if ((v instanceof ViewGroup)) {
            ViewGroup vg = (ViewGroup) v;
            for (int i = 0;
            i < vg.getChildCount();
            i++) {
               View child = vg.getChildAt(i);
               overrideFonts(context, child);
            }
         }
         else {
            if ((v instanceof TextView)) {
               ((TextView) v).setTypeface(typeace);
            }
            else {
               if ((v instanceof EditText )) {
                  ((EditText) v).setTypeface(typeace);
               }
               else {
                  if ((v instanceof Button)) {
                     ((Button) v).setTypeface(typeace);
                  }
               }
            }
         }
      }
      catch(Exception e)

      {

      };
   }


   public void clickAnimation (final View view) {
      ScaleAnimation fade_in = new ScaleAnimation(0.9f, 1f, 0.9f, 1f,
Animation.RELATIVE_TO_SELF, 0.5f, Animation.RELATIVE_TO_SELF, 0.7f);
      fade_in.setDuration(300);
      fade_in.setFillAfter(true);
      view.startAnimation(fade_in);
   }


   public void ONCREATE () {
      Position =
Double.parseDouble(getIntent().getStringExtra("Position"));
      Cal = Calendar.getInstance();
      if (!meals.getString("Meals", "").equals("")) {
         meal_list = new Gson().fromJson(meals.getString("Meals", ""), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      }
      Date = getIntent().getStringExtra("Date");
```

```
      Time = new SimpleDateFormat("hh:mm a").format(Cal.getTime());
      _GetMealData();
   }


   public void _GetMealData () {
      Get_Notes = meal_list.get((int)Position);
      txtBreakfastTitle.setText(Get_Notes.get("Breakfast").toString());

txtBreakfastClass.setText(Get_Notes.get("Breakfastclass").toString());
      txtKCal.setText(Get_Notes.get("Breakfastkcal").toString());
      txtLunchTitle.setText(Get_Notes.get("Lunch").toString());
      txtLunchClass.setText(Get_Notes.get("Lunchclass").toString());
      txtLunchKcal.setText(Get_Notes.get("Lunchkkcal").toString());
      txtSnackTitle.setText(Get_Notes.get("Snack").toString());
      txtSnckClass.setText(Get_Notes.get("Snackclass").toString());
      txtSnackKcal.setText(Get_Notes.get("Snackkcal").toString());
      txtDinnerTitle.setText(Get_Notes.get("Dinner").toString());
      txtDinnerClass.setText(Get_Notes.get("Dinnerclass").toString());
      txtDinnerKcal.setText(Get_Notes.get("Dinnerkcal").toString());
      txtDate.setText(Get_Notes.get("Date").toString());
   }


   public class Notes_listAdapter extends BaseAdapter {
      ArrayList<HashMap<String, Object>> data;
      public Notes_listAdapter(ArrayList<HashMap<String, Object>> arr) {
         data = arr;
      }

      @Override
      public int getCount() {
         return data.size();
      }

      @Override
      public HashMap<String, Object> getItem(int index) {
         return data.get(index);
      }

      @Override
      public long getItemId(int index) {
         return index;
      }
      @Override
      public View getView(final int position, View v, ViewGroup container)
{
         LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
         View view = v;
         if (view == null) {
            view = inflater.inflate(R.layout.custom_meal_list, null);
         }

         final LinearLayout bg =  view.findViewById(R.id.bg);
         final LinearLayout linear1 =  view.findViewById(R.id.linear1);
         final ImageView imageview1 =  view.findViewById(R.id.imageview1);
         final LinearLayout main =  view.findViewById(R.id.main);
         final LinearLayout side_lin =  view.findViewById(R.id.side_lin);
         final LinearLayout text_lin =  view.findViewById(R.id.text_lin);
```

```
        final LinearLayout title_time_lin =
view.findViewById(R.id.title_time_lin);
        final LinearLayout note_lin =  view.findViewById(R.id.note_lin);
        final TextView date = (TextView) view.findViewById(R.id.date);
        final LinearLayout linear2 =  view.findViewById(R.id.linear2);
        final TextView time = (TextView) view.findViewById(R.id.time);
        final TextView title = (TextView) view.findViewById(R.id.title);
        final TextView txtFoodClasses = (TextView)
view.findViewById(R.id.txtFoodClasses);
        final TextView note = (TextView) view.findViewById(R.id.note);
        final TextView txt_methods = (TextView)
view.findViewById(R.id.txt_methods);
        final TextView txt_calories = (TextView)
view.findViewById(R.id.txt_calories);
        final TextView txtminutes = (TextView)
view.findViewById(R.id.txtminutes);
        final TextView textview4 = (TextView)
view.findViewById(R.id.textview4);
        final TextView txtServing = (TextView)
view.findViewById(R.id.txtServing);

        linear1.setVisibility(View.GONE);

date.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
ium.ttf"), Typeface.NORMAL);

title.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansbo
ld.ttf"), Typeface.BOLD);

time.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
ium.ttf"), Typeface.BOLD);

note.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
ium.ttf"), Typeface.BOLD);
        side_lin.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)60, 0xFF3F51B6));
        if (Selected) {

imageview1.setImageResource(R.drawable.ic_radio_button_off_black);
            if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Select").toString().equals("True")) {
                imageview1.setVisibility(View.VISIBLE);

imageview1.setImageResource(R.drawable.ic_radio_button_on_black);
            }
            else {

imageview1.setImageResource(R.drawable.ic_radio_button_off_black);
            }
        }
        else {
            imageview1.setVisibility(View.GONE);
        }
        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Title")) {
            if (Settings.getString("Text Size", "").equals("Small")) {
               if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString().length() > 28) {
                  title.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Title").toString().substring((int)(0),
```

```
(int)(28)).concat("..."));
                }
                else {
                   title.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Title").toString());
                }
             }
             else {
                if (Settings.getString("Text Size", "").equals("Medium")) {
                   if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString().length() > 24) {
                      title.setText(Notes_List.get((int)(Notes_List.size() -
1) - position).get("Title").toString().substring((int)(0),
(int)(24)).concat("..."));
                   }
                   else {
                      title.setText(Notes_List.get((int)(Notes_List.size() -
1) - position).get("Title").toString());
                   }
                }
                else {
                   if (Settings.getString("Text Size", "").equals("Large"))
{
                      if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString().length() > 22) {

title.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString().substring((int)(0),
(int)(22)).concat("..."));
                      }
                      else {

title.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString());
                      }
                   }
                   else {

                   }
                }
             }
          }
          if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Time")) {
             time.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Time").toString());
          }
          if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Date")) {
             date.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Date").toString());
          }
          if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Method")) {
             note.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Method").toString());
          }
          if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Note")) {
             txt_methods.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Note").toString());
```

```
            }
            if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Calories")) {
                txt_calories.setText(Notes_List.get((int)(Notes_List.size() -
1) - position).get("Calories").toString());
            }
            if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Minutes")) {
                txtminutes.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Minutes").toString());
            }
            if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Ingredients")) {
                txt_methods.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Ingredients").toString());
            }
            if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Class")) {
                txtFoodClasses.setText(Notes_List.get((int)(Notes_List.size() -
1) - position).get("Class").toString());
            }
            note.setVisibility(View.GONE);

            return view;
        }
    }

    @Deprecated
    public void showMessage(String msg) {
        Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
    }

    @Deprecated
    public int getLocationX(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
```

```
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Meal Planner List Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import android.graphics.drawable.GradientDrawable;
import android.os.Bundle;
import android.os.Parcelable;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
```

```
import
com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import com.sdsmdg.tastytoast.TastyToast;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;


public class MealPlannerListActivity extends AppCompatActivity {

   private FloatingActionButton fab;
   private double Position = 0;
   private boolean Selected = false;
   private double Selected_Length = 0;
   private boolean IsSelectAll = false;
   private String fontName = "";
   private String typeace = "";

   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> temp_maplist = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> meal_list = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> temp_mealslist = new
ArrayList<>();

   private LinearLayout toolbar;
   private LinearLayout main;
   private ImageView back_img;
   private TextView toolbar_txt;
   private ImageView select_all_img;
   private ListView notes_list;

   private Intent To_Noteview = new Intent();
   private SharedPreferences Settings;
   private SharedPreferences AllMeals;
   private AlertDialog.Builder Delete_Dialog;
   private SharedPreferences meal;
   private SharedPreferences meals;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.meal_planner_list);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      fab = (FloatingActionButton) findViewById(R.id.fab);

      toolbar =  findViewById(R.id.toolbar);
      main =  findViewById(R.id.main);
      back_img =  findViewById(R.id.back_img);
      toolbar_txt = (TextView) findViewById(R.id.toolbar_txt);
      select_all_img =  findViewById(R.id.select_all_img);
```

```
      notes_list = (ListView) findViewById(R.id.notes_list);
      Settings = getSharedPreferences("Settings", Activity.MODE_PRIVATE);
      AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);
      Delete_Dialog = new AlertDialog.Builder(this);
      meal = getSharedPreferences("meal", Activity.MODE_PRIVATE);
      meals = getSharedPreferences("meals", Activity.MODE_PRIVATE);

      back_img.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            finish();
         }
      });

      select_all_img.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            selectAll();
         }
      });

      notes_list.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
         @Override
         public void onItemClick(AdapterView<?> param1, View param2, int
param3, long param4) {
            final int position = param3;
            if (Selected) {
               if (meal_list.get((int)(meal_list.size() - 1) -
position).get("Select").toString().equals("True")) {
                  meal_list.get((int)(meal_list.size() - 1) -
position).put("Select", "False");
                  Selected_Length--;
                  IsSelectAll = false;
               }
               else {
                  meal_list.get((int)(meal_list.size() - 1) -
position).put("Select", "True");
                  Selected_Length++;
               }
               refresh(Notes_List);
               _Toolbar(Selected_Length);
               if (Selected_Length == 0) {
                  IsSelectAll = false;
                  select_all_img.setEnabled(true);
                  select_all_img.setVisibility(View.INVISIBLE);
                  toolbar_txt.setText("Your Meals");
                  fab(false);
                  Selected = false;

((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
               }
            }
            else {
               To_Noteview.setAction(Intent.ACTION_VIEW);
               To_Noteview.setClass(getApplicationContext(),
ViewMealActivity.class);
               To_Noteview.putExtra("Position",
String.valueOf((long)((Notes_List.size() - 1) - position)));
               startActivity(To_Noteview);
               finish();
```

```
            }
        }
    });

    notes_list.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
        @Override
        public boolean onItemLongClick(AdapterView<?> param1, View param2,
int param3, long param4) {
            final int position = param3;
            if (Selected) {
               ToastUtil.showMessage(getApplicationContext(), "Selection
Mode Is Enabled");
            }
            else {
               Selected = true;
               meal_list.get((int)(meal_list.size() - 1) -
position).put("Select", "True");
               refresh(meal_list);
               fab(true);
               Selected_Length++;
               _Toolbar(Selected_Length);
               select_all_img.setVisibility(View.VISIBLE);
               select_all_img.setEnabled(true);
            }
            return true;
        }
    });

    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Delete_Dialog.setTitle("Delete ?");
            Delete_Dialog.setMessage("Are You Sure To Delete
".concat(String.valueOf((long)(Selected_Length)).concat(" Selected Items ?
This Action Cannot Be Undone.")));
            Delete_Dialog.setPositiveButton("Delete", new
DialogInterface.OnClickListener() {
               @Override
               public void onClick(DialogInterface dialog, int which) {
                  Selected = false;
                  Position = meal_list.size() - 1;
                  while(true) {
                     if (Position == -1) {
                        break;
                     }
                     else {
                        if
(meal_list.get((int)Position).get("Select").toString().equals("True")) {
                           meal_list.remove((int)(Position));
                        }
                        Position--;
                     }
                  }
                  refresh(Notes_List);
                  Selected_Length = 0;
                  toolbar_txt.setText("Your Meals");
                  fab(false);
                  select_all_img.setVisibility(View.INVISIBLE);
                  saveData();
                  TastyToast.makeText(getApplicationContext(), "Your
```

```
Planned Meals Have Been Successfuly Deleted ", TastyToast.LENGTH_LONG,
TastyToast.SUCCESS);
                }
            });
            Delete_Dialog.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    _UnSelectAll();
                }
            });
            Delete_Dialog.create().show();
        }
    });
  }

  private void initializeLogic() {
    changeActivityFont("railway_semisolid");
    removeScrollBar(notes_list);
    getSettingsData();
    fab(false);
    select_all_img.setVisibility(View.INVISIBLE);
    select_all_img.setEnabled(false);
    if (!meals.getString("Meals", "").equals("")) {
      meal_list = new Gson().fromJson(meals.getString("Meals", ""), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      notes_list.setAdapter(new Notes_listAdapter(meal_list));
      ((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
    }
  }

  @Override
  protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode) {

      default:
      break;
    }
  }

  @Override
  public void onBackPressed() {
    if (Selected) {
      _UnSelectAll();
    }
    else {
      finish();
    }
  }
  public void setRadiusToView (final View view, final double radius, final
String Colour) {
    android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
gd.setColor(Color.parseColor(Colour)); gd.setCornerRadius((int)radius);
view.setBackground(gd);
  }


  public void setRadiusToView (final View view, final double value) {
```

```
      view.setElevation((float)value);
   }


   public void corners (final View view, final String _color1, final String
_color2, final double msgtr, final double _n1, final double _n2, final
double _n3, final double _n4) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();

      gd.setColor(Color.parseColor(_color1));

      gd.setStroke((int)msgtr, Color.parseColor(_color2));

      gd.setCornerRadii(new
float[]{(int)_n1,(int)_n1,(int)_n2,(int)_n2,(int)_n3,(int)_n3,(int)_n4,(int
)_n4});

      view.setBackground(gd);

      view.setElevation(4);
   }


   public void Add (final String Colour, final ImageView Imageview) {
      Imageview.getDrawable().setColorFilter(Color.parseColor(Colour),
PorterDuff.Mode.SRC_IN);
   }


   public void removeScrollBar (final View view) {
      view.setVerticalScrollBarEnabled(false);
view.setHorizontalScrollBarEnabled(false);
   }


   public void getSettingsData () {
      if (Settings.getString("Text Size", "").equals("Small")) {

      }
      else {
         if (Settings.getString("Text Size", "").equals("Small")) {
            textSize(toolbar_txt, 20);
         }
         else {
            if (Settings.getString("Text Size", "").equals("Large")) {
               textSize(toolbar_txt, 22);
            }
            else {

            }
         }
      }
   }


   public void textSize (final TextView TextView1, final double size) {
      int j = (int) size;
      TextView1.setTextSize(j);
   }
```

```
   public void fab (final boolean visibility) {
      if (visibility) {
         fab.show();
      }
      else {
         fab.hide();
      }
   }


   public void _Toolbar (final double _Length) {

toolbar_txt.setText(String.valueOf((long)(_Length)).concat("/".concat(Strin
g.valueOf((long)(meal_list.size())).concat(" Selected"))));
   }


   public void _UnSelectAll () {
      Selected = false;
      Selected_Length = 0;
      toolbar_txt.setText("Your Meals");
      select_all_img.setVisibility(View.INVISIBLE);
      select_all_img.setEnabled(false);
      IsSelectAll = false;
      Position = 0;
      for(int i = 0; i < (int)(meal_list.size()); i++) {
         if
(meal_list.get((int)Position).get("Select").toString().equals("True")) {
            meal_list.get((int)Position).put("Select", "False");
         }
         Position++;
      }
      refresh(meal_list);
      fab(false);
   }


   public void refresh (final ArrayList<HashMap<String, Object>> ListMap) {
      Parcelable state =
      notes_list.onSaveInstanceState();
      notes_list.setAdapter(new Notes_listAdapter(ListMap));
      ((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
      notes_list.onRestoreInstanceState(state);
   }


   public void selectAll () {
      if (IsSelectAll) {
         _UnSelectAll();
         IsSelectAll = false;
      }
      else {
         IsSelectAll = true;
         Position = 0;
         for(int i = 0; i < (int)(meal_list.size()); i++) {
            if
(meal_list.get((int)Position).get("Select").toString().equals("False")) {
               meal_list.get((int)Position).put("Select", "True");
               Selected_Length++;
            }
```

```
            Position++;
            _Toolbar(Selected_Length);
         }
         refresh(meal_list);
      }
   }


   public void saveData () {

   }


   public void changeActivityFont (final String fontname) {
      fontName = "fonts/".concat(fontname.concat(".ttf"));
      overrideFonts(this,getWindow().getDecorView());
   }
   private void overrideFonts(final android.content.Context context, final
View v) {

      try {
         Typeface
         typeace = Typeface.createFromAsset(getAssets(), fontName);;
         if ((v instanceof ViewGroup)) {
            ViewGroup vg = (ViewGroup) v;
            for (int i = 0;
            i < vg.getChildCount();
            i++) {
               View child = vg.getChildAt(i);
               overrideFonts(context, child);
            }
         }
         else {
            if ((v instanceof TextView)) {
               ((TextView) v).setTypeface(typeace);
            }
            else {
               if ((v instanceof EditText )) {
                  ((EditText) v).setTypeface(typeace);
               }
               else {
                  if ((v instanceof Button)) {
                     ((Button) v).setTypeface(typeace);
                  }
               }
            }
         }
      }
      catch(Exception e)

      {

      };
   }


   public class Notes_listAdapter extends BaseAdapter {
      ArrayList<HashMap<String, Object>> data;
      public Notes_listAdapter(ArrayList<HashMap<String, Object>> arr) {
         data = arr;
      }
```

```
      @Override
      public int getCount() {
         return data.size();
      }

      @Override
      public HashMap<String, Object> getItem(int index) {
         return data.get(index);
      }

      @Override
      public long getItemId(int index) {
         return index;
      }
      @Override
      public View getView(final int position, View v, ViewGroup container)
{
         LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
         View view = v;
         if (view == null) {
            view = inflater.inflate(R.layout.mealslist, null);
         }

         final LinearLayout bg =  view.findViewById(R.id.bg);
         final LinearLayout linear4 =  view.findViewById(R.id.linear4);
         final LinearLayout linear3 =  view.findViewById(R.id.linear3);
         final LinearLayout main =  view.findViewById(R.id.main);
         final ImageView imageview1 =  view.findViewById(R.id.imageview1);
         final ImageView imgReaction =
view.findViewById(R.id.imgReaction);
         final LinearLayout text_lin =  view.findViewById(R.id.text_lin);
         final LinearLayout title_time_lin =
view.findViewById(R.id.title_time_lin);
         final LinearLayout linear2 =  view.findViewById(R.id.linear2);
         final LinearLayout linear5 =  view.findViewById(R.id.linear5);
         final TextView title = (TextView) view.findViewById(R.id.title);
         final LinearLayout linear7 =  view.findViewById(R.id.linear7);
         final TextView txtCalories = (TextView)
view.findViewById(R.id.txtCalories);
         final TextView txtTotalkCal = (TextView)
view.findViewById(R.id.txtTotalkCal);
         final TextView time = (TextView) view.findViewById(R.id.time);
         final TextView date = (TextView) view.findViewById(R.id.date);
         final TextView txtBreakfast = (TextView)
view.findViewById(R.id.txtBreakfast);
         final TextView txtLunch = (TextView)
view.findViewById(R.id.txtLunch);
         final TextView txtSnack = (TextView)
view.findViewById(R.id.txtSnack);
         final TextView txtDinner = (TextView)
view.findViewById(R.id.txtDinner);

         linear4.setVisibility(View.GONE);
         linear3.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)100, 0xFFFFFFFF));
         main.setElevation((float)0);
         linear3.setElevation((float)10);
```

```
         main.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)10, 0xFFFFFFFF));
         if (Selected) {

imageview1.setImageResource(R.drawable.ic_radio_button_off_black);
            if (meal_list.get((int)(meal_list.size() - 1) -
position).get("Select").toString().equals("True")) {
               imageview1.setVisibility(View.VISIBLE);

imageview1.setImageResource(R.drawable.ic_radio_button_on_black);
            }
            else {

imageview1.setImageResource(R.drawable.ic_radio_button_off_black);
            }
         }
         else {
            imageview1.setVisibility(View.GONE);
         }
         if (meal_list.get((int)(meal_list.size() - 1) -
position).containsKey("Time")) {
            time.setText(meal_list.get((int)(meal_list.size() - 1) -
position).get("Time").toString());
         }
         if (meal_list.get((int)(meal_list.size() - 1) -
position).containsKey("Date")) {
            date.setText(meal_list.get((int)(meal_list.size() - 1) -
position).get("Date").toString());
         }
         if (meal_list.get((int)(meal_list.size() - 1) -
position).containsKey("Breakfast")) {
            txtBreakfast.setText(meal_list.get((int)(meal_list.size() - 1)
- position).get("Breakfast").toString());
         }
         if (meal_list.get((int)(meal_list.size() - 1) -
position).containsKey("Lunch")) {
            txtLunch.setText(meal_list.get((int)(meal_list.size() - 1) -
position).get("Lunch").toString());
         }
         if (meal_list.get((int)(meal_list.size() - 1) -
position).containsKey("Snack")) {
            txtSnack.setText(meal_list.get((int)(meal_list.size() - 1) -
position).get("Snack").toString());
         }
         if (meal_list.get((int)(meal_list.size() - 1) -
position).containsKey("Dinner")) {
            txtDinner.setText(meal_list.get((int)(meal_list.size() - 1) -
position).get("Dinner").toString());
         }
         if (txtBreakfast.getText().toString().equals("Breakfast Not
Entered")) {
            title.setText("Incomplete Meal:\nBreakfast is missing");
            imgReaction.setImageResource(R.drawable.emoticonsad);

         }
         else {
            if (txtLunch.getText().toString().equals("Lunch Not Entered"))
{
               title.setText("Incomplete Meal:\nLunch is missing");
               imgReaction.setImageResource(R.drawable.emoticonsad);
```

```
            }
            else {
               if (txtSnack.getText().toString().equals("Snack Not
Entered")) {
                  title.setText("Incomplete Meal:\nSnack is missing");
                  imgReaction.setImageResource(R.drawable.emoticonsad);
               }
               else {
                  if (txtDinner.getText().toString().equals("Dinner Not
Entered")) {
                     title.setText("Incomplete Meal:\nDinner is missing");
                     imgReaction.setImageResource(R.drawable.emoticonsad);
                  }
                  else {
                     title.setText("Complete Meal ");
                     imgReaction.setImageResource(R.drawable.emoticoncool);
                  }
               }
            }
         }

         return view;
      }
   }

   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }

   @Deprecated
   public int getRandom(int min, int max) {
      Random random = new Random();
      return random.nextInt(max - min + 1) + min;
   }

   @Deprecated
   public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
      ArrayList<Double> result = new ArrayList<Double>();
      SparseBooleanArray arr = _list.getCheckedItemPositions();
      for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
         if (arr.valueAt(iIdx))
         result.add((double)arr.keyAt(iIdx));
      }
      return result;
```

```
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Meals Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.animation.ObjectAnimator;
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import android.os.Build;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;

import
com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;
import java.util.Timer;
```

```
import java.util.TimerTask;


public class MealsActivity extends AppCompatActivity {
   private Timer timer = new Timer();

   private FloatingActionButton fab;
   private String App_Theme = "";
   private double Position = 0;
   private HashMap<String, Object> GetNotes = new HashMap<>();
   private String fontName = "";
   private String typeace = "";

   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> Important_Notes_List = new
ArrayList<>();

   private LinearLayout main;
   private LinearLayout up_lin;
   private LinearLayout btm_lin;
   private LinearLayout toolbar;
   private LinearLayout length_lin;
   private LinearLayout toolbar_txt_lin;
   private ImageView ic_settings;
   private ImageView imageview1;
   private TextView welcome_title;
   private LinearLayout notes_im_lin;
   private LinearLayout notes_lin;
   private LinearLayout important_lin;
   private TextView notes_length_title;
   private TextView notes_length;
   private TextView important_len_title;
   private TextView important_length;
   private TextView recent_title;
   private LinearLayout notes_list_lin;
   private LinearLayout notes_list_lin_main;
   private LinearLayout empty_lin;
   private LinearLayout note_main;
   private LinearLayout side_lin;
   private LinearLayout text_lin;
   private LinearLayout title_time_lin;
   private LinearLayout note_lin;
   private TextView date;
   private TextView title;
   private TextView time;
   private TextView note;
   private ImageView empty_img;
   private TextView empty_txt;

   private SharedPreferences AllMeals;
   private TimerTask Anim;
   private Intent To_Add_New = new Intent();
   private Intent To_Notes = new Intent();
   private Intent Tosettings = new Intent();
   private Intent To_Noteview = new Intent();
   private SharedPreferences Settingsetup;
   private SharedPreferences Settings;

   @RequiresApi(api = Build.VERSION_CODES.N_MR1)
   @Override
```

```
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.meals);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      fab = (FloatingActionButton) findViewById(R.id.fab);

      main =  findViewById(R.id.main);
      up_lin =  findViewById(R.id.up_lin);
      btm_lin =  findViewById(R.id.btm_lin);
      toolbar =  findViewById(R.id.toolbar);
      length_lin =  findViewById(R.id.length_lin);
      toolbar_txt_lin =  findViewById(R.id.toolbar_txt_lin);
      ic_settings =  findViewById(R.id.ic_settings);
      imageview1 =  findViewById(R.id.imageview1);
      welcome_title = (TextView) findViewById(R.id.welcome_title);
      notes_im_lin =  findViewById(R.id.notes_im_lin);
      notes_lin =  findViewById(R.id.notes_lin);
      important_lin =  findViewById(R.id.important_lin);
      notes_length_title = (TextView)
findViewById(R.id.notes_length_title);
      notes_length = (TextView) findViewById(R.id.notes_length);
      important_len_title = (TextView)
findViewById(R.id.important_len_title);
      important_length = (TextView) findViewById(R.id.important_length);
      recent_title = (TextView) findViewById(R.id.recent_title);
      notes_list_lin =  findViewById(R.id.notes_list_lin);
      notes_list_lin_main =  findViewById(R.id.notes_list_lin_main);
      empty_lin =  findViewById(R.id.empty_lin);
      note_main =  findViewById(R.id.note_main);
      side_lin =  findViewById(R.id.side_lin);
      text_lin =  findViewById(R.id.text_lin);
      title_time_lin =  findViewById(R.id.title_time_lin);
      note_lin =  findViewById(R.id.note_lin);
      date = (TextView) findViewById(R.id.date);
      title = (TextView) findViewById(R.id.title);
      time = (TextView) findViewById(R.id.time);
      note = (TextView) findViewById(R.id.note);
      empty_img =  findViewById(R.id.empty_img);
      empty_txt = (TextView) findViewById(R.id.empty_txt);
      AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);
      Settingsetup = getSharedPreferences("Settings Set Up",
Activity.MODE_PRIVATE);
      Settings = getSharedPreferences("Settings", Activity.MODE_PRIVATE);

      ic_settings.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            Animator(ic_settings, "scaleX", 1.1d, 200);
            Animator(ic_settings, "scaleY", 1.1d, 200);
            Anim = new TimerTask() {
               @Override
               public void run() {
                  runOnUiThread(new Runnable() {
                     @Override
                     public void run() {
                        Animator(ic_settings, "scaleX", 1, 200);
```

```
                        Animator(ic_settings, "scaleY", 1, 200);
                        Tosettings.setAction(Intent.ACTION_VIEW);
                        Tosettings.setClass(getApplicationContext(),
ProfileActivity.class);
                        startActivity(Tosettings);
                    }
                });
            }
        };
        timer.schedule(Anim, (int)(200));
    }
});

notes_lin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Animator(notes_lin, "scaleX", 1.05d, 200);
        Animator(notes_lin, "scaleY", 1.05d, 200);
        Anim = new TimerTask() {
            @Override
            public void run() {
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        Animator(notes_lin, "scaleX", 1, 200);
                        Animator(notes_lin, "scaleY", 1, 200);
                        To_Notes.setAction(Intent.ACTION_VIEW);
                        To_Notes.setClass(getApplicationContext(),
MealsListActivity.class);
                        To_Notes.putExtra("Data", new
Gson().toJson(Notes_List));
                        To_Notes.putExtra("Type", "All");
                        startActivity(To_Notes);
                    }
                });
            }
        };
        timer.schedule(Anim, (int)(200));
    }
});

important_lin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Animator(important_lin, "scaleX", 1.05d, 200);
        Animator(important_lin, "scaleY", 1.05d, 200);
        Anim = new TimerTask() {
            @Override
            public void run() {
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        Animator(important_lin, "scaleX", 1, 200);
                        Animator(important_lin, "scaleY", 1, 200);
                        To_Notes.setAction(Intent.ACTION_VIEW);
                        To_Notes.setClass(getApplicationContext(),
MealsListActivity.class);
                        To_Notes.putExtra("Data", new
Gson().toJson(Important_Notes_List));
                        To_Notes.putExtra("Type", "Important");
                        startActivity(To_Notes);
```

```
                    }
                });
            }
        };
        timer.schedule(Anim, (int)(200));
      }
    });

    note_main.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        To_Noteview.setAction(Intent.ACTION_VIEW);
        To_Noteview.setClass(getApplicationContext(),
ViewMealActivity.class);
        To_Noteview.putExtra("Position",
String.valueOf((long)(Notes_List.size() - 1)));
        startActivity(To_Noteview);
      }
    });

    fab.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        To_Add_New.setAction(Intent.ACTION_VIEW);
        To_Add_New.setClass(getApplicationContext(),
EntermealActivity.class);
        To_Add_New.putExtra("dT", "true");
        startActivity(To_Add_New);
      }
    });
  }

  @RequiresApi(api = Build.VERSION_CODES.N_MR1)
  private void initializeLogic() {
    App_Theme = "#3F51B6";
    _DynamicShortcutApp();
    _UIDESIGN();
    ripple(ic_settings);
    ONCREATE_LIST();
  }

  @Override
  protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode) {

      default:
      break;
    }
  }

  @Override
  public void onStart() {
    super.onStart();
    ONCREATE_LIST();
  }
  public void _UIDESIGN () {
    Add(App_Theme, empty_img);
    corners(side_lin, App_Theme, App_Theme, 0, 30, 30, 30, 30);
    setRadiusToView(note_main, 30, "#F6F7FB");
```

```
      setRadiusToView(note_main, 5);
      setRadiusToView(important_lin, 30, "#F6F7FB");
      setRadiusToView(notes_lin, 10);
      setRadiusToView(important_lin, 10);
      int[] colors = {Color.parseColor(App_Theme),
Color.parseColor("#3F51B6")};

      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable(android.graphics.drawable.Gradie
ntDrawable.Orientation.TR_BL, colors);
      gd.setCornerRadius(30f);
      notes_lin.setBackgroundDrawable(gd);
   }


   public void setRadiusToView (final View view, final double radius, final
String Colour) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
gd.setColor(Color.parseColor(Colour)); gd.setCornerRadius((int)radius);
view.setBackground(gd);
   }


   public void setRadiusToView (final View view, final double value) {
      view.setElevation((float)value);
   }


   public void Add (final String Colour, final ImageView Imageview) {
      Imageview.getDrawable().setColorFilter(Color.parseColor(Colour),
PorterDuff.Mode.SRC_IN);
   }


   public void corners (final View view, final String _color1, final String
_color2, final double msgtr, final double _n1, final double _n2, final
double _n3, final double _n4) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();

      gd.setColor(Color.parseColor(_color1));

      gd.setStroke((int)msgtr, Color.parseColor(_color2));

      gd.setCornerRadii(new
float[]{(int)_n1,(int)_n1,(int)_n2,(int)_n2,(int)_n3,(int)_n3,(int)_n4,(int
)_n4});

      view.setBackground(gd);

      view.setElevation(4);
   }


   public void ripple (final View view) {

      int[] attrs = new
int[]{android.R.attr.selectableItemBackgroundBorderless};
      android.content.res.TypedArray typedArray =
this.obtainStyledAttributes(attrs);
```

```
      int backgroundResource = typedArray.getResourceId(0, 0);
view.setBackgroundResource(backgroundResource);
      view.setClickable(true);
   }


   public void Animator (final View view, final String _propertyName, final
double value, final double duration) {
      ObjectAnimator anim = new ObjectAnimator();
      anim.setTarget(view);
      anim.setPropertyName(_propertyName);
      anim.setFloatValues((float)value);
      anim.setDuration((long)duration);
      anim.start();
   }


   public void ONCREATE_LIST () {
      _Preferances();
      getSettingsData();
      Typeface();
      if (!AllMeals.getString("Meals", "").equals("")) {
         Notes_List = new Gson().fromJson(AllMeals.getString("Meals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
         GetImportantNotes();
      }
      if (Notes_List.size() == 0) {
         notes_length.setText("0");
         important_length.setText("0");
         notes_list_lin_main.setVisibility(View.GONE);
         recent_title.setVisibility(View.GONE);
         empty_lin.setVisibility(View.VISIBLE);
      }
      else {
         GetRecentDataFrom(Notes_List.size() - 1);
         notes_length.setText(String.valueOf((long)(Notes_List.size())));

important_length.setText(String.valueOf((long)(Important_Notes_List.size())
));
         notes_list_lin_main.setVisibility(View.VISIBLE);
         recent_title.setVisibility(View.VISIBLE);
         empty_lin.setVisibility(View.GONE);
      }
   }


   public void _Preferances () {
      if (Settingsetup.getString("Settings Set Up", "").equals("")) {
         Settingsetup.edit().putString("Settings Set Up", "True").commit();
         Settings.edit().putString("Text Size", "Small").commit();
         Settings.edit().putString("Theme", "Default").commit();
         Settings.edit().putString("Detect Link", "True").commit();
      }
      else {

      }
   }


   public void getSettingsData () {
```

```
    }


    public void textSize (final TextView TextView1, final double size) {
        int j = (int) size;
        TextView1.setTextSize(j);
    }


    public void Typeface () {
        changeActivityFont("railway_semisolid");
    }


    public void GetRecentDataFrom (final double position) {
        if (Notes_List.get((int)position).containsKey("Title")) {
            if (Settings.getString("Text Size", "").equals("Small")) {
                if
(Notes_List.get((int)position).get("Title").toString().length() > 28) {

title.setText(Notes_List.get((int)position).get("Title").toString().substri
ng((int)(0), (int)(28)).concat("..."));
                }
                else {

title.setText(Notes_List.get((int)position).get("Title").toString());
                }
            }
            else {
                if (Settings.getString("Text Size", "").equals("Medium")) {
                    if
(Notes_List.get((int)position).get("Title").toString().length() > 24) {

title.setText(Notes_List.get((int)position).get("Title").toString().substri
ng((int)(0), (int)(24)).concat("..."));
                    }
                    else {

title.setText(Notes_List.get((int)position).get("Title").toString());
                    }
                }
                else {
                    if (Settings.getString("Text Size", "").equals("Large")) {
                        if
(Notes_List.get((int)position).get("Title").toString().length() > 22) {

title.setText(Notes_List.get((int)position).get("Title").toString().substri
ng((int)(0), (int)(22)).concat("..."));
                        }
                        else {

title.setText(Notes_List.get((int)position).get("Title").toString());
                        }
                    }
                    else {

                    }
                }
            }
        }
        if (Notes_List.get((int)position).containsKey("Time")) {
```

```
time.setText(Notes_List.get((int)position).get("Time").toString());
      }
      if (Notes_List.get((int)position).containsKey("Date")) {

date.setText(Notes_List.get((int)position).get("Date").toString());
      }
      if (Notes_List.get((int)position).containsKey("Method")) {
         if (Settings.getString("Text Size", "").equals("Small")) {
            if
(Notes_List.get((int)position).get("Method").toString().length() > 150) {

note.setText(Notes_List.get((int)position).get("Method").toString().substri
ng((int)(0), (int)(150)).concat("..."));
            }
            else {

note.setText(Notes_List.get((int)position).get("Method").toString());
            }
         }
         else {
            if (Settings.getString("Text Size", "").equals("Medium")) {
               if
(Notes_List.get((int)position).get("Method").toString().length() > 140) {

note.setText(Notes_List.get((int)position).get("Method").toString().substri
ng((int)(0), (int)(140)).concat("..."));
               }
               else {

note.setText(Notes_List.get((int)position).get("Method").toString());
               }
            }
            else {
               if (Settings.getString("Text Size", "").equals("Large")) {
                  if
(Notes_List.get((int)position).get("Method").toString().length() > 130) {

note.setText(Notes_List.get((int)position).get("Method").toString().substri
ng((int)(0), (int)(130)).concat("..."));
                  }
                  else {

note.setText(Notes_List.get((int)position).get("Method").toString());
                  }
               }
               else {

               }
            }
         }
      }
   }


   public void GetImportantNotes () {
      Important_Notes_List.clear();
      Position = 0;
      for(int repeat10 = 0; repeat10 < (int)(Notes_List.size());
repeat10++) {
         if
```

```
(Notes_List.get((int)Position).get("Important").toString().equals("True"))
{
            GetNotes = Notes_List.get((int)Position);
            Important_Notes_List.add(GetNotes);
         }
         Position++;
      }
   }


   @RequiresApi(api = Build.VERSION_CODES.N_MR1)
   public void _DynamicShortcutApp () {
      android.content.pm.ShortcutManager shortcutManager = null;
      if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.N_MR1) {
         shortcutManager =
getSystemService(android.content.pm.ShortcutManager.class);
      }
      if (shortcutManager != null) {
         android.content.pm.ShortcutInfo shortcut_1 = new
android.content.pm.ShortcutInfo.Builder(MealsActivity.this, "Activity1")
                        .setShortLabel("Add Meal")
                        .setLongLabel("Add New Meal")
                        .setRank(0)
                        .setIntent(new
android.content.Intent(android.content.Intent.ACTION_VIEW, null,
MealsActivity.this, AddMealActivity.class))

.setIcon(android.graphics.drawable.Icon.createWithResource(MealsActivity.th
is, R.drawable.ic_add_note))
                        .build();
         android.content.pm.ShortcutInfo shortcut_2= new
android.content.pm.ShortcutInfo.Builder(MealsActivity.this, "Activity2")
                        .setShortLabel("Settings")
                        .setLongLabel("Go To Settings")
                        .setRank(1)
                        .setIntent(new
android.content.Intent(android.content.Intent.ACTION_VIEW, null,
MealsActivity.this, ProfileActivity.class))

.setIcon(android.graphics.drawable.Icon.createWithResource(MealsActivity.th
is, R.drawable.ic_settings))
                        .build();

shortcutManager.setDynamicShortcuts(java.util.Arrays.asList(shortcut_1,shor
tcut_2));
      }
   }


   public void changeActivityFont (final String fontname) {
      fontName = "fonts/".concat(fontname.concat(".ttf"));
      overrideFonts(this,getWindow().getDecorView());
   }
   private void overrideFonts(final android.content.Context context, final
View v) {

      try {
         Typeface
         typeace = Typeface.createFromAsset(getAssets(), fontName);;
         if ((v instanceof ViewGroup)) {
            ViewGroup vg = (ViewGroup) v;
```

```
            for (int i = 0;
            i < vg.getChildCount();
            i++) {
               View child = vg.getChildAt(i);
               overrideFonts(context, child);
            }
         }
         else {
            if ((v instanceof TextView)) {
               ((TextView) v).setTypeface(typeace);
            }
            else {
               if ((v instanceof EditText )) {
                  ((EditText) v).setTypeface(typeace);
               }
               else {
                  if ((v instanceof Button)) {
                     ((Button) v).setTypeface(typeace);
                  }
               }
            }
         }
      }
      catch(Exception e)

      {

      };
   }


   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }

   @Deprecated
   public int getRandom(int min, int max) {
      Random random = new Random();
      return random.nextInt(max - min + 1) + min;
   }

   @Deprecated
   public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
      ArrayList<Double> result = new ArrayList<Double>();
```

```
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Meals List Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import android.graphics.drawable.GradientDrawable;
import android.os.Bundle;
import android.os.Parcelable;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.view.WindowManager;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
```

```
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import
com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import com.sdsmdg.tastytoast.TastyToast;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;


public class MealsListActivity extends AppCompatActivity {

   private FloatingActionButton fab;
   private double Position = 0;
   private boolean Selected = false;
   private double Selected_Length = 0;
   private boolean IsSelectAll = false;
   private String fontName = "";
   private String typeace = "";

   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> temp_maplist = new
ArrayList<>();

   private LinearLayout toolbar;
   private LinearLayout main;
   private ImageView back_img;
   private TextView toolbar_txt;
   private ImageView select_all_img;
   private ListView notes_list;

   private Intent To_Noteview = new Intent();
   private SharedPreferences Settings;
   private SharedPreferences AllMeals;
   private AlertDialog.Builder Delete_Dialog;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.meals_list);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      fab = (FloatingActionButton) findViewById(R.id.fab);

      toolbar =  findViewById(R.id.toolbar);
      main =  findViewById(R.id.main);
      back_img =  findViewById(R.id.back_img);
      toolbar_txt = (TextView) findViewById(R.id.toolbar_txt);
      select_all_img =  findViewById(R.id.select_all_img);
      notes_list = (ListView) findViewById(R.id.notes_list);
      Settings = getSharedPreferences("Settings", Activity.MODE_PRIVATE);
```

```
      AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);
      Delete_Dialog = new AlertDialog.Builder(this);

      back_img.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            finish();
         }
      });

      select_all_img.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            selectAll();
         }
      });

      notes_list.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
         @Override
         public void onItemClick(AdapterView<?> param1, View param2, int
param3, long param4) {
            final int position = param3;
            if (Selected) {
               if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Select").toString().equals("True")) {
                  Notes_List.get((int)(Notes_List.size() - 1) -
position).put("Select", "False");
                  Selected_Length--;
                  IsSelectAll = false;
               }
               else {
                  Notes_List.get((int)(Notes_List.size() - 1) -
position).put("Select", "True");
                  Selected_Length++;
               }
               refresh(Notes_List);
               _Toolbar(Selected_Length);
               if (Selected_Length == 0) {
                  IsSelectAll = false;
                  select_all_img.setEnabled(true);
                  select_all_img.setVisibility(View.INVISIBLE);
                  toolbar_txt.setText("Your Meals");
                  fab(false);
                  Selected = false;

((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
               }
            }
            else {
               To_Noteview.setAction(Intent.ACTION_VIEW);
               To_Noteview.setClass(getApplicationContext(),
ViewMealActivity.class);
               To_Noteview.putExtra("Position",
String.valueOf((long)((Notes_List.size() - 1) - position)));
               startActivity(To_Noteview);
               finish();
            }
         }
      });
```

```
      notes_list.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
         @Override
         public boolean onItemLongClick(AdapterView<?> param1, View param2,
int param3, long param4) {
            final int position = param3;
            if (Selected) {
               ToastUtil.showMessage(getApplicationContext(), "Selection
Mode Is Enabled");
            }
            else {
               Selected = true;
               Notes_List.get((int)(Notes_List.size() - 1) -
position).put("Select", "True");
               refresh(Notes_List);
               fab(true);
               Selected_Length++;
               _Toolbar(Selected_Length);
               select_all_img.setVisibility(View.VISIBLE);
               select_all_img.setEnabled(true);
            }
            return true;
         }
      });

      fab.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            Delete_Dialog.setTitle("Delete ?");
            Delete_Dialog.setMessage("Are You Sure To Delete
".concat(String.valueOf((long)(Selected_Length)).concat(" Selected Items ?
This Action Cannot Be Undone.")));
            Delete_Dialog.setPositiveButton("Delete", new
DialogInterface.OnClickListener() {
               @Override
               public void onClick(DialogInterface dialog, int which) {
                  Selected = false;
                  Position = Notes_List.size() - 1;
                  while(true) {
                     if (Position == -1) {
                        break;
                     }
                     else {
                        if
(Notes_List.get((int)Position).get("Select").toString().equals("True")) {
                           Notes_List.remove((int)(Position));
                        }
                        Position--;
                     }
                  }
                  refresh(Notes_List);
                  Selected_Length = 0;
                  toolbar_txt.setText("Your Meals");
                  fab(false);
                  select_all_img.setVisibility(View.INVISIBLE);
                  saveData();
                  TastyToast.makeText(getApplicationContext(), "Your Meals
Have Been Successfuly Deleted ", TastyToast.LENGTH_LONG,
TastyToast.SUCCESS);
               }
            });
```

```
            Delete_Dialog.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    _UnSelectAll();
                }
            });
            Delete_Dialog.create().show();
        }
      });
   }

   private void initializeLogic() {
      changeActivityFont("railway_semisolid");
      removeScrollBar(notes_list);
      getSettingsData();
      fab(false);
      select_all_img.setVisibility(View.INVISIBLE);
      select_all_img.setEnabled(false);
      if (!getIntent().getStringExtra("Data").equals("")) {
         Notes_List = new
Gson().fromJson(getIntent().getStringExtra("Data"), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
         notes_list.setAdapter(new Notes_listAdapter(Notes_List));
         ((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
      }
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   @Override
   public void onBackPressed() {
      if (Selected) {
         _UnSelectAll();
      }
      else {
         finish();
      }
   }
   public void setRadiusToView (final View view, final double radius, final
String Colour) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
gd.setColor(Color.parseColor(Colour)); gd.setCornerRadius((int)radius);
view.setBackground(gd);
   }


   public void setRadiusToView (final View view, final double value) {
      view.setElevation((float)value);
   }
```

```
   public void corners (final View view, final String _color1, final String
_color2, final double msgtr, final double _n1, final double _n2, final
double _n3, final double _n4) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();

      gd.setColor(Color.parseColor(_color1));

      gd.setStroke((int)msgtr, Color.parseColor(_color2));

      gd.setCornerRadii(new
float[]{(int)_n1,(int)_n1,(int)_n2,(int)_n2,(int)_n3,(int)_n3,(int)_n4,(int
)_n4});

      view.setBackground(gd);

      view.setElevation(4);
   }


   public void Add (final String Colour, final ImageView Imageview) {
      Imageview.getDrawable().setColorFilter(Color.parseColor(Colour),
PorterDuff.Mode.SRC_IN);
   }


   public void removeScrollBar (final View view) {
      view.setVerticalScrollBarEnabled(false);
view.setHorizontalScrollBarEnabled(false);
   }


   public void getSettingsData () {
      if (Settings.getString("Text Size", "").equals("Small")) {

      }
      else {
         if (Settings.getString("Text Size", "").equals("Small")) {
            textSize(toolbar_txt, 20);
         }
         else {
            if (Settings.getString("Text Size", "").equals("Large")) {
               textSize(toolbar_txt, 22);
            }
            else {

            }
         }
      }
      if (Settings.getString("Theme", "").equals("Default")) {
         Window w =
MealsListActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.
FLAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_S
YSTEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#FFFFFF"));
main.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
         Add("#3770FD", back_img);
         Add("#3770FD", select_all_img);
      }
      else {
         if (Settings.getString("Theme", "").equals("Dark")) {
```

```
            Window w =
MealsListActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.
FLAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_S
YSTEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#263238"));
            toolbar.setBackgroundColor(0xFF263238);
            toolbar_txt.setTextColor(0xFFFFFFFF);
            main.setBackgroundColor(0xFF263238);

fab.setBackgroundTintList(android.content.res.ColorStateList.valueOf(Color.
parseColor("#263238")));
            Add("#FFFFFF", back_img);
            Add("#FFFFFF", select_all_img);
         }
         else {
            if (Settings.getString("Theme", "").equals("Blue Grey")) {
               Window w =
MealsListActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.
FLAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_S
YSTEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#FFFFFF"));

main.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
               toolbar.setBackgroundColor(0xFFFFFFFF);
               toolbar_txt.setTextColor(0xFF607D8B);
               main.setBackgroundColor(0xFFFFFFFF);

fab.setBackgroundTintList(android.content.res.ColorStateList.valueOf(Color.
parseColor("#607D8B")));
               Add("#607D8B", back_img);
               Add("#607D8B", select_all_img);
            }
            else {
               if (Settings.getString("Theme", "").equals("Orange")) {
                  Window w =
MealsListActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.
FLAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_S
YSTEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#FFFFFF"));

main.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
                  toolbar.setBackgroundColor(0xFFFFFFFF);
                  toolbar_txt.setTextColor(0xFFFF5722);
                  main.setBackgroundColor(0xFFFFFFFF);

fab.setBackgroundTintList(android.content.res.ColorStateList.valueOf(Color.
parseColor("#FF5722")));
                  Add("#FF5722", back_img);
                  Add("#FF5722", select_all_img);
               }
               else {
                  if (Settings.getString("Theme", "").equals("Indigo")) {
                     Window w =
MealsListActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.
FLAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_S
YSTEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#FFFFFF"));

main.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
                     toolbar.setBackgroundColor(0xFFFFFFFF);
                     toolbar_txt.setTextColor(0xFF3F51B5);
                     main.setBackgroundColor(0xFFFFFFFF);

fab.setBackgroundTintList(android.content.res.ColorStateList.valueOf(Color.
parseColor("#3F51B5")));
```

```
                    Add("#3F51B5", back_img);
                    Add("#3F51B5", select_all_img);
                }
                else {

                }
            }
        }
    }
}


public void textSize (final TextView TextView1, final double size) {
    int j = (int) size;
    TextView1.setTextSize(j);
}


public void fab (final boolean visibility) {
    if (visibility) {
        fab.show();
    }
    else {
        fab.hide();
    }
}


public void _Toolbar (final double _Length) {

toolbar_txt.setText(String.valueOf((long)(_Length)).concat("/".concat(Strin
g.valueOf((long)(Notes_List.size())).concat(" Selected"))));
}


public void _UnSelectAll () {
    Selected = false;
    Selected_Length = 0;
    toolbar_txt.setText("Your Meals");
    select_all_img.setVisibility(View.INVISIBLE);
    select_all_img.setEnabled(false);
    IsSelectAll = false;
    Position = 0;
    for(int i = 0; i < (int)(Notes_List.size()); i++) {
        if
(Notes_List.get((int)Position).get("Select").toString().equals("True")) {
            Notes_List.get((int)Position).put("Select", "False");
        }
        Position++;
    }
    refresh(Notes_List);
    fab(false);
}


public void refresh (final ArrayList<HashMap<String, Object>> ListMap) {
    Parcelable state =
    notes_list.onSaveInstanceState();
    notes_list.setAdapter(new Notes_listAdapter(ListMap));
    ((BaseAdapter)notes_list.getAdapter()).notifyDataSetChanged();
```

```
      notes_list.onRestoreInstanceState(state);
   }


   public void selectAll () {
      if (IsSelectAll) {
         _UnSelectAll();
         IsSelectAll = false;
      }
      else {
         IsSelectAll = true;
         Position = 0;
         for(int i = 0; i < (int)(Notes_List.size()); i++) {
            if
(Notes_List.get((int)Position).get("Select").toString().equals("False")) {
               Notes_List.get((int)Position).put("Select", "True");
               Selected_Length++;
            }
            Position++;
            _Toolbar(Selected_Length);
         }
         refresh(Notes_List);
      }
   }


   public void saveData () {
      temp_maplist = new Gson().fromJson(AllMeals.getString("Meals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      if (Notes_List.size() == 0) {
         if (getIntent().getStringExtra("Type").equals("All")) {
            AllMeals.edit().putString("Meals", new
Gson().toJson(Notes_List)).commit();
         }
         else {
            if (getIntent().getStringExtra("Type").equals("Important")) {
               Position = temp_maplist.size() - 1;
               for(int i = 0; i < (int)(temp_maplist.size()); i++) {
                  if
(temp_maplist.get((int)Position).get("Important").toString().equals("True")
) {
                     temp_maplist.remove((int)(Position));
                  }
                  Position--;
               }
               AllMeals.edit().putString("Meals", new
Gson().toJson(temp_maplist)).commit();
            }
         }
      }
      else {
         AllMeals.edit().putString("Meals", new
Gson().toJson(Notes_List)).commit();
      }
   }


   public void changeActivityFont (final String fontname) {
      fontName = "fonts/".concat(fontname.concat(".ttf"));
      overrideFonts(this,getWindow().getDecorView());
   }
```

```
   private void overrideFonts(final android.content.Context context, final
View v) {

      try {
         Typeface
         typeace = Typeface.createFromAsset(getAssets(), fontName);;
         if ((v instanceof ViewGroup)) {
            ViewGroup vg = (ViewGroup) v;
            for (int i = 0;
            i < vg.getChildCount();
            i++) {
               View child = vg.getChildAt(i);
               overrideFonts(context, child);
            }
         }
         else {
            if ((v instanceof TextView)) {
               ((TextView) v).setTypeface(typeace);
            }
            else {
               if ((v instanceof EditText )) {
                  ((EditText) v).setTypeface(typeace);
               }
               else {
                  if ((v instanceof Button)) {
                     ((Button) v).setTypeface(typeace);
                  }
               }
            }
         }
      }
      catch(Exception e)

      {

      };
   }


   public class Notes_listAdapter extends BaseAdapter {
      ArrayList<HashMap<String, Object>> data;
      public Notes_listAdapter(ArrayList<HashMap<String, Object>> arr) {
         data = arr;
      }

      @Override
      public int getCount() {
         return data.size();
      }

      @Override
      public HashMap<String, Object> getItem(int index) {
         return data.get(index);
      }

      @Override
      public long getItemId(int index) {
         return index;
      }
      @Override
      public View getView(final int position, View v, ViewGroup container)
```

```
{
        LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
        View view = v;
        if (view == null) {
           view = inflater.inflate(R.layout.custom_meal_list, null);
        }

        final LinearLayout bg =  view.findViewById(R.id.bg);
        final LinearLayout linear1 =  view.findViewById(R.id.linear1);
        final ImageView imageview1 =  view.findViewById(R.id.imageview1);
        final LinearLayout main =  view.findViewById(R.id.main);
        final LinearLayout side_lin =  view.findViewById(R.id.side_lin);
        final LinearLayout text_lin =  view.findViewById(R.id.text_lin);
        final LinearLayout title_time_lin =
view.findViewById(R.id.title_time_lin);
        final LinearLayout note_lin =  view.findViewById(R.id.note_lin);
        final TextView date = (TextView) view.findViewById(R.id.date);
        final LinearLayout linear2 =  view.findViewById(R.id.linear2);
        final TextView time = (TextView) view.findViewById(R.id.time);
        final TextView title = (TextView) view.findViewById(R.id.title);
        final TextView txtFoodClasses = (TextView)
view.findViewById(R.id.txtFoodClasses);
        final TextView note = (TextView) view.findViewById(R.id.note);
        final TextView txt_methods = (TextView)
view.findViewById(R.id.txt_methods);
        final TextView txt_calories = (TextView)
view.findViewById(R.id.txt_calories);
        final TextView txtminutes = (TextView)
view.findViewById(R.id.txtminutes);
        final TextView textview4 = (TextView)
view.findViewById(R.id.textview4);
        final TextView txtServing = (TextView)
view.findViewById(R.id.txtServing);

        linear1.setVisibility(View.GONE);

date.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
ium.ttf"), Typeface.NORMAL);

title.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansbo
ld.ttf"), Typeface.NORMAL);

time.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
ium.ttf"), Typeface.NORMAL);

note.setTypeface(Typeface.createFromAsset(getAssets(),"fonts/productsansmed
ium.ttf"), Typeface.NORMAL);
        side_lin.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)60, 0xFF3F51B6));
        if (Selected) {

imageview1.setImageResource(R.drawable.ic_radio_button_off_black);
            if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Select").toString().equals("True")) {
                imageview1.setVisibility(View.VISIBLE);

imageview1.setImageResource(R.drawable.ic_radio_button_on_black);
            }
```

```
            else {

imageview1.setImageResource(R.drawable.ic_radio_button_off_black);
            }
         }
         else {
            imageview1.setVisibility(View.GONE);
         }
         if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Title")) {
            if (Settings.getString("Text Size", "").equals("Small")) {
               title.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString());
               if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Title").toString().length() > 28) {
                  title.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Title").toString().substring((int)(0),
(int)(28)).concat("..."));
               }
               else {

               }
            }
            else {

            }
         }
         if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Time")) {
            time.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Time").toString());
         }
         if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Date")) {
            date.setText(Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Date").toString());
         }
         if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Method")) {
            if (Settings.getString("Text Size", "").equals("Small")) {
               if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Method").toString().length() > 150) {
                  note.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Method").toString().substring((int)(0),
(int)(150)).concat("..."));
               }
               else {
                  note.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Method").toString());
               }
            }
            else {
               if (Settings.getString("Text Size", "").equals("Medium")) {
                  if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Method").toString().length() > 140) {
                     note.setText(Notes_List.get((int)(Notes_List.size() -
1) - position).get("Method").toString().substring((int)(0),
(int)(140)).concat("..."));
                  }
                  else {
                     note.setText(Notes_List.get((int)(Notes_List.size() -
```

```
1) - position).get("Method").toString());
                    }
                }
                else {
                    if (Settings.getString("Text Size", "").equals("Large"))
{
                        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).get("Method").toString().length() > 130) {
                            note.setText(Notes_List.get((int)(Notes_List.size()
- 1) - position).get("Method").toString().substring((int)(0),
(int)(130)).concat("..."));
                        }
                        else {
                            note.setText(Notes_List.get((int)(Notes_List.size()
- 1) - position).get("Method").toString());
                        }
                    }
                    else {

                    }
                }
            }
        }
        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Note")) {
            txt_methods.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Note").toString());
        }
        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Calories")) {
            txt_calories.setText(Notes_List.get((int)(Notes_List.size() -
1) - position).get("Calories").toString());
        }
        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Minutes")) {
            txtminutes.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Minutes").toString());
        }
        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Ingredients")) {
            txt_methods.setText(Notes_List.get((int)(Notes_List.size() - 1)
- position).get("Ingredients").toString());
        }
        if (Notes_List.get((int)(Notes_List.size() - 1) -
position).containsKey("Class")) {
            txtFoodClasses.setText(Notes_List.get((int)(Notes_List.size() -
1) - position).get("Class").toString());
        }

        return view;
    }
  }

  @Deprecated
  public void showMessage(String msg) {
     Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
  }

  @Deprecated
  public int getLocationX(View v) {
```

```
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Planned Meals View Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.animation.ObjectAnimator;
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
```

```
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import android.graphics.drawable.GradientDrawable;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.Animation;
import android.view.animation.ScaleAnimation;
import android.widget.Button;
import android.widget.EditText;
import android.widget.FrameLayout;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.airbnb.lottie.LottieAnimationView;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;


public class PlannedMealsViewActivity extends AppCompatActivity {

   private String Title = "";
   private String Note = "";
   private HashMap<String, Object> Get_Note = new HashMap<>();
   private double Position = 0;
   private String fontName = "";
   private String typeace = "";
   private String Method = "";
   private String Ingredients = "";
   private String Calories = "";
   private String Hours = "";
   private String Minutes = "";
   private String classes = "";
   private HashMap<String, Object> Get_Meal = new HashMap<>();
   private String Breakfastclass = "";
   private String Breakfastkcal = "";
   private String Lunch = "";
   private String Lunchclass = "";
   private String Lunchkcal = "";
   private String Snack = "";
   private String Snackclass = "";
   private String Snackkcal = "";
   private String Dinner = "";
   private String Dinnerclass = "";
   private String Dinnerkcal = "";
   private String Date = "";
   private String Time = "";
```

```
   private String Serving = "";
   private String Breakfast = "";
   private String Breakfastserv = "";
   private String Lunchserv = "";
   private String Snackserv = "";
   private String Dinnerserv = "";

   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> meal_list = new
ArrayList<>();

   private LinearLayout toolbar;
   private LinearLayout linear17;
   private LinearLayout linear24;
   private LinearLayout linear27;
   private LinearLayout linear25;
   private ImageView back;
   private ImageView edit_img;
   private LinearLayout linear18;
   private LinearLayout linear19;
   private ScrollView vscroll1;
   private LinearLayout linear23;
   private TextView txtMealToolbar;
   private LinearLayout linrMealBoard;
   private LinearLayout linear14;
   private LinearLayout linear31;
   private LinearLayout linear28;
   private LinearLayout linear26;
   private LinearLayout linear30;
   private FrameLayout linear32;
   private TextView textview13;
   private ProgressBar progressbar1;
   private TextView txtCalorieFigure;
   private LottieAnimationView lottie1;
   private TextView txtMeal;
   private TextView txtFoodClass;
   private FrameLayout linear33;
   private TextView textview15;
   private ProgressBar progressbar2;
   private TextView txtServing;
   private LinearLayout linrDate;
   private LinearLayout linrUpdated;
   private ImageView imageview6;
   private TextView txtDateCreated;
   private TextView txtDate;
   private ImageView imageview7;
   private TextView txtLastTimeFigure;
   private TextView txtUpdated;
   private LinearLayout linr_breakfast;
   private LinearLayout linr_lunch;
   private LinearLayout linrsnack;
   private LinearLayout linr_dinner;
   private TextView textview20;
   private TextView textview21;
   private TextView textview22;
   private TextView textview23;

   private TextToSpeech NoteSpeak;
   private SharedPreferences Settings;
   private SharedPreferences AllMeals;
```

```
private Intent Note_Edit = new Intent();
private SharedPreferences meals;
private Intent meal_edit = new Intent();
private ObjectAnimator anime = new ObjectAnimator();

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.planned_meals_view);
    initialize(savedInstanceState);
    com.google.firebase.FirebaseApp.initializeApp(this);
    initializeLogic();
}

private void initialize(Bundle savedInstanceState) {
    toolbar =  findViewById(R.id.toolbar);
    linear17 =  findViewById(R.id.linear17);
    linear24 =  findViewById(R.id.linear24);
    linear27 =  findViewById(R.id.linear27);
    linear25 =  findViewById(R.id.linear25);
    back =  findViewById(R.id.back);
    edit_img =  findViewById(R.id.edit_img);
    linear18 =  findViewById(R.id.linear18);
    linear19 =  findViewById(R.id.linear19);
    vscroll1 =  findViewById(R.id.vscroll1);
    linear23 =  findViewById(R.id.linear23);
    txtMealToolbar = (TextView) findViewById(R.id.txtMealToolbar);
    linrMealBoard =  findViewById(R.id.linrMealBoard);
    linear14 =  findViewById(R.id.linear14);
    linear31 =  findViewById(R.id.linear31);
    linear28 =  findViewById(R.id.linear28);
    linear26 =  findViewById(R.id.linear26);
    linear30 =  findViewById(R.id.linear30);
    linear32 = (FrameLayout) findViewById(R.id.linear32);
    textview13 = (TextView) findViewById(R.id.textview13);
    progressbar1 = (ProgressBar) findViewById(R.id.progressbar1);
    txtCalorieFigure = (TextView) findViewById(R.id.txtCalorieFigure);
    lottie1 = (LottieAnimationView) findViewById(R.id.lottie1);
    txtMeal = (TextView) findViewById(R.id.txtMeal);
    txtFoodClass = (TextView) findViewById(R.id.txtFoodClass);
    linear33 = (FrameLayout) findViewById(R.id.linear33);
    textview15 = (TextView) findViewById(R.id.textview15);
    progressbar2 = (ProgressBar) findViewById(R.id.progressbar2);
    txtServing = (TextView) findViewById(R.id.txtServing);
    linrDate =  findViewById(R.id.linrDate);
    linrUpdated =  findViewById(R.id.linrUpdated);
    imageview6 =  findViewById(R.id.imageview6);
    txtDateCreated = (TextView) findViewById(R.id.txtDateCreated);
    txtDate = (TextView) findViewById(R.id.txtDate);
    imageview7 =  findViewById(R.id.imageview7);
    txtLastTimeFigure = (TextView) findViewById(R.id.txtLastTimeFigure);
    txtUpdated = (TextView) findViewById(R.id.txtUpdated);
    linr_breakfast =  findViewById(R.id.linr_breakfast);
    linr_lunch =  findViewById(R.id.linr_lunch);
    linrsnack =  findViewById(R.id.linrsnack);
    linr_dinner =  findViewById(R.id.linr_dinner);
    textview20 = (TextView) findViewById(R.id.textview20);
    textview21 = (TextView) findViewById(R.id.textview21);
    textview22 = (TextView) findViewById(R.id.textview22);
    textview23 = (TextView) findViewById(R.id.textview23);
    NoteSpeak = new TextToSpeech(getApplicationContext(), null);
```

```
        Settings = getSharedPreferences("Settings", Activity.MODE_PRIVATE);
        AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);
        meals = getSharedPreferences("meals", Activity.MODE_PRIVATE);

        linear24.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                clickAnimation(linear24);
                finish();
            }
        });

        linear25.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                meal_edit.setAction(Intent.ACTION_VIEW);
                meal_edit.setClass(getApplicationContext(),
MealEditActivity.class);
                meal_edit.putExtra("Position",
getIntent().getStringExtra("Position"));
                startActivity(meal_edit);
            }
        });

        back.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                finish();
            }
        });

        edit_img.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                meal_edit.setAction(Intent.ACTION_VIEW);
                meal_edit.setClass(getApplicationContext(),
MealEditActivity.class);
                meal_edit.putExtra("Position",
getIntent().getStringExtra("Position"));
                startActivity(meal_edit);
                finish();
            }
        });

        linr_breakfast.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                txtMealToolbar.setText("Breakfast");
                linr_breakfast.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)50, 0xFF3F51B6));
                linr_lunch.setBackgroundColor(0xFFFFFFFF);
                linrsnack.setBackgroundColor(0xFFFFFFFF);
                linr_dinner.setBackgroundColor(0xFFFFFFFF);
                clickAnimation(linr_breakfast);
                clickAnimation(linrMealBoard);
                try{
                    txtMeal.setText(Breakfast);
                    txtFoodClass.setText(Breakfastclass);
                    txtCalorieFigure.setText(Breakfastkcal.concat("/1000"));
                    txtServing.setText(Breakfastserv.concat("/5"));
```

```
progressbar1.setProgress((int)Double.parseDouble(Breakfastkcal));

progressbar2.setProgress((int)Double.parseDouble(Breakfastserv));
            }catch(Exception e){

            }
         }
      });

      linr_lunch.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            linr_lunch.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)50, 0xFF3F51B6));
            linr_breakfast.setBackgroundColor(0xFFFFFFFF);
            linrsnack.setBackgroundColor(0xFFFFFFFF);
            linr_dinner.setBackgroundColor(0xFFFFFFFF);
            clickAnimation(linr_lunch);
            clickAnimation(linrMealBoard);
            try{
               txtMealToolbar.setText("Lunch");
               txtMeal.setText(Lunch);
               txtCalorieFigure.setText(Lunchkcal.concat("/1000"));
               txtServing.setText(Lunchserv.concat("/5"));
               txtFoodClass.setText(Lunchclass);

progressbar1.setProgress((int)Double.parseDouble(Lunchkcal));

progressbar2.setProgress((int)Double.parseDouble(Lunchserv));
            }catch(Exception e){

            }
         }
      });

      linrsnack.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            linrsnack.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)50, 0xFF3F51B6));
            linr_lunch.setBackgroundColor(0xFFFFFFFF);
            linr_breakfast.setBackgroundColor(0xFFFFFFFF);
            linr_dinner.setBackgroundColor(0xFFFFFFFF);
            clickAnimation(linrsnack);
            clickAnimation(linrMealBoard);
            try{
               txtMeal.setText(Snack);
               txtCalorieFigure.setText(Snackkcal.concat("/1000"));
               txtServing.setText(Snackserv.concat("/5"));
               txtFoodClass.setText(Snackclass);

progressbar1.setProgress((int)Double.parseDouble(Snackkcal));

progressbar2.setProgress((int)Double.parseDouble(Snackserv));
               txtMealToolbar.setText("Snack");
            }catch(Exception e){

            }
```

```
         }
      });

      linr_dinner.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            linr_dinner.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)50, 0xFF3F51B6));
            linr_lunch.setBackgroundColor(0xFFFFFFFF);
            linrsnack.setBackgroundColor(0xFFFFFFFF);
            linr_breakfast.setBackgroundColor(0xFFFFFFFF);
            clickAnimation(linr_dinner);
            clickAnimation(linrMealBoard);
            try{
               txtMeal.setText(Dinner);
               txtCalorieFigure.setText(Dinnerkcal.concat("/1000"));
               txtServing.setText(Dinnerserv.concat("/5"));
               txtFoodClass.setText(Dinnerclass);

progressbar1.setProgress((int)Double.parseDouble(Dinnerkcal));

progressbar2.setProgress((int)Double.parseDouble(Dinnerserv));
               txtMealToolbar.setText("Dinner");
            }catch(Exception e){

            }
         }
      });
   }

   private void initializeLogic() {
      progressbar1.setProgressDrawable(getDrawable(R.drawable.progress));
      progressbar2.setProgressDrawable(getDrawable(R.drawable.progress));

      changeActivityFont("railway_semisolid");
      ONCREATE();
      linr_breakfast.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)50, 0xFF3F51B6));
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   @Override
   public void onStart() {
      super.onStart();
      _GetNote();
   }
   public void Typeface () {

   }
```

```
   public void ONCREATE (){
      Position =
Double.parseDouble(getIntent().getStringExtra("Position"));
      Typeface();
      _GetNote();
      try{
         txtMeal.setText(Breakfast);
         txtFoodClass.setText(Breakfastclass);
         txtCalorieFigure.setText(Breakfastkcal.concat("/1000"));
         txtServing.setText(Breakfastserv.concat("/5"));
         progressbar1.setProgress((int)Double.parseDouble(Breakfastkcal));
         progressbar2.setProgress((int)Double.parseDouble(Breakfastserv));
         txtDateCreated.setText(Date);
         txtLastTimeFigure.setText(Time);
      }catch(Exception e){
         ToastUtil.showMessage(getApplicationContext(), "Data not fetched
correctly");
      }
   }


   public void Add (final String Colour, final ImageView Imageview) {
      Imageview.getDrawable().setColorFilter(Color.parseColor(Colour),
PorterDuff.Mode.SRC_IN);
   }


   public void getSettingsData (){

   }


   public void _detectLinks (final TextView _txt_linkify) {
      _txt_linkify.setClickable(true);
      android.text.util.Linkify.addLinks(_txt_linkify,
android.text.util.Linkify.ALL);
      _txt_linkify.setLinkTextColor(Color.parseColor("#FF3770FD"));
      _txt_linkify.setLinksClickable(true);
   }


   public void textSize (final TextView TextView1, final double size) {
      int j = (int) size;
      TextView1.setTextSize(j);
   }


   public void _GetNote (){
      if (!meals.getString("Meals", "").equals("")) {
         meal_list = new Gson().fromJson(meals.getString("Meals", ""), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
         Get_Meal = meal_list.get((int)Position);
         Breakfast = Get_Meal.get("Breakfast").toString();
         Breakfastclass = Get_Meal.get("Breakfastclass").toString();
         Breakfastkcal = Get_Meal.get("Breakfastkcal").toString();
         Breakfastserv = Get_Meal.get("Breakfastserv").toString();
         Lunch = Get_Meal.get("Lunch").toString();
         Lunchclass = Get_Meal.get("Lunchclass").toString();
         Lunchkcal = Get_Meal.get("Lunchkkcal").toString();
```

```
         Lunchserv = Get_Meal.get("Lunchserv").toString();
         Snack = Get_Meal.get("Snack").toString();
         Snackclass = Get_Meal.get("Snackclass").toString();
         Snackkcal = Get_Meal.get("Snackkcal").toString();
         Snackserv = Get_Meal.get("Snackserv").toString();
         Dinner = Get_Meal.get("Dinner").toString();
         Dinnerclass = Get_Meal.get("Dinnerclass").toString();
         Dinnerkcal = Get_Meal.get("Dinnerkcal").toString();
         Dinnerserv = Get_Meal.get("Dinnerserv").toString();
         Date = Get_Meal.get("Date").toString();
         Time = Get_Meal.get("Time").toString();
      }
   }


   public void changeActivityFont (final String fontname) {
      fontName = "fonts/".concat(fontname.concat(".ttf"));
      overrideFonts(this,getWindow().getDecorView());
   }
   private void overrideFonts(final android.content.Context context, final
View v) {

      try {
         Typeface
         typeace = Typeface.createFromAsset(getAssets(), fontName);;
         if ((v instanceof ViewGroup)) {
            ViewGroup vg = (ViewGroup) v;
            for (int i = 0;
            i < vg.getChildCount();
            i++) {
               View child = vg.getChildAt(i);
               overrideFonts(context, child);
            }
         }
         else {
            if ((v instanceof TextView)) {
               ((TextView) v).setTypeface(typeace);
            }
            else {
               if ((v instanceof EditText )) {
                  ((EditText) v).setTypeface(typeace);
               }
               else {
                  if ((v instanceof Button)) {
                     ((Button) v).setTypeface(typeace);
                  }
               }
            }
         }
      }
      catch(Exception e)

      {

      };
   }


   public void clickAnimation (final View view) {
      ScaleAnimation fade_in = new ScaleAnimation(0.9f, 1f, 0.9f, 1f,
Animation.RELATIVE_TO_SELF, 0.5f, Animation.RELATIVE_TO_SELF, 0.7f);
```

```
        fade_in.setDuration(300);
        fade_in.setFillAfter(true);
        view.startAnimation(fade_in);
        //aauraparti YouTube channel//
    }


    @Deprecated
    public void showMessage(String msg) {
        Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
    }

    @Deprecated
    public int getLocationX(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Privacy Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.content.Intent;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;
import java.util.Random;


public class PrivacyActivity extends AppCompatActivity {

   private LinearLayout linear2;
   private ScrollView vscroll1;
   private ImageView imageview1;
   private TextView textview1;
   private LinearLayout linear3;
   private TextView textview2;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.privacy);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      linear2 =  findViewById(R.id.linear2);
      vscroll1 =  findViewById(R.id.vscroll1);
      imageview1 =  findViewById(R.id.imageview1);
      textview1 = (TextView) findViewById(R.id.textview1);
      linear3 =  findViewById(R.id.linear3);
      textview2 = (TextView) findViewById(R.id.textview2);
   }

   private void initializeLogic() {
      textview2.setText("DietTracker built the Diet Tracker app as a Free
app. This SERVICE is provided by DietTracker at no cost and is intended for
use as is.\n\nThis page is used to inform visitors regarding my policies
with the collection, use, and disclosure of Personal Information if anyone
decided to use my Service.\n\nIf you choose to use my Service, then you
```

```
agree to the collection and use of information in relation to this policy. The Personal Information that I collect is used for providing and improving the Service. I will not use or share your information with anyone except as described in this Privacy Policy.\n\nThe terms used in this Privacy Policy have the same meanings as in our Terms and Conditions, which is accessible at Diet Tracker unless otherwise defined in this Privacy Policy.\n\nInformation Collection and Use:\n\nFor a better experience, while using our Service, I may require you to provide us with certain personally identifiable information, including but not limited to email, age, height, weight. The information that I request will be retained on your device and is not collected by me in any way.\n\nThe app does use third party services that may collect information used to identify you.\n\nReference to third party service providers used by the app are below;\n\nFirebase.    Google Analytics.              \n\nLog Data:\n\nI want to inform you that whenever you use my Service, in a case of an error in the app I collect data and information (through third party products) on your phone called Log Data. This Log Data may include information such as your device Internet Protocol (“IP”) address, device name, operating system version, the configuration of the app when utilizing my Service, the time and date of your use of the Service, and other statistics.\n\nCookies:\n\nCookies are files with a small amount of data that are commonly used as anonymous unique identifiers. These are sent to your browser from the websites that you visit and are stored on your device's internal memory.\n\nThis Service does not use these “cookies” explicitly. However, the app may use third party code and libraries that use “cookies” to collect information and improve their services. You have the option to either accept or refuse these cookies and know when a cookie is being sent to your device. If you choose to refuse our cookies, you may not be able to use some portions of this Service.\n\nService Providers:\n\nI may employ third-party companies and individuals due to the following reasons:\n\nTo facilitate our Service;\nTo provide the Service on our behalf;\nTo perform Service-related services; or\nTo assist us in analyzing how our Service is used.\n\nI want to inform users of this Service that these third parties have access to your Personal Information. The reason is to perform the tasks assigned to them on our behalf. However, they are obligated not to disclose or use the information for any other purpose.\n\nSecurity:\n\nI value your trust in providing us your Personal Information, thus we are striving to use commercially acceptable means of protecting it. But remember that no method of transmission over the internet, or method of electronic storage is 100% secure and reliable, and I cannot guarantee its absolute security.\n\nLinks to Other Sites:\n\nThis Service may contain links to other sites. If you click on a third-party link, you will be directed to that site. Note that these external sites are not operated by me. Therefore, I strongly advise you to review the Privacy Policy of these websites. I have no control over and assume no responsibility for the content, privacy policies, or practices of any third-party sites or services.\n\nChildren’s Privacy:\n\nThese Services do not address anyone under the age of 13. I do not knowingly collect personally identifiable information from children under 13. In a case I discover that a child under 13 has provided me with personal information, I immediately delete this from our servers. If you are a parent or guardian and you are aware that your child has provided us with personal information, please contact me so that I will be able to take necessary actions.\n\nChanges to This Privacy Policy:\n\nI may update our Privacy Policy from time to time. Thus, you are advised to review this page periodically for any changes. I will notify you of any changes by posting the new Privacy Policy on this page.\n\nThis policy is effective as of 13-03-2022\n\nContact Us:\n\nIf you have any questions or suggestions about my Privacy Policy, do not hesitate to contact me at ayelabolaj@gmail.com");
    }
```

```
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);
        switch (requestCode) {

            default:
            break;
        }
    }

    @Deprecated
    public void showMessage(String msg) {
        Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
    }

    @Deprecated
    public int getLocationX(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
```

```
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Profile Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.Typeface;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.view.WindowManager;
import android.view.animation.Animation;
import android.view.animation.ScaleAnimation;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.blogspot.atifsoftwares.animatoolib.Animatoo;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

import java.util.ArrayList;
import java.util.Random;


public class ProfileActivity extends AppCompatActivity {

    private String fontName = "";
    private String typeace = "";
```

```
private LinearLayout bg_hevo;
private LinearLayout b_hevo_1;
private LinearLayout linear3;
private LinearLayout bg_hevo_2;
private ImageView image_hevo1;
private LinearLayout b_hevo_2;
private ImageView image_hevo2;
private CheckBox check_hevo;
private LinearLayout linear4;
private ImageView img_edit_profile;
private ImageView image_hevo4;
private TextView txt_username;
private ScrollView vscroll1;
private LinearLayout linear5;
private LinearLayout btn_privacy;
private LinearLayout bt_hevo_2;
private LinearLayout bt_hevo_3;
private LinearLayout bt_hevo_4;
private LinearLayout bt_hevo_5;
private LinearLayout bt_hevo_6;
private ImageView image_hevo5;
private TextView text_hevo3;
private ImageView image_hevo_bt1;
private ImageView image_hevo6;
private TextView text_hevo4;
private ImageView image_hevo_bt2;
private ImageView image_hevo7;
private TextView text_hevo5;
private ImageView image_hevo_bt3;
private ImageView image_hevo8;
private TextView text_hevo6;
private ImageView image_hevo_bt4;
private ImageView image_hevo9;
private TextView text_hevo7;
private ImageView image_hevo_bt5;
private ImageView image_hevo10;
private TextView text_hevo8;
private ImageView image_hevo_bt6;

private SharedPreferences hevo;
private Intent privacy = new Intent();
private Intent weightlog = new Intent();
private Intent support = new Intent();
private AlertDialog.Builder logout;
private FirebaseAuth auth;
private OnCompleteListener<Void> auth_updateEmailListener;
private OnCompleteListener<Void> auth_updatePasswordListener;
private OnCompleteListener<Void> auth_emailVerificationSentListener;
private OnCompleteListener<Void> auth_deleteUserListener;
private OnCompleteListener<Void> auth_updateProfileListener;
private OnCompleteListener<AuthResult> auth_phoneAuthListener;
private OnCompleteListener<AuthResult> auth_googleSignInListener;
private OnCompleteListener<AuthResult> auth_create_user_listener;
private OnCompleteListener<AuthResult> authsign_in_listener;
private OnCompleteListener<Void> _auth_reset_password_listener;
private Intent login = new Intent();
private SharedPreferences name;
private Intent edit = new Intent();

@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.profile);
    initialize(savedInstanceState);
    com.google.firebase.FirebaseApp.initializeApp(this);
    initializeLogic();
}

private void initialize(Bundle savedInstanceState) {
    bg_hevo =  findViewById(R.id.bg_hevo);
    b_hevo_1 =  findViewById(R.id.b_hevo_1);
    linear3 =  findViewById(R.id.linear3);
    bg_hevo_2 =  findViewById(R.id.bg_hevo_2);
    image_hevo1 =  findViewById(R.id.image_hevo1);
    b_hevo_2 =  findViewById(R.id.b_hevo_2);
    image_hevo2 =  findViewById(R.id.image_hevo2);
    check_hevo = (CheckBox) findViewById(R.id.check_hevo);
    linear4 =  findViewById(R.id.linear4);
    img_edit_profile =  findViewById(R.id.img_edit_profile);
    image_hevo4 =  findViewById(R.id.image_hevo4);
    txt_username = (TextView) findViewById(R.id.txt_username);
    vscroll1 =  findViewById(R.id.vscroll1);
    linear5 =  findViewById(R.id.linear5);
    btn_privacy =  findViewById(R.id.btn_privacy);
    bt_hevo_2 =  findViewById(R.id.bt_hevo_2);
    bt_hevo_3 =  findViewById(R.id.bt_hevo_3);
    bt_hevo_4 =  findViewById(R.id.bt_hevo_4);
    bt_hevo_5 =  findViewById(R.id.bt_hevo_5);
    bt_hevo_6 =  findViewById(R.id.bt_hevo_6);
    image_hevo5 =  findViewById(R.id.image_hevo5);
    text_hevo3 = (TextView) findViewById(R.id.text_hevo3);
    image_hevo_bt1 =  findViewById(R.id.image_hevo_bt1);
    image_hevo6 =  findViewById(R.id.image_hevo6);
    text_hevo4 = (TextView) findViewById(R.id.text_hevo4);
    image_hevo_bt2 =  findViewById(R.id.image_hevo_bt2);
    image_hevo7 =  findViewById(R.id.image_hevo7);
    text_hevo5 = (TextView) findViewById(R.id.text_hevo5);
    image_hevo_bt3 =  findViewById(R.id.image_hevo_bt3);
    image_hevo8 =  findViewById(R.id.image_hevo8);
    text_hevo6 = (TextView) findViewById(R.id.text_hevo6);
    image_hevo_bt4 =  findViewById(R.id.image_hevo_bt4);
    image_hevo9 =  findViewById(R.id.image_hevo9);
    text_hevo7 = (TextView) findViewById(R.id.text_hevo7);
    image_hevo_bt5 =  findViewById(R.id.image_hevo_bt5);
    image_hevo10 =  findViewById(R.id.image_hevo10);
    text_hevo8 = (TextView) findViewById(R.id.text_hevo8);
    image_hevo_bt6 =  findViewById(R.id.image_hevo_bt6);
    hevo = getSharedPreferences("mode", Activity.MODE_PRIVATE);
    logout = new AlertDialog.Builder(this);
    auth = FirebaseAuth.getInstance();
    name = getSharedPreferences("name", Activity.MODE_PRIVATE);

    image_hevo1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            finish();
        }
    });

    image_hevo2.setOnClickListener(new View.OnClickListener() {
        @Override
```

```
        public void onClick(View view) {
            _performClick(check_hevo);
        }
    });

    check_hevo.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton param1, boolean
param2)  {
            final boolean _isChecked = param2;
            if (_isChecked) {
                hevo.edit().putString("mode", "true").commit();
                _mod_hevo();
            }
            else {
                if (!_isChecked) {
                    hevo.edit().putString("mode", "false").commit();
                    _mod_hevo();
                }
            }
        }
    });

    img_edit_profile.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            edit.setClass(getApplicationContext(),
EditprofileActivity.class);
            startActivity(edit);
        }
    });

    btn_privacy.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(ProfileActivity.this,
PrivacyActivity.class)); Animatoo.animateFade(ProfileActivity.this);
            clickAnimation(btn_privacy);
        }
    });

    bt_hevo_2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(ProfileActivity.this,
WeightLogsActivity.class)); Animatoo.animateFade(ProfileActivity.this);
            clickAnimation(bt_hevo_2);
        }
    });

    bt_hevo_3.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            support.setAction(Intent.ACTION_VIEW);
            support.setData(Uri.parse("mailto:ayelabolaj@gmail.com"));
            startActivity(support);
        }
    });

    bt_hevo_4.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View view) {
            startActivity(new Intent(ProfileActivity.this,
EditprofileActivity.class)); Animatoo.animateFade(ProfileActivity.this);
            clickAnimation(bt_hevo_4);
        }
    });

    bt_hevo_5.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            String apk = "DietTracker.apk";
            String uri = (getApplicationContext().getPackageName());

            try {
                  android.content.pm.PackageInfo pi =
getPackageManager().getPackageInfo(uri,
android.content.pm.PackageManager.GET_ACTIVITIES);

                  apk = pi.applicationInfo.publicSourceDir;
            } catch (Exception e) {
                  showMessage(e.toString());
            }
            Intent iten = new Intent(Intent.ACTION_SEND);
            iten.setType("*/*");
            iten.putExtra(Intent.EXTRA_STREAM, Uri.fromFile(new
java.io.File(apk)));

            startActivity(Intent.createChooser(iten, "Share And Recommend
Diet Tracker To Friends"));
        }
    });

    bt_hevo_6.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            logout.setTitle("Log Out?");
            logout.setMessage("Are you sure to log out?");
            logout.setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    FirebaseAuth.getInstance().signOut();
                    ToastUtil.showMessage(getApplicationContext(),
"Successfully Logged Out");
                    login.setClass(getApplicationContext(),
CreateaccountActivity.class);
                    startActivity(login);
                    finish();
                }
            });
            logout.setNegativeButton("No", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {

                }
            });
            logout.create().show();
        }
    });
```

```
      auth_updateEmailListener = new OnCompleteListener<Void>() {
         @Override
         public void onComplete(Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };

      auth_updatePasswordListener = new OnCompleteListener<Void>() {
         @Override
         public void onComplete(Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };

      auth_emailVerificationSentListener = new OnCompleteListener<Void>() {
         @Override
         public void onComplete(Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };

      auth_deleteUserListener = new OnCompleteListener<Void>() {
         @Override
         public void onComplete(Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };

      auth_phoneAuthListener = new OnCompleteListener<AuthResult>() {
         @Override
         public void onComplete(Task<AuthResult> task){
            final boolean success = task.isSuccessful();
            final String errorMessage = task.getException() != null ?
task.getException().getMessage() : "";

         }
      };

      auth_updateProfileListener = new OnCompleteListener<Void>() {
         @Override
         public void onComplete(Task<Void> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };
```

```
      auth_googleSignInListener = new OnCompleteListener<AuthResult>() {
         @Override
         public void onComplete(Task<AuthResult> task){
            final boolean success = task.isSuccessful();
            final String errorMessage = task.getException() != null ?
task.getException().getMessage() : "";

         }
      };

      auth_create_user_listener = new OnCompleteListener<AuthResult>() {
         @Override
         public void onComplete(Task<AuthResult> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };

      authsign_in_listener = new OnCompleteListener<AuthResult>() {
         @Override
         public void onComplete(Task<AuthResult> param1) {
            final boolean success = param1.isSuccessful();
            final String errorMessage = param1.getException() != null ?
param1.getException().getMessage() : "";

         }
      };

      _auth_reset_password_listener = new OnCompleteListener<Void>() {
         @Override
         public void onComplete(Task<Void> param1) {
            final boolean success = param1.isSuccessful();

         }
      };
   }

   private void initializeLogic() {
      _NavStatusBarColor("#FFFFFF", "#FFFFFF");
      _DARK_ICONS();
      changeActivityFont("railway_semisolid");
      _addCardView(image_hevo4, 0, 360, 4, 4, true, "#101D24");
      _rippleRoundStroke(img_edit_profile, "#FF4433", "#FFFFFF", 360, 2,
"#FF4433");
      _rippleRoundStroke(btn_privacy, "#F3F7FB", "#FFFFFF", 80, 0,
"#FFFFFF");
      _rippleRoundStroke(bt_hevo_2, "#F3F7FB", "#FFFFFF", 80, 0,
"#FFFFFF");
      _rippleRoundStroke(bt_hevo_3, "#F3F7FB", "#FFFFFF", 80, 0,
"#FFFFFF");
      _rippleRoundStroke(bt_hevo_4, "#F3F7FB", "#FFFFFF", 80, 0,
"#FFFFFF");
      _rippleRoundStroke(bt_hevo_5, "#F3F7FB", "#FFFFFF", 80, 0,
"#FFFFFF");
      _rippleRoundStroke(bt_hevo_6, "#F3F7FB", "#FFFFFF", 80, 0,
"#FFFFFF");
      _ICC(image_hevo1, "#4D4D4D", "#FFFFFF");
      _ICC(image_hevo2, "#4D4D4D", "#FFFFFF");
      _ICC(image_hevo5, "#4D4D4D", "#FFFFFF");
```

```
      _ICC(image_hevo6, "#4D4D4D", "#FFFFFF");
      _ICC(image_hevo7, "#4D4D4D", "#FFFFFF");
      _ICC(image_hevo8, "#4D4D4D", "#FFFFFF");
      _ICC(image_hevo9, "#4D4D4D", "#FFFFFF");
      _ICC(image_hevo10, "#4D4D4D", "#FFFFFF");
      _ICC(image_hevo_bt1, "#4D4D4D", "#FFFFFF");
      _ICC(image_hevo_bt2, "#4D4D4D", "#FFFFFF");
      _ICC(image_hevo_bt3, "#4D4D4D", "#FFFFFF");
      _ICC(image_hevo_bt4, "#4D4D4D", "#FFFFFF");
      _ICC(image_hevo_bt5, "#4D4D4D", "#FFFFFF");
      _ICC(image_hevo_bt6, "#4D4D4D", "#FFFFFF");
      _ICC(img_edit_profile, "#FFFFFF", "#F3F7FB");
      txt_username.setText(name.getString("name", ""));
      removeScrollBar(vscroll1);
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   public void _DARK_ICONS () {

getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_
STATUS_BAR);
   }


   public void changeActivityFont (final String fontname) {
      fontName = "fonts/".concat(fontname.concat(".ttf"));
      overrideFonts(this,getWindow().getDecorView());
   }
   private void overrideFonts(final android.content.Context context, final
View v) {

      try {
         Typeface
         typeace = Typeface.createFromAsset(getAssets(), fontName);;
         if ((v instanceof ViewGroup)) {
            ViewGroup vg = (ViewGroup) v;
            for (int i = 0;
            i < vg.getChildCount();
            i++) {
               View child = vg.getChildAt(i);
               overrideFonts(context, child);
            }
         }
         else {
            if ((v instanceof TextView)) {
               ((TextView) v).setTypeface(typeace);
            }
            else {
               if ((v instanceof EditText )) {
                  ((EditText) v).setTypeface(typeace);
               }
```

```
                else {
                    if ((v instanceof Button)) {
                        ((Button) v).setTypeface(typeace);
                    }
                }
            }
        }
      }
      catch(Exception e)

      {
      };
   }


   public void _rippleRoundStroke (final View view, final String _focus,
final String _pressed, final double _round, final double msgtroke, final
String msgtrokeclr) {
      android.graphics.drawable.GradientDrawable GG = new
android.graphics.drawable.GradientDrawable();
      GG.setColor(Color.parseColor(_focus));
      GG.setCornerRadius((float)_round);
      GG.setStroke((int) msgtroke,
      Color.parseColor("#" + msgtrokeclr.replace("#", "")));
      android.graphics.drawable.RippleDrawable RE = new
android.graphics.drawable.RippleDrawable(new
android.content.res.ColorStateList(new int[][]{new int[]{}}, new int[]{
Color.parseColor(_pressed)}), GG, null);
      view.setBackground(RE);
   }


   public void _mod_hevo () {

   }


   public void _addCardView (final View _layoutView, final double _margins,
final double _cornerRadius, final double _cardElevation, final double
_cardMaxElevation, final boolean _preventCornerOverlap, final String
_backgroundColor) {
      androidx.cardview.widget.CardView cv = new
androidx.cardview.widget.CardView(this);
      LinearLayout.LayoutParams lp = new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,
LinearLayout.LayoutParams.WRAP_CONTENT);
      int m = (int)_margins;
      lp.setMargins(m,m,m,m);
      cv.setLayoutParams(lp);
      int c = Color.parseColor(_backgroundColor);
      cv.setCardBackgroundColor(c);
      cv.setRadius((float)_cornerRadius);
      cv.setCardElevation((float)_cardElevation);
      cv.setMaxCardElevation((float)_cardMaxElevation);
      cv.setPreventCornerOverlap(_preventCornerOverlap);
      if(_layoutView.getParent() instanceof LinearLayout){
         ViewGroup vg = ((ViewGroup)_layoutView.getParent());
         vg.removeView(_layoutView);
         vg.removeAllViews();
         vg.addView(cv);
         cv.addView(_layoutView);
```

```
      }else{

      }
   }


   public void setCornerRadius (final View view, final double radius, final
double msghadow, final String color) {
      android.graphics.drawable.GradientDrawable ab = new
android.graphics.drawable.GradientDrawable();

      ab.setColor(Color.parseColor(color));
      ab.setCornerRadius((float) radius);
      view.setElevation((float) msghadow);
      view.setBackground(ab);

      //Add More block in OnCreateActivity :
   }


   public void _NavStatusBarColor (final String _color1, final String
_color2) {
      if (Build.VERSION.SDK_INT > Build.VERSION_CODES.LOLLIPOP) {
         Window w = this.getWindow();
w.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);
w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
         w.setStatusBarColor(Color.parseColor("#" + _color1.replace("#",
"")));  w.setNavigationBarColor(Color.parseColor("#" + _color2.replace("#",
"")));
      }
   }


   public void _ICC (final ImageView _img, final String _c1, final String
_c2) {
      _img.setImageTintList(new android.content.res.ColorStateList(new
int[][] {{-
android.R.attr.state_pressed},{android.R.attr.state_pressed}},new
int[]{Color.parseColor(_c1), Color.parseColor(_c2)}));
   }


   public void _performClick (final View view) {
      view.performClick();
      //by hevo team
   }


   public void clickAnimation (final View view) {
      ScaleAnimation fade_in = new ScaleAnimation(0.9f, 1f, 0.9f, 1f,
Animation.RELATIVE_TO_SELF, 0.5f, Animation.RELATIVE_TO_SELF, 0.7f);
      fade_in.setDuration(300);
      fade_in.setFillAfter(true);
      view.startAnimation(fade_in);
   }


   public void removeScrollBar (final View view) {
      view.setHorizontalScrollBarEnabled(false);
      view.setOverScrollMode(ScrollView.OVER_SCROLL_NEVER);
```

```
    }


    @Deprecated
    public void showMessage(String msg) {
        Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
    }

    @Deprecated
    public int getLocationX(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Recipe Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.content.Intent;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.widget.ListView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;
import java.util.Random;


public class RecipeActivity extends AppCompatActivity {

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.recipe);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
   }

   private void initializeLogic() {
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
```

```
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Recipe Search Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
```

```java
import android.content.SharedPreferences;
import android.graphics.drawable.GradientDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.coordinatorlayout.widget.CoordinatorLayout;

import com.bumptech.glide.Glide;
import com.google.android.material.appbar.AppBarLayout;
import com.google.gson.Gson;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;


public class RecipesearchedActivity extends AppCompatActivity {

   private Toolbar _toolbar;
   private AppBarLayout _app_bar;
   private CoordinatorLayout _coordinator;
   private HashMap<String, Object> h = new HashMap<>();
   private HashMap<String, Object> pn = new HashMap<>();
   private String name = "";
   private String img = "";
   private HashMap<String, Object> foods = new HashMap<>();
   private String product = "";
   private String calories = "";
   private String protein = "";
   private String carbo = "";
   private String fat = "";
   private String sugar = "";
   private String cholesterol = "";
   private String ingredients = "";
   private HashMap<String, Object> param = new HashMap<>();
   private HashMap<String, Object> recipe = new HashMap<>();
   private String fiber = "";

   private ArrayList<HashMap<String, Object>> listmap2 = new ArrayList<>();
   private ArrayList<HashMap<String, Object>> listmap = new ArrayList<>();

   private LinearLayout linrLoading;
   private ListView listview1;
   private ProgressBar progressbar1;
   private TextView textview3;
```

```
   private RequestNetwork w;
   private RequestNetwork.RequestListener _w_request_listener;
   private AlertDialog.Builder a;
   private SharedPreferences holdrec;
   private Intent i = new Intent();

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.recipesearched);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      _app_bar = (AppBarLayout) findViewById(R.id._app_bar);
      _coordinator = (CoordinatorLayout) findViewById(R.id._coordinator);
      _toolbar = (Toolbar) findViewById(R.id._toolbar);
      setSupportActionBar(_toolbar);
      getSupportActionBar().setDisplayHomeAsUpEnabled(true);
      getSupportActionBar().setHomeButtonEnabled(true);
      _toolbar.setNavigationOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            onBackPressed();
         }
      });
      linrLoading =  findViewById(R.id.linrLoading);
      listview1 = (ListView) findViewById(R.id.listview1);
      progressbar1 = (ProgressBar) findViewById(R.id.progressbar1);
      textview3 = (TextView) findViewById(R.id.textview3);
      w = new RequestNetwork(this);
      a = new AlertDialog.Builder(this);
      holdrec = getSharedPreferences("holdrec", Activity.MODE_PRIVATE);

      listview1.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
         @Override
         public void onItemClick(AdapterView<?> param1, View param2, int
param3, long param4) {
            final int position = param3;
            try{
               i.putExtra("name",
listmap.get((int)position).get("name").toString());
               i.putExtra("img",
listmap.get((int)position).get("img").toString());
               i.putExtra("ingredients",
listmap.get((int)position).get("ingredients").toString());
               i.putExtra("calories",
listmap.get((int)position).get("calories").toString());
               i.putExtra("fat",
listmap.get((int)position).get("fat").toString());
               i.putExtra("carbo",
listmap.get((int)position).get("carbo").toString());
               i.putExtra("protein",
listmap.get((int)position).get("protein").toString());
               i.putExtra("fiber",
listmap.get((int)position).get("fiber").toString());
               i.putExtra("sugar",
```

```
listmap.get((int)position).get("sugar").toString());
                i.putExtra("position", String.valueOf((long)(position)));
                i.putExtra("cholesterol",
listmap.get((int)position).get("cholesterol").toString());
                i.putExtra("text",
listmap.get((int)position).get("text").toString());
                i.setClass(getApplicationContext(),
ViewrecipeActivity.class);
                startActivity(i);
             }catch(Exception e){
                ToastUtil.showMessage(getApplicationContext(),
e.toString());
             }
          }
       });

       _w_request_listener = new RequestNetwork.RequestListener() {
          @Override
          public void onResponse(String param1, String param2,
HashMap<String, Object> param3) {
             final String tag = param1;
             final String response = param2;
             final HashMap<String, Object> _responseHeaders = param3;
             linrLoading.setVisibility(View.GONE);
             if (tag.equals("a")) {
                try {
                   org.json.JSONObject obj = new
org.json.JSONObject(response);
                   org.json.JSONArray array = obj.getJSONArray("hits");
                   for(int i=0;i<array.length();i++){ org.json.JSONObject
object = array.getJSONObject(i);
                      if (object.getJSONObject("recipe").has("label")) {
                         name =
object.getJSONObject("recipe").getString("label").toString(); }

                      if (object.getJSONObject("recipe").has("image")) {
                         img =
object.getJSONObject("recipe").getString("image").toString(); }

                      if (object.getJSONObject("recipe").has("calories")) {
                         calories =
object.getJSONObject("recipe").getString("calories").toString(); }

                      if
(object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObje
ct("FAT").has("quantity")) {
                         fat =
object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObjec
t("FAT").getString("quantity").toString(); }

                      if
(object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObje
ct("CHOCDF").has("quantity")) {
                         carbo =
object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObjec
t("CHOCDF").getString("quantity").toString(); }

                      if
(object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObje
ct("SUGAR").has("quantity")) {
                         sugar =
```

```
object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObjec
t("SUGAR").getString("quantity").toString(); }

                    if
(object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObje
ct("FIBTG").has("quantity")) {
                        fiber =
object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObjec
t("FIBTG").getString("quantity").toString(); }

                    if
(object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObje
ct("PROCNT").has("quantity")) {
                        protein =
object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObjec
t("PROCNT").getString("quantity").toString(); }

                    if
(object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObje
ct("CHOLE").has("quantity")) {
                        cholesterol =
object.getJSONObject("recipe").getJSONObject("totalNutrients").getJSONObjec
t("CHOLE").getString("quantity").toString(); }

                    org.json.JSONArray arrayIng =
object.getJSONObject("recipe").getJSONArray("ingredients");
                    for(int j=0;j<arrayIng.length();j++){
                       org.json.JSONObject objects =
arrayIng.getJSONObject(j);
                       if (objects.has("text")){
                          ingredients = objects.getString("text");
                       }
                       recipe = new HashMap<>();
                       recipe.put("ingredients", ingredients);
                       listmap2.add(recipe);
                    }
                    holdrec.edit().putString("recipe", new
Gson().toJson(listmap2)).commit();
                    param = new HashMap<>();
                    param.put("name", name);
                    param.put("img", img);
                    param.put("calories", calories);
                    param.put("fat", fat);
                    param.put("carbo", carbo);
                    param.put("sugar", sugar);
                    param.put("fiber", fiber);
                    param.put("protein", protein);
                    param.put("cholesterol", cholesterol);
                    param.put("ingredients", new Gson().toJson(listmap2));
                    param.put("text", ingredients);
                    listmap.add(param);
                 }
                 listview1.setAdapter(new Listview1Adapter(listmap));

((BaseAdapter)listview1.getAdapter()).notifyDataSetChanged();
              } catch(final org.json.JSONException ex) {
                 ToastUtil.showMessage(getApplicationContext(),
ex.toString());
              }
           }
        }
```

```
        @Override
        public void onErrorResponse(String param1, String param2) {
           final String tag = param1;
           final String message = param2;

        }
     };
  }

  private void initializeLogic() {
     w.startRequestNetwork(RequestNetworkController.GET,
getIntent().getStringExtra("search"), "a", _w_request_listener);
     setTitle("Search Results");
  }

  @Override
  protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
     super.onActivityResult(requestCode, resultCode, data);
     switch (requestCode) {

        default:
        break;
     }
  }

  public class Listview1Adapter extends BaseAdapter {
     ArrayList<HashMap<String, Object>> data;
     public Listview1Adapter(ArrayList<HashMap<String, Object>> arr) {
        data = arr;
     }

     @Override
     public int getCount() {
        return data.size();
     }

     @Override
     public HashMap<String, Object> getItem(int index) {
        return data.get(index);
     }

     @Override
     public long getItemId(int index) {
        return index;
     }
     @Override
     public View getView(final int position, View v, ViewGroup container)
{
        LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
        View view = v;
        if (view == null) {
           view = inflater.inflate(R.layout.txt, null);
        }

        final TextView textview1 = (TextView)
view.findViewById(R.id.textview1);
        final androidx.cardview.widget.CardView cardview1 =
```

```
(androidx.cardview.widget.CardView) view.findViewById(R.id.cardview1);
        final TextView textview2 = (TextView)
view.findViewById(R.id.textview2);
        final androidx.cardview.widget.CardView cardview2 =
(androidx.cardview.widget.CardView) view.findViewById(R.id.cardview2);
        final LinearLayout linear19 =  view.findViewById(R.id.linear19);
        final LinearLayout linear6 =  view.findViewById(R.id.linear6);
        final LinearLayout linear2 =  view.findViewById(R.id.linear2);
        final ImageView imageview1 =  view.findViewById(R.id.imageview1);
        final ImageView imageview2 =  view.findViewById(R.id.imageview2);
        final LinearLayout imageview3 =
view.findViewById(R.id.imageview3);
        final LinearLayout linear3 =  view.findViewById(R.id.linear3);
        final LinearLayout linear4 =  view.findViewById(R.id.linear4);
        final LinearLayout linear5 =  view.findViewById(R.id.linear5);
        final TextView txtBigBannerHead = (TextView)
view.findViewById(R.id.txtBigBannerHead);
        final LinearLayout linear7 =  view.findViewById(R.id.linear7);
        final TextView txtBigDescrip = (TextView)
view.findViewById(R.id.txtBigDescrip);
        final Button btnAdd = (Button) view.findViewById(R.id.btnAdd);
        final LinearLayout linrBack =  view.findViewById(R.id.linrBack);
        final ImageView imgDisplay =  view.findViewById(R.id.imgDisplay);
        final LinearLayout linear18 =  view.findViewById(R.id.linear18);
        final TextView txtMeal = (TextView)
view.findViewById(R.id.txtMeal);
        final LinearLayout linear17 =  view.findViewById(R.id.linear17);
        final ImageView imageview8 =  view.findViewById(R.id.imageview8);
        final TextView txtCalories = (TextView)
view.findViewById(R.id.txtCalories);
        final ImageView imageview9 =  view.findViewById(R.id.imageview9);
        final TextView txtIngredientsQty = (TextView)
view.findViewById(R.id.txtIngredientsQty);
        final TextView txtIngredients = (TextView)
view.findViewById(R.id.txtIngredients);
        final TextView txtMealType = (TextView)
view.findViewById(R.id.txtMealType);
        final TextView txtFatFig = (TextView)
view.findViewById(R.id.txtFatFig);
        final TextView txtCarboFig = (TextView)
view.findViewById(R.id.txtCarboFig);
        final TextView txtProteinFig = (TextView)
view.findViewById(R.id.txtProteinFig);
        final TextView txtSugar = (TextView)
view.findViewById(R.id.txtSugar);
        final TextView txtFiber = (TextView)
view.findViewById(R.id.txtFiber);
        final TextView txtCholestrol = (TextView)
view.findViewById(R.id.txtCholestrol);

        cardview1.setVisibility(View.GONE);

Glide.with(getApplicationContext()).load(Uri.parse(listmap.get((int)positio
n).get("img").toString())).into(imgDisplay);

txtMeal.setText(listmap.get((int)position).get("name").toString());
        if (listmap.get((int)position).get("calories").toString().length()
> 4) {
            txtCalories.setText("Calories:
".concat(listmap.get((int)position).get("calories").toString().substring((i
nt)(0), (int)(4)).concat("")));
```

```
        }

txtFatFig.setText(listmap.get((int)position).get("fat").toString());

txtCarboFig.setText(listmap.get((int)position).get("carbo").toString());

txtSugar.setText(listmap.get((int)position).get("sugar").toString());

txtProteinFig.setText(listmap.get((int)position).get("protein").toString())
;

txtFatFig.setText(listmap.get((int)position).get("fiber").toString());

txtBigDescrip.setText(listmap.get((int)position).get("cholesterol").toStrin
g());
        cardview1.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)30, 0xFFFFFFFF));
        cardview2.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)20, 0xFFFFFFFF));
        btnAdd.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b, int c, int d) {
this.setCornerRadius(a); this.setStroke(b, c); this.setColor(d); return
this; } }.getIns((int)30, (int)2, 0xFFFAFAFA, 0xFF000000));
        linear19.setVisibility(View.GONE);
        textview1.setVisibility(View.GONE);
        textview2.setVisibility(View.GONE);

        return view;
      }
   }

   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }

   @Deprecated
   public int getRandom(int min, int max) {
      Random random = new Random();
      return random.nextInt(max - min + 1) + min;
   }

   @Deprecated
   public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
```

```
{
      ArrayList<Double> result = new ArrayList<Double>();
      SparseBooleanArray arr = _list.getCheckedItemPositions();
      for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
         if (arr.valueAt(iIdx))
         result.add((double)arr.keyAt(iIdx));
      }
      return result;
   }

   @Deprecated
   public float getDip(int input) {
      return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
   }

   @Deprecated
   public int getDisplayWidthPixels() {
      return getResources().getDisplayMetrics().widthPixels;
   }

   @Deprecated
   public int getDisplayHeightPixels() {
      return getResources().getDisplayMetrics().heightPixels;
   }
}
```

Recipe View Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.drawable.GradientDrawable;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;
```

```
public class RecipeviewActivity extends AppCompatActivity {

   private String meal = "";
   private String date = "";
   private String recipe = "";
   private double position = 0;
   private HashMap<String, Object> GetRecipe = new HashMap<>();

   private ArrayList<HashMap<String, Object>> Recipe_List = new
ArrayList<>();

   private LinearLayout toolbar;
   private LinearLayout linear1;
   private ImageView back_img;
   private TextView toolbar_txt;
   private ScrollView vscroll1;
   private LinearLayout linear2;
   private TextView txtMeal;
   private LinearLayout linear3;
   private TextView textview3;
   private TextView txtRecipe;
   private TextView txtDate;

   private SharedPreferences Recipe;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.recipeview);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      toolbar =  findViewById(R.id.toolbar);
      linear1 =  findViewById(R.id.linear1);
      back_img =  findViewById(R.id.back_img);
      toolbar_txt = (TextView) findViewById(R.id.toolbar_txt);
      vscroll1 =  findViewById(R.id.vscroll1);
      linear2 =  findViewById(R.id.linear2);
      txtMeal = (TextView) findViewById(R.id.txtMeal);
      linear3 =  findViewById(R.id.linear3);
      textview3 = (TextView) findViewById(R.id.textview3);
      txtRecipe = (TextView) findViewById(R.id.txtRecipe);
      txtDate = (TextView) findViewById(R.id.txtDate);
      Recipe = getSharedPreferences("Recipe", Activity.MODE_PRIVATE);

      back_img.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            finish();
         }
      });
   }

   private void initializeLogic() {
      _getRecipe();
      try{
```

```
         txtMeal.setText(meal);
         txtDate.setText(date);
         txtRecipe.setText(recipe);
      }catch(Exception e){

      }
      linear3.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)60, 0xFF));
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   public void _getRecipe () {
      try{
         if (!Recipe.getString("Recipe", "").equals("")) {
            Recipe_List = new Gson().fromJson(Recipe.getString("Recipe",
""), new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
            GetRecipe = Recipe_List.get((int)position);
            meal = GetRecipe.get("Food").toString();
            date = GetRecipe.get("Date").toString();
            recipe = GetRecipe.get("Ingredt").toString();
         }
      }catch(Exception e){

      }
   }


   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }

   @Deprecated
   public int getRandom(int min, int max) {
      Random random = new Random();
```

```
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Request Network

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;

import java.util.HashMap;

public class RequestNetwork {
private HashMap<String, Object> params = new HashMap<>();
private HashMap<String, Object> headers = new HashMap<>();

private Activity activity;

private int requestType = 0;

public RequestNetwork(Activity activity) {
this.activity = activity;
}

public void setHeaders(HashMap<String, Object> headers) {
this.headers = headers;
```

```
}

public void setParams(HashMap<String, Object> params, int requestType) {
this.params = params;
this.requestType = requestType;
}

public HashMap<String, Object> getParams() {
return params;
}

public HashMap<String, Object> getHeaders() {
return headers;
}

public Activity getActivity() {
return activity;
}

public int getRequestType() {
return requestType;
}

public void startRequestNetwork(String method, String url, String tag,
RequestListener requestListener) {
RequestNetworkController.getInstance().execute(this, method, url, tag,
requestListener);
}

public interface RequestListener {
public void onResponse(String tag, String response, HashMap<String, Object>
responseHeaders);
public void onErrorResponse(String tag, String message);
}
}
```

Request Network Controller

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import com.google.gson.Gson;

import java.io.IOException;
import java.security.cert.CertificateException;
import java.util.HashMap;
import java.util.concurrent.TimeUnit;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.SSLSocketFactory;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
```

```
import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.FormBody;
import okhttp3.Headers;
import okhttp3.HttpUrl;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;

public class RequestNetworkController {
public static final String GET      = "GET";
public static final String POST     = "POST";
public static final String PUT      = "PUT";
public static final String DELETE   = "DELETE";

public static final int REQUEST_PARAM = 0;
public static final int REQUEST_BODY  = 1;

private static final int SOCKET_TIMEOUT = 15000;
private static final int READ_TIMEOUT   = 25000;

protected OkHttpClient client;

private static RequestNetworkController mInstance;

public static synchronized RequestNetworkController getInstance() {
if(mInstance == null) {
mInstance = new RequestNetworkController();
}
return mInstance;
}

private OkHttpClient getClient() {
if (client == null) {
OkHttpClient.Builder builder = new OkHttpClient.Builder();

try {
final TrustManager[] trustAllCerts = new TrustManager[]{
new X509TrustManager() {
@Override
public void checkClientTrusted(java.security.cert.X509Certificate[] chain,
String authType)
throws CertificateException {
}

@Override
public void checkServerTrusted(java.security.cert.X509Certificate[] chain,
String authType)
throws CertificateException {
}

@Override
public java.security.cert.X509Certificate[] getAcceptedIssuers() {
return new java.security.cert.X509Certificate[]{};
}
}
};

final SSLContext sslContext = SSLContext.getInstance("TLS");
sslContext.init(null, trustAllCerts, new java.security.SecureRandom());
```

```
final SSLSocketFactory sslSocketFactory = sslContext.getSocketFactory();
builder.sslSocketFactory(sslSocketFactory, (X509TrustManager)
trustAllCerts[0]);
builder.connectTimeout(SOCKET_TIMEOUT, TimeUnit.MILLISECONDS);
builder.readTimeout(READ_TIMEOUT, TimeUnit.MILLISECONDS);
builder.writeTimeout(READ_TIMEOUT, TimeUnit.MILLISECONDS);
builder.hostnameVerifier(new HostnameVerifier() {
@Override
public boolean verify(String hostname, SSLSession session) {
return true;
}
});
} catch (Exception e) {
}

client = builder.build();
}

return client;
}

public void execute(final RequestNetwork requestNetwork, String method,
String url, final String tag, final RequestNetwork.RequestListener
requestListener) {
Request.Builder reqBuilder = new Request.Builder();
Headers.Builder headerBuilder = new Headers.Builder();

if(requestNetwork.getHeaders().size() > 0) {
HashMap<String, Object> headers = requestNetwork.getHeaders();

for(HashMap.Entry<String, Object> header : headers.entrySet()) {
headerBuilder.add(header.getKey(), String.valueOf(header.getValue()));
}
}

try {
if (requestNetwork.getRequestType() == REQUEST_PARAM) {
if (method.equals(GET)) {
HttpUrl.Builder httpBuilder;

try {
httpBuilder = HttpUrl.parse(url).newBuilder();
} catch (NullPointerException ne) {
throw new NullPointerException("unexpected url: " + url);
}

if (requestNetwork.getParams().size() > 0) {
HashMap<String, Object> params = requestNetwork.getParams();

for (HashMap.Entry<String, Object> param : params.entrySet()) {
httpBuilder.addQueryParameter(param.getKey(),
String.valueOf(param.getValue()));
}
}

reqBuilder.url(httpBuilder.build()).headers(headerBuilder.build()).get();
} else {
FormBody.Builder formBuilder = new FormBody.Builder();
if (requestNetwork.getParams().size() > 0) {
HashMap<String, Object> params = requestNetwork.getParams();
```

```
for (HashMap.Entry<String, Object> param : params.entrySet()) {
formBuilder.add(param.getKey(), String.valueOf(param.getValue()));
}
}

RequestBody reqBody = formBuilder.build();

reqBuilder.url(url).headers(headerBuilder.build()).method(method, reqBody);
}
} else {
RequestBody reqBody =
RequestBody.create(okhttp3.MediaType.parse("application/json"), new
Gson().toJson(requestNetwork.getParams()));

if (method.equals(GET)) {
reqBuilder.url(url).headers(headerBuilder.build()).get();
} else {
reqBuilder.url(url).headers(headerBuilder.build()).method(method, reqBody);
}
}

Request req = reqBuilder.build();

getClient().newCall(req).enqueue(new Callback() {
@Override
public void onFailure(Call call, final IOException e) {
requestNetwork.getActivity().runOnUiThread(new Runnable() {
@Override
public void run() {
requestListener.onErrorResponse(tag, e.getMessage());
}
});
}

@Override
public void onResponse(Call call, final Response response) throws
IOException {
final String responseBody = response.body().string().trim();
requestNetwork.getActivity().runOnUiThread(new Runnable() {
@Override
public void run() {
Headers b = response.headers();
HashMap<String, Object> map = new HashMap<>();
for(String s : b.names()) {
map.put(s, b.get(s) != null ? b.get(s) : "null");
}
requestListener.onResponse(tag, responseBody, map);
}
});
}
});
} catch (Exception e) {
requestListener.onErrorResponse(tag, e.getMessage());
}
}
}
```

Scancode Activity

```java
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.Manifest;
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.widget.ListView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.coordinatorlayout.widget.CoordinatorLayout;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.core.content.FileProvider;

import com.budiyev.android.codescanner.CodeScanner;
import com.budiyev.android.codescanner.CodeScannerView;
import com.budiyev.android.codescanner.DecodeCallback;
import com.google.android.material.appbar.AppBarLayout;
import com.google.zxing.Result;

import java.io.File;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;


public class ScancodeActivity extends AppCompatActivity {

// Make a camera request code
   public final int REQ_CD_CAMERA = 101;

   private Toolbar _toolbar;
   private AppBarLayout _app_bar;
   private CoordinatorLayout _coordinator;
   private String a = "";
   private String b = "";
   private HashMap<String, Object> map = new HashMap<>();
   private String url = "";
   private String gprl = "";
   private String suprl = "";
   private String data = "";
   private  CodeScanner mCodeScanner;

   private CodeScannerView scannerview;
```

```
   private Intent camera = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
   private File      file_camera;
   private Intent i = new Intent();
   private SharedPreferences sp;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.scancode);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      if (ContextCompat.checkSelfPermission(this,
Manifest.permission.CAMERA) == PackageManager.PERMISSION_DENIED
      || ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_DENIED
      || ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_DENIED) {
         ActivityCompat.requestPermissions(this, new String[]
{Manifest.permission.CAMERA, Manifest.permission.READ_EXTERNAL_STORAGE,
Manifest.permission.WRITE_EXTERNAL_STORAGE}, 1000);
      }
      else {
         initializeLogic();
      }
   }

   @Override
   public void onRequestPermissionsResult(int requestCode, String[]
permissions, int[] grantResults) {
      super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
      if (requestCode == 1000) {
         initializeLogic();
      }
   }

   private void initialize(Bundle savedInstanceState) {
      _app_bar = (AppBarLayout) findViewById(R.id._app_bar);
      _coordinator = (CoordinatorLayout) findViewById(R.id._coordinator);
      _toolbar = (Toolbar) findViewById(R.id._toolbar);
      setSupportActionBar(_toolbar);
      getSupportActionBar().setDisplayHomeAsUpEnabled(true);
      getSupportActionBar().setHomeButtonEnabled(true);
      _toolbar.setNavigationOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            onBackPressed();
         }
      });

//    This code works after the CodeScannerView has scanned the QR code or
BarCode, it decodes and passes the uri to the next activity
//    calling the scannerview to begin scanning
      scannerview = (CodeScannerView) findViewById(R.id.scannerview);

           file_camera =
FileUtil.createNewPictureFile(getApplicationContext());
      Uri _uri_camera = null;
      if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.N) {
```

```
//       initiate the file provider on the URI uri_camera
         _uri_camera= FileProvider.getUriForFile(getApplicationContext(),
getApplicationContext().getPackageName() + ".provider",      file_camera);
      }
      else {
         _uri_camera = Uri.fromFile(     file_camera);
      }
      camera.putExtra(MediaStore.EXTRA_OUTPUT, _uri_camera);
      camera.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
      sp = getSharedPreferences("sp", Activity.MODE_PRIVATE);
   }

   private void initializeLogic() {

      //Scan barcode or QR code with the code below
      CodeScannerView scannerView = findViewById(R.id.scannerview);
      mCodeScanner = new CodeScanner(this, scannerView);

      mCodeScanner.setDecodeCallback(new DecodeCallback() {
                @Override public void onDecoded(@NonNull final Result
result) { runOnUiThread(new Runnable() {
                                          @Override
                                           public void run() {
                                                    data =
result.getText();
                                                    }
                                       });
               }});
      // On completion of barcode, result parameter is called from the
method ocDecoded to get the text
//    then the text is sent to scanpreviewActivity. This is done so that we
cam search the code
//    we get the scanning to be searched on openfoodfacts
      mCodeScanner.setDecodeCallback(new DecodeCallback() {
                  @Override public void onDecoded(@NonNull final Result
result) { runOnUiThread(new Runnable() {
                                          @Override
                                           public void run() {

i.setClass(getApplicationContext(), ScanPreviewActivity.class);
                                          i.putExtra("c",
result.getText());
                                          startActivity(i);
                                          finish();
                                          } }

                                      ); }
                           }
                );
      mCodeScanner.startPreview();
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
```

```
      }
   }

   //no code

   @Override
   protected void onPostCreate(Bundle savedInstanceState) {
      super.onPostCreate(savedInstanceState);

   }

   @Override
   public void onBackPressed() {
      finish();
   }
   public void _privateVariables () {
   }
   android.content.pm.ShortcutManager shortcutManager;
   android.content.pm.ShortcutInfo shortcut;
   android.content.pm.ShortcutInfo shortcut1,shortcut2; //Add this if you
are using double shortcuts

   {
   }


   public void _rippleRoundStroke (final View view, final String _focus,
final String _pressed, final double _round, final double msgtroke, final
String msgtrokeclr) {
      android.graphics.drawable.GradientDrawable GG = new
android.graphics.drawable.GradientDrawable();
      GG.setColor(Color.parseColor(_focus));
      GG.setCornerRadius((float)_round);
      GG.setStroke((int) msgtroke,
      Color.parseColor("#" + msgtrokeclr.replace("#", "")));
      android.graphics.drawable.RippleDrawable RE = new
android.graphics.drawable.RippleDrawable(new
android.content.res.ColorStateList(new int[][]{new int[]{}}, new int[]{
Color.parseColor(_pressed)}), GG, null);
      view.setBackground(RE);
   }


   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
```

```
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Scanner Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
```

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.coordinatorlayout.widget.CoordinatorLayout;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.core.content.FileProvider;

import com.budiyev.android.codescanner.CodeScanner;
import com.budiyev.android.codescanner.CodeScannerView;
import com.google.android.material.appbar.AppBarLayout;

import java.io.File;
import java.util.ArrayList;
import java.util.Random;


public class ScannerActivity extends AppCompatActivity {
   public final int REQ_CD_C = 101;

   private Toolbar _toolbar;
   private AppBarLayout _app_bar;
   private CoordinatorLayout _coordinator;
   private  CodeScanner mCodeScanner;

   private CodeScannerView scannerview;
   private LinearLayout linear1;
   private TextView textview1;

   private Intent c = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
   private File _file_c;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.scanner);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      if (ContextCompat.checkSelfPermission(this,
Manifest.permission.CAMERA) == PackageManager.PERMISSION_DENIED
      || ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_DENIED
      || ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_DENIED) {
         ActivityCompat.requestPermissions(this, new String[]
{Manifest.permission.CAMERA, Manifest.permission.READ_EXTERNAL_STORAGE,
Manifest.permission.WRITE_EXTERNAL_STORAGE}, 1000);
      }
      else {
         initializeLogic();
      }
   }

   @Override
   public void onRequestPermissionsResult(int requestCode, String[]
permissions, int[] grantResults) {
      super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
      if (requestCode == 1000) {
         initializeLogic();
```

```
      }
   }

   private void initialize(Bundle savedInstanceState) {
      _app_bar = (AppBarLayout) findViewById(R.id._app_bar);
      _coordinator = (CoordinatorLayout) findViewById(R.id._coordinator);
      _toolbar = (Toolbar) findViewById(R.id._toolbar);
      setSupportActionBar(_toolbar);
      getSupportActionBar().setDisplayHomeAsUpEnabled(true);
      getSupportActionBar().setHomeButtonEnabled(true);
      _toolbar.setNavigationOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            onBackPressed();
         }
      });
      scannerview = (CodeScannerView) findViewById(R.id.scannerview);
      linear1 =  findViewById(R.id.linear1);
      textview1 = (TextView) findViewById(R.id.textview1);
      _file_c = FileUtil.createNewPictureFile(getApplicationContext());
      Uri _uri_c = null;
      if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.N) {
         _uri_c= FileProvider.getUriForFile(getApplicationContext(),
getApplicationContext().getPackageName() + ".provider", _file_c);
      }
      else {
         _uri_c = Uri.fromFile(_file_c);
      }
      c.putExtra(MediaStore.EXTRA_OUTPUT, _uri_c);
      c.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
   }

   private void initializeLogic() {



   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }
```

```
    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.content.Intent;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
```

```
import java.util.ArrayList;
import java.util.Random;
import java.util.Timer;
import java.util.TimerTask;


public class ScanPreviewActivity extends AppCompatActivity {
   private Timer timer = new Timer();

   private LinearLayout linear1;
   private LinearLayout linear2;
   private ImageView imageview1;
   private TextView textview1;
   private TextView textview2;
   private ProgressBar progressbar1;

   private Intent i = new Intent();
   private Intent in = new Intent();
   private TimerTask t;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.scan_preview);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      linear1 =  findViewById(R.id.linear1);
      linear2 =  findViewById(R.id.linear2);
      imageview1 =  findViewById(R.id.imageview1);
      textview1 = (TextView) findViewById(R.id.textview1);
      textview2 = (TextView) findViewById(R.id.textview2);
      progressbar1 = (ProgressBar) findViewById(R.id.progressbar1);

      imageview1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {

         }
      });
   }
   // onCreate of this activity, we get the value of the barcode from the
scan code activity with key "c"
   private void initializeLogic() {
      if (true) {
         textview2.setText(getIntent().getStringExtra("c"));
         ToastUtil.showMessage(getApplicationContext(), "Scan complete. Now
searching");

         t = new TimerTask() {
            @Override
            public void run() {
               runOnUiThread(new Runnable() {
                  @Override
                  public void run() {
//                   After getting the barcode, we make a request to
openfoodfact to search for the barcode in the next activity
```

```
//                    i.e ViewMealRequestedActivity
                      i.putExtra("barcode",
"https://world.openfoodfacts.org/api/v2/product/".concat(textview2.getText(
).toString()).concat("/.json"));
                      i.putExtra("Type", "barcode");
                      i.setClass(getApplicationContext(),
ViewMealRequestedActivity.class);
                      startActivity(i);
                      finish();
                   }
                });
             }
          };
          timer.schedule(t, (int)(700));
       }
       else {

       }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
       super.onActivityResult(requestCode, resultCode, data);
       switch (requestCode) {

          default:
          break;
       }
    }

    @Override
    public void onBackPressed() {

    }
    @Deprecated
    public void showMessage(String msg) {
       Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
    }

    @Deprecated
    public int getLocationX(View v) {
       int location[] = new int[2];
       v.getLocationInWindow(location);
       return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
       int location[] = new int[2];
       v.getLocationInWindow(location);
       return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
       Random random = new Random();
       return random.nextInt(max - min + 1) + min;
    }
```

```
    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Search Meal Results Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.drawable.GradientDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
```

```
import com.bumptech.glide.Glide;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;


public class SearchmealresultsActivity extends AppCompatActivity {

   private HashMap<String, Object> header = new HashMap<>();
   private HashMap<String, Object> param = new HashMap<>();
   private String st = "";
   private HashMap<String, Object> foods = new HashMap<>();
   private String name = "";
   private String img = "";
   private String ingredients = "";
   private String kcal = "";
   private String carbo = "";
   private String protein = "";
   private String fat = "";
   private String sugar = "";
   private String salt = "";

   private ArrayList<HashMap<String, Object>> listmap = new ArrayList<>();
   private ArrayList<HashMap<String, Object>> listmap2 = new ArrayList<>();

   private LinearLayout linear2;
   private LinearLayout linrLoading;
   private ListView listview1;
   private LinearLayout linear4;
   private TextView textview2;
   private ProgressBar progressbar1;
   private TextView textview3;

   private RequestNetwork webapi;
   private RequestNetwork.RequestListener _webapi_request_listener;
   private Intent i = new Intent();
   private AlertDialog.Builder d1;
   private Intent res = new Intent();

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.searchmealresults);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();

      //GET request to connect the api to the app with tag a
      webapi.startRequestNetwork(RequestNetworkController.GET,
"https://world.openfoodfacts.org/cgi/search.pl?search_terms=".concat(getInt
ent().getStringExtra("text")).concat("&action=process&json=1"), "a",
_webapi_request_listener);
      textview2.setText("Search results for
".concat("'".concat(getIntent().getStringExtra("text").concat("'"))));

   }

   private void initialize(Bundle savedInstanceState) {
      linear2 =  findViewById(R.id.linear2);
```

```
      linrLoading =  findViewById(R.id.linrLoading);
      listview1 = (ListView) findViewById(R.id.listview1);
      linear4 =  findViewById(R.id.linear4);
      textview2 = (TextView) findViewById(R.id.textview2);
      progressbar1 = (ProgressBar) findViewById(R.id.progressbar1);
      textview3 = (TextView) findViewById(R.id.textview3);
      webapi = new RequestNetwork(this);
      d1 = new AlertDialog.Builder(this);

      //when the listView for the API meal requested results is clicked

      listview1.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
         @Override
         public void onItemClick(AdapterView<?> param1, View param2, int
param3, long param4) {
            final int position = param3;
            try{

               //Open the ViewMealRequestedActivity with the intent extra
and send the needed data
               i.putExtra("ingredients",
listmap.get((int)position).get("ingredients").toString());
               i.putExtra("img",
listmap.get((int)position).get("img").toString());
               i.putExtra("name",
listmap.get((int)position).get("name").toString());
               i.putExtra("carbo",
listmap.get((int)position).get("carbo").toString());
               i.putExtra("kcal",
listmap.get((int)position).get("kcal").toString());
               i.putExtra("fat",
listmap.get((int)position).get("fat").toString());
               i.putExtra("protein",
listmap.get((int)position).get("protein").toString());
               i.putExtra("sugar",
listmap.get((int)position).get("sugar").toString());
               i.putExtra("salt",
listmap.get((int)position).get("salt").toString());
               i.putExtra("Type", "all");
               i.setClass(getApplicationContext(),
ViewMealRequestedActivity.class);
               startActivity(i);
            }catch(Exception e){

            }
         }
      });

      //This webapi request listener works with RequestNetwork and
RequestNetworkController java files
      //to make request
      //to the site api.
      //onResponse needs some parameters such as tag and response strings
      _webapi_request_listener = new RequestNetwork.RequestListener() {
         @Override
         public void onResponse(String param1, String param2,
HashMap<String, Object> param3) {
            final String tag = param1;
            final String response = param2;
            final HashMap<String, Object> _responseHeaders = param3;
```

```
            linrLoading.setVisibility(View.GONE);
            if (tag.equals("a")) {
               try {

                  //Our response from openfoodfacts api begins with object
                  //Create an object org.json.JSONObject for the response
                  //from the json response, the detail needed are stored in
an array "products"
                  //Loop the array the to get individual objects in the
products array
                  //make sure you don't remove the block if(...).has(..)
because this checks if
                  // the object value we're requesting is made available
                  org.json.JSONObject obj = new
org.json.JSONObject(response);
                  org.json.JSONArray array = obj.getJSONArray("products");

                  //Loop products array to get the value in the respective
value to a listView or
                  // RecyclerView

                  for(int i=0;i<array.length();i++){

                     //Creates strings to save the value from the object
                     org.json.JSONObject object = array.getJSONObject(i);
                     if (object.has("product_name")) {
                        name = object.getString("product_name");
                         }
                     if (object.has("image_front_small_url")) {
                        img = object.getString("image_front_small_url");
                         }

                     if (object.has("ingredients_text")) {
                        ingredients = object.getString("ingredients_text");
                         }
                     if (object.getJSONObject("nutriments").has("energy-
kcal")) {
                        kcal =
object.getJSONObject("nutriments").getString("energy").toString(); }

                     if
(object.getJSONObject("nutriments").has("carbohydrates")) {
                        carbo =
object.getJSONObject("nutriments").getString("carbohydrates"); }
                     if
(object.getJSONObject("nutriments").has("proteins")) {
                        protein =
object.getJSONObject("nutriments").getString("proteins"); }
                     if (object.getJSONObject("nutriments").has("fat")) {
                        fat =
object.getJSONObject("nutriments").getString("fat"); }

                     if (object.getJSONObject("nutriments").has("sugar")) {
                        sugar =
object.getJSONObject("nutriments").getString("sugars"); }


                     if (object.getJSONObject("nutriments").has("salt")) {
                        salt =
object.getJSONObject("nutriments").getString("salt");
                         }
```

```
                    //Create HashMap (foods) to store the different
strings with appropriate keys
                    foods = new HashMap<>();
                    foods.put("name", name);
                    foods.put("img", img);
                    foods.put("ingredients", ingredients);
                    foods.put("kcal", kcal);
                    foods.put("carbo", carbo);
                    foods.put("protein", protein);
                    foods.put("fat", fat);
                    foods.put("sugar", sugar);
                    foods.put("salt", salt);

                    //Save all foods data to an ArrayList listmap
                    listmap.add(foods);
                }

                //setAdapter for the listView
                listview1.setAdapter(new Listview1Adapter(listmap));

((BaseAdapter)listview1.getAdapter()).notifyDataSetChanged();
            } catch(final org.json.JSONException ex) {

            }
        }
    }

    //onErrorResponse: Display a dialog to notify users
    @Override
    public void onErrorResponse(String param1, String param2) {
        final String tag = param1;
        final String message = param2;
        linrLoading.setVisibility(View.GONE);
        d1.setTitle("Failed !!");
        d1.setMessage("Failed To Connect To The Server . Please Try
Again.");
        d1.setPositiveButton("Retry", new
DialogInterface.OnClickListener() {

            //retry button : reloads the webapi
            @Override
            public void onClick(DialogInterface dialog, int which) {
                webapi.startRequestNetwork(RequestNetworkController.GET,
"https://us.openfoodfacts.org/.json", "a", _webapi_request_listener);
            }
        });
        d1.setNegativeButton("Exit", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                finish();
            }
        });
        d1.setCancelable(false);
        d1.create().show();
    }
  };
}

private void initializeLogic() {
```

```
      }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }


   //ListView Adapter
// List to show all meals from openfoodfact
// Customize or inflate the listview with the layout created for the
listview "mealsapi.xml"
   public class Listview1Adapter extends BaseAdapter {
      ArrayList<HashMap<String, Object>> data;
      public Listview1Adapter(ArrayList<HashMap<String, Object>> arr) {
         data = arr;
      }

      @Override
      public int getCount() {
         return data.size();
      }

      @Override
      public HashMap<String, Object> getItem(int index) {
         return data.get(index);
      }

      @Override
      public long getItemId(int index) {
         return index;
      }
      @Override
      public View getView(final int position, View v, ViewGroup container)
{
         LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
         View view = v;
         if (view == null) {
            view = inflater.inflate(R.layout.mealsapi, null);
         }

         //Creates the necessary elements i.e textview etc for the list
mealsapi layout
         final LinearLayout linear1 =  view.findViewById(R.id.linear1);
         final LinearLayout linear4 =  view.findViewById(R.id.linear4);
         final LinearLayout linear2 =  view.findViewById(R.id.linear2);
         final LinearLayout linear3 =  view.findViewById(R.id.linear3);
         final ImageView imgFoodImg =  view.findViewById(R.id.imgFoodImg);
         final TextView txtFoodName = (TextView)
view.findViewById(R.id.txtFoodName);
         final TextView textview2 = (TextView)
view.findViewById(R.id.textview2);
         final TextView carbo = (TextView) view.findViewById(R.id.carbo);
```

```
        final TextView kcal = (TextView) view.findViewById(R.id.kcal);
        final TextView ingrdt = (TextView) view.findViewById(R.id.ingrdt);
        final TextView fat = (TextView) view.findViewById(R.id.fat);
        final TextView protein = (TextView)
view.findViewById(R.id.protein);
        final TextView sugar = (TextView) view.findViewById(R.id.sugar);
        final TextView salt = (TextView) view.findViewById(R.id.salt);

        linear4.setVisibility(View.GONE);
        linear1.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b, int c, int d) {
           this.setCornerRadius(a);
           this.setStroke(b, c);
           this.setColor(d);
           return this;
        } }.getIns((int)10, (int)0, 0xFFAA00FF, 0xFFFFFFFF));

        //Get the data from the ArrayList listmap to display as a list

Glide.with(getApplicationContext()).load(Uri.parse(listmap.get((int)positio
n).get("img").toString())).into(imgFoodImg);

txtFoodName.setText(listmap.get((int)position).get("name").toString());
        carbo.setText(listmap.get((int)position).get("carbo").toString());
        kcal.setText(listmap.get((int)position).get("kcal").toString());
        fat.setText(listmap.get((int)position).get("fat").toString());

protein.setText(listmap.get((int)position).get("protein").toString());

ingrdt.setText(listmap.get((int)position).get("ingredients").toString());
        salt.setText(listmap.get((int)position).get("salt").toString());

        return view;
      }
   }

   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }

   @Deprecated
   public int getRandom(int min, int max) {
      Random random = new Random();
      return random.nextInt(max - min + 1) + min;
   }
```

```
    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Search Scan Meal Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.drawable.GradientDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;
```

```java
import androidx.appcompat.app.AppCompatActivity;

import com.bumptech.glide.Glide;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;


public class SearchscanmealActivity extends AppCompatActivity {

   private HashMap<String, Object> header = new HashMap<>();
   private HashMap<String, Object> param = new HashMap<>();
   private String st = "";
   private HashMap<String, Object> foods = new HashMap<>();
   private String name = "";
   private String img = "";
   private String ingredients = "";
   private String kcal = "";
   private String carbo = "";
   private String protein = "";
   private String fat = "";
   private String sugar = "";
   private String salt = "";

   private ArrayList<HashMap<String, Object>> listmap = new ArrayList<>();
   private ArrayList<HashMap<String, Object>> listmap2 = new ArrayList<>();

   private LinearLayout headsearch;
   private LinearLayout linrLoading;
   private ListView listview1;
   private LinearLayout linear4;
   private LinearLayout linear3;
   private TextView textview2;
   private ImageView imageview2;
   private EditText edittext2;
   private ImageView imageview1;
   private ProgressBar progressbar1;
   private TextView textview3;

   private RequestNetwork webapi;
   private RequestNetwork.RequestListener _webapi_request_listener;
   private Intent i = new Intent();
   private AlertDialog.Builder d1;
   private Intent res = new Intent();

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.searchscanmeal);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      headsearch =  findViewById(R.id.headsearch);
      linrLoading =  findViewById(R.id.linrLoading);
      listview1 = (ListView) findViewById(R.id.listview1);
      linear4 =  findViewById(R.id.linear4);
      linear3 =  findViewById(R.id.linear3);
```

```
      textview2 = (TextView) findViewById(R.id.textview2);
      imageview2 =  findViewById(R.id.imageview2);
      edittext2 = (EditText) findViewById(R.id.edittext2);
      imageview1 =  findViewById(R.id.imageview1);
      progressbar1 = (ProgressBar) findViewById(R.id.progressbar1);
      textview3 = (TextView) findViewById(R.id.textview3);
      webapi = new RequestNetwork(this);
      d1 = new AlertDialog.Builder(this);

      //when the listView for the API meal requested results is clicked

      listview1.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
         @Override
         public void onItemClick(AdapterView<?> param1, View param2, int
param3, long param4) {
            final int position = param3;
            try{

               //Open the ViewMealRequestedActivity with the intent extra
and send the needed data
               i.putExtra("ingredients",
listmap.get((int)position).get("ingredients").toString());
               i.putExtra("img",
listmap.get((int)position).get("img").toString());
               i.putExtra("name",
listmap.get((int)position).get("name").toString());
               i.putExtra("carbo",
listmap.get((int)position).get("carbo").toString());
               i.putExtra("kcal",
listmap.get((int)position).get("kcal").toString());
               i.putExtra("fat",
listmap.get((int)position).get("fat").toString());
               i.putExtra("protein",
listmap.get((int)position).get("protein").toString());
               i.putExtra("sugar",
listmap.get((int)position).get("sugar").toString());
               i.putExtra("salt",
listmap.get((int)position).get("salt").toString());
               i.putExtra("Type", "all");
               i.setClass(getApplicationContext(),
ViewMealRequestedActivity.class);
               startActivity(i);
            }catch(Exception e){

            }
         }
      });

      imageview2.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            i.setClass(getApplicationContext(), ScancodeActivity.class);
            startActivity(i);
         }
      });

      imageview1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            res.setClass(getApplicationContext(),
```

```
SearchmealresultsActivity.class);
            res.putExtra("text", edittext2.getText().toString());
            startActivity(res);
         }
      });



      //This webapi request listener works with RequestNetwork and
RequestNetworkController java files
      //to make request
      //to the site api.
      //onResponse needs some parameters such as tag and response strings
      _webapi_request_listener = new RequestNetwork.RequestListener() {
         @Override
         public void onResponse(String param1, String param2,
HashMap<String, Object> param3) {
            final String tag = param1;
            final String response = param2;
            final HashMap<String, Object> _responseHeaders = param3;
            linrLoading.setVisibility(View.GONE);
            if (tag.equals("a")) {
               try {


                  //Our response from openfoodfacts api begins with object
                  //Create an object org.json.JSONObject for the response
                  //from the json response, the detail needed are stored in
an array "products"
                  //Loop the array the to get individual objects in the
products array
                  //make sure you don't remove the block if(...).has(..)
because this checks if
                  // the object value we're requesting is made available
                  org.json.JSONObject obj = new
org.json.JSONObject(response);

                  //Loop products array to get the value in the respective
value to a listView or
                  // RecyclerView

                  org.json.JSONArray array = obj.getJSONArray("products");
                  for(int i=0;i<array.length();i++){ org.json.JSONObject
object = array.getJSONObject(i);


                     //Creates strings to save the value from the object
                     if (object.has("product_name")) {
                        name = object.getString("product_name");
                         }
                     if (object.has("image_front_small_url")) {
                        img = object.getString("image_front_small_url");
                         }

                     if (object.has("ingredients_text")) {
                        ingredients = object.getString("ingredients_text");
                         }
                     if (object.getJSONObject("nutriments").has("energy-
kcal")) {
                        kcal =
object.getJSONObject("nutriments").getString("energy").toString(); }
```

```
                    if
(object.getJSONObject("nutriments").has("carbohydrates")) {
                        carbo =
object.getJSONObject("nutriments").getString("carbohydrates"); }
                    if
(object.getJSONObject("nutriments").has("proteins")) {
                        protein =
object.getJSONObject("nutriments").getString("proteins"); }
                    if (object.getJSONObject("nutriments").has("fat")) {
                        fat =
object.getJSONObject("nutriments").getString("fat"); }

                    if (object.getJSONObject("nutriments").has("salt")) {
                        salt =
object.getJSONObject("nutriments").getString("salt");
                         }


                    //Create HashMap (foods) to store the different
strings with appropriate keys
                    foods = new HashMap<>();
                    foods.put("name", name);
                    foods.put("img", img);
                    foods.put("ingredients", ingredients);
                    foods.put("kcal", kcal);
                    foods.put("carbo", carbo);
                    foods.put("protein", protein);
                    foods.put("fat", fat);
                    foods.put("sugar", sugar);
                    foods.put("salt", salt);

                    //Save all foods data to an ArrayList listmap
                    listmap.add(foods);
                }


                //setAdapter for the listView
                listview1.setAdapter(new Listview1Adapter(listmap));

((BaseAdapter)listview1.getAdapter()).notifyDataSetChanged();
            } catch(final org.json.JSONException ex) {
                ToastUtil.showMessage(getApplicationContext(),
ex.toString());
            }
        }
    }

    @Override
    public void onErrorResponse(String param1, String param2) {
        final String tag = param1;
        final String message = param2;
        linrLoading.setVisibility(View.GONE);
        d1.setTitle("Failed !!");
        d1.setMessage("Failed To Connect To The Server . Please Try
Again.");
        d1.setPositiveButton("Retry", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
```

```
//                You may make request from us.openfood..... or
world.openfoo....
//                us.openfoo...support only US foods and in english
//                world.openfoo.... is global with any food name
                  webapi.startRequestNetwork(RequestNetworkController.GET,
"https://us.openfoodfacts.org/.json", "a", _webapi_request_listener);
               }
            });
            d1.setNegativeButton("Exit", new
DialogInterface.OnClickListener() {
               @Override
               public void onClick(DialogInterface dialog, int which) {
                  finish();
               }
            });
            d1.setCancelable(false);
            d1.create().show();
         }
      };
   }

   private void initializeLogic() {
      linear3.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)10, 0xFFFFFFFF));
      webapi.startRequestNetwork(RequestNetworkController.GET,
"https://world.openfoodfacts.org/.json", "a", _webapi_request_listener);
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   public class Listview1Adapter extends BaseAdapter {
      ArrayList<HashMap<String, Object>> data;
      public Listview1Adapter(ArrayList<HashMap<String, Object>> arr) {
         data = arr;
      }

      @Override
      public int getCount() {
         return data.size();
      }

      @Override
      public HashMap<String, Object> getItem(int index) {
         return data.get(index);
      }

      @Override
      public long getItemId(int index) {
         return index;
      }
      @Override
```

```
      public View getView(final int position, View v, ViewGroup container)
{
         LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
         View view = v;
         if (view == null) {
            view = inflater.inflate(R.layout.mealsapi, null);
         }

         final LinearLayout linear1 =  view.findViewById(R.id.linear1);
         final LinearLayout linear4 =  view.findViewById(R.id.linear4);
         final LinearLayout linear2 =  view.findViewById(R.id.linear2);
         final LinearLayout linear3 =  view.findViewById(R.id.linear3);
         final ImageView imgFoodImg =  view.findViewById(R.id.imgFoodImg);
         final TextView txtFoodName = (TextView)
view.findViewById(R.id.txtFoodName);
         final TextView textview2 = (TextView)
view.findViewById(R.id.textview2);
         final TextView carbo = (TextView) view.findViewById(R.id.carbo);
         final TextView kcal = (TextView) view.findViewById(R.id.kcal);
         final TextView ingrdt = (TextView) view.findViewById(R.id.ingrdt);
         final TextView fat = (TextView) view.findViewById(R.id.fat);
         final TextView protein = (TextView)
view.findViewById(R.id.protein);
         final TextView sugar = (TextView) view.findViewById(R.id.sugar);
         final TextView salt = (TextView) view.findViewById(R.id.salt);

         linear4.setVisibility(View.GONE);
         linear1.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b, int c, int d) {
this.setCornerRadius(a); this.setStroke(b, c); this.setColor(d); return
this; } }.getIns((int)10, (int)0, 0xFFAA00FF, 0xFFFFFFFF));

Glide.with(getApplicationContext()).load(Uri.parse(listmap.get((int)positio
n).get("img").toString())).into(imgFoodImg);

txtFoodName.setText(listmap.get((int)position).get("name").toString());
         carbo.setText(listmap.get((int)position).get("carbo").toString());
         fat.setText(listmap.get((int)position).get("fat").toString());

protein.setText(listmap.get((int)position).get("protein").toString());

ingrdt.setText(listmap.get((int)position).get("ingredients").toString());
         salt.setText(listmap.get((int)position).get("salt").toString());
         kcal.setText(listmap.get((int)position).get("kcal").toString());
         textview2.setText("Calories :
".concat(listmap.get((int)position).get("kcal").toString()));

         return view;
      }
   }

   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
```

```
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Splash Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.content.Intent;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
```

```
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;
import java.util.Random;
import java.util.Timer;
import java.util.TimerTask;


public class SplashActivity extends AppCompatActivity {
   private Timer timer = new Timer();

   private double num = 0;

   private LinearLayout linear1;
   private ImageView imageview1;
   private LinearLayout linear2;
   private ImageView imageview2;
   private TextView textview1;

   private TimerTask _timer;
   private Intent main = new Intent();

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.splash);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      linear1 =  findViewById(R.id.linear1);
      imageview1 =  findViewById(R.id.imageview1);
      linear2 =  findViewById(R.id.linear2);
      imageview2 =  findViewById(R.id.imageview2);
      textview1 = (TextView) findViewById(R.id.textview1);
   }

   private void initializeLogic() {
      num = ToastUtil.getRandom((int)(1), (int)(2));
      if (num == 1) {
         imageview1.setImageResource(R.drawable.img1_transcpr);
      }
      else {
         if (num == 2) {
            imageview1.setImageResource(R.drawable.img2_transcpr);
         }
      }
      _timer = new TimerTask() {
         @Override
         public void run() {
            runOnUiThread(new Runnable() {
               @Override
               public void run() {
```

```
                    main.setClass(getApplicationContext(),
MainActivity.class);
                    startActivity(main);
                    finish();
                }
            });
        }
    };
    timer.schedule(_timer, (int)(2000));
  }

  @Override
  protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode) {

      default:
      break;
    }
  }

  @Deprecated
  public void showMessage(String msg) {
    Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
  }

  @Deprecated
  public int getLocationX(View v) {
    int location[] = new int[2];
    v.getLocationInWindow(location);
    return location[0];
  }

  @Deprecated
  public int getLocationY(View v) {
    int location[] = new int[2];
    v.getLocationInWindow(location);
    return location[1];
  }

  @Deprecated
  public int getRandom(int min, int max) {
    Random random = new Random();
    return random.nextInt(max - min + 1) + min;
  }

  @Deprecated
  public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
    ArrayList<Double> result = new ArrayList<Double>();
    SparseBooleanArray arr = _list.getCheckedItemPositions();
    for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
      if (arr.valueAt(iIdx))
      result.add((double)arr.keyAt(iIdx));
    }
    return result;
  }

  @Deprecated
```

```
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Toast Util

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.annotation.SuppressLint;
import android.app.*;
import android.content.*;
import android.graphics.drawable.*;
import android.net.*;
import android.util.*;
import android.view.*;
import android.view.inputmethod.*;
import android.widget.*;

import java.io.*;
import java.util.*;

public class ToastUtil {

    public static int TOP = 1;
    public static int CENTER = 2;
    public static int BOTTOM = 3;

    public static void CustomToast(Context context, String message, int
textColor, int textSize, int bgColor, int radius, int gravity) {
        Toast toast = Toast.makeText(context, message, Toast.LENGTH_SHORT);
        View view = toast.getView();
        TextView textView = view.findViewById(android.R.id.message);
        textView.setTextSize(textSize);
        textView.setTextColor(textColor);
        textView.setGravity(Gravity.CENTER);

        GradientDrawable gradientDrawable = new GradientDrawable();
        gradientDrawable.setColor(bgColor);
        gradientDrawable.setCornerRadius(radius);
        view.setBackgroundDrawable(gradientDrawable);
        view.setPadding(15, 10, 15, 10);
        view.setElevation(10);
```

```
      switch (gravity) {
         case 1:
            toast.setGravity(Gravity.TOP, 0, 150);
            break;

         case 2:
            toast.setGravity(Gravity.CENTER, 0, 0);
            break;

         case 3:
            toast.setGravity(Gravity.BOTTOM, 0, 150);
            break;
      }
      toast.show();
   }

   public static void CustomToastWithIcon(Context context, String message,
int textColor, int textSize, int bgColor, int radius, int gravity, int
_icon) {
      Toast toast = Toast.makeText(context, message, Toast.LENGTH_SHORT);
      View view = toast.getView();
      TextView textView = (TextView)
view.findViewById(android.R.id.message);
      textView.setTextSize(textSize);
      textView.setTextColor(textColor);
      textView.setCompoundDrawablesWithIntrinsicBounds(_icon, 0, 0, 0);
      textView.setGravity(Gravity.CENTER);
      textView.setCompoundDrawablePadding(10);

      GradientDrawable gradientDrawable = new GradientDrawable();
      gradientDrawable.setColor(bgColor);
      gradientDrawable.setCornerRadius(radius);
      view.setBackgroundDrawable(gradientDrawable);
      view.setPadding(10, 10, 10, 10);
      view.setElevation(10);

      switch (gravity) {
         case 1:
            toast.setGravity(Gravity.TOP, 0, 150);
            break;

         case 2:
            toast.setGravity(Gravity.CENTER, 0, 0);
            break;

         case 3:
            toast.setGravity(Gravity.BOTTOM, 0, 150);
            break;
      }
      toast.show();
   }

   public static void sortListMap(final ArrayList<HashMap<String, Object>>
listMap, final String key, final boolean isNumber, final boolean ascending)
{
      Collections.sort(listMap, new Comparator<HashMap<String, Object>>() {
         public int compare(HashMap<String, Object> compareMap1,
HashMap<String, Object> compareMap2) {
            if (isNumber) {
               int count1 =
```

```
Integer.valueOf(compareMap1.get(key).toString());
                int count2 =
Integer.valueOf(compareMap2.get(key).toString());
                if (ascending) {
                   return count1 < count2 ? -1 : count1 < count2 ? 1 : 0;
                } else {
                   return count1 > count2 ? -1 : count1 > count2 ? 1 : 0;
                }
             } else {
                if (ascending) {
                   return
(compareMap1.get(key).toString()).compareTo(compareMap2.get(key).toString()
);
                } else {
                   return
(compareMap2.get(key).toString()).compareTo(compareMap1.get(key).toString()
);
                }
             }
          }
       });
    }

    public static void CropImage(Activity activity, String path, int
requestCode) {
       try {
          Intent intent = new Intent("com.android.camera.action.CROP");
          File file = new File(path);
          Uri _contentUri = Uri.fromFile(file);
          intent.setDataAndType(_contentUri, "image/*");
          intent.putExtra("crop", "true");
          intent.putExtra("aspectX", 1);
          intent.putExtra("aspectY", 1);
          intent.putExtra("outputX", 280);
          intent.putExtra("outputY", 280);
          intent.putExtra("return-data", false);
          activity.startActivityForResult(intent, requestCode);
       } catch (ActivityNotFoundException _e) {
          Toast.makeText(activity, "Your device doesn't support the crop
action!", Toast.LENGTH_SHORT).show();
       }
    }

    public static boolean isConnected(Context context) {
       ConnectivityManager connectivityManager = (ConnectivityManager)
context.getSystemService(Context.CONNECTIVITY_SERVICE);
       @SuppressLint("MissingPermission") NetworkInfo _activeNetworkInfo =
connectivityManager.getActiveNetworkInfo();
       return _activeNetworkInfo != null &&
_activeNetworkInfo.isConnected();
    }

    public static String copyFromInputStream(InputStream inputStream) {
       ByteArrayOutputStream _outputStream = new ByteArrayOutputStream();
       byte[] _buf = new byte[1024];
       int _i;
       try {
          while ((_i = inputStream.read(_buf)) != -1){
             _outputStream.write(_buf, 0, _i);
          }
          _outputStream.close();
```

```
            inputStream.close();
        } catch (IOException _e) {
        }

        return _outputStream.toString();
    }

    public static void hideKeyboard(Context context) {
        InputMethodManager inputMethodManager = (InputMethodManager)
context.getSystemService(Context.INPUT_METHOD_SERVICE);

inputMethodManager.toggleSoftInput(InputMethodManager.HIDE_IMPLICIT_ONLY,
0);
    }

    public static void showKeyboard(Context context) {
        InputMethodManager inputMethodManager = (InputMethodManager)
context.getSystemService(Context.INPUT_METHOD_SERVICE);
        inputMethodManager.toggleSoftInput(InputMethodManager.SHOW_FORCED,
0);
    }

    public static void showMessage(Context context, String msg) {
        Toast.makeText(context, msg, Toast.LENGTH_SHORT).show();
    }

    public static int getLocationX(View view) {
        int location[] = new int[2];
        view.getLocationInWindow(location);
        return location[0];
    }

    public static int getLocationY(View view) {
        int location[] = new int[2];
        view.getLocationInWindow(location);
        return location[1];
    }

    public static int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    public static ArrayList<Double> getCheckedItemPositionsToArray(ListView
_list) {
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
                result.add((double) arr.keyAt(iIdx));
        }
        return result;
    }

    public static float getDip(Context context, int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
context.getResources().getDisplayMetrics());
    }

    public static int getDisplayWidthPixels(Context context) {
        return context.getResources().getDisplayMetrics().widthPixels;
```

```
    }

    public static int getDisplayHeightPixels(Context context) {
        return context.getResources().getDisplayMetrics().heightPixels;
    }

    public static void getAllKeysFromMap(Map<String, Object> _map,
ArrayList<String> _output) {
        if (_output == null) return;
        _output.clear();
        if (_map == null || _map.size() < 1) return;
        Iterator _itr = _map.entrySet().iterator();
        while (_itr.hasNext()) {
            Map.Entry<String, String> _entry = (Map.Entry) _itr.next();
            _output.add(_entry.getKey());
        }
    }
}
```

Viewing Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import androidx.appcompat.app.AppCompatActivity;
import androidx.annotation.*;
import android.app.*;
import android.os.*;
import android.view.*;
import android.view.View.*;
import android.widget.*;
import android.content.*;
import android.content.res.*;
import android.graphics.*;
import android.graphics.drawable.*;
import android.media.*;
import android.net.*;
import android.text.*;
import android.text.style.*;
import android.util.*;
import android.webkit.*;
import android.animation.*;
import android.view.animation.*;
import java.util.*;
import java.util.regex.*;
import java.text.*;
import org.json.*;
import android.widget.LinearLayout;
import android.widget.ImageView;
import com.bumptech.glide.Glide;
import com.budiyev.android.codescanner.*;
import com.jjoe64.graphview.*;
import com.google.android.gms.*;
import com.blogspot.atifsoftwares.animatoolib.*;
import com.sdsmdg.tastytoast.*;
```

```
import com.google.zxing.*;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.DialogFragment;


public class ViewimgActivity extends AppCompatActivity {

   private LinearLayout linear1;
   private ImageView imageview1;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.viewimg);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      linear1 =  findViewById(R.id.linear1);
      imageview1 =  findViewById(R.id.imageview1);
   }

   private void initializeLogic() {

Glide.with(getApplicationContext()).load(Uri.parse(getIntent().getStringExt
ra("img"))).into(imageview1);
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }
```

```
    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

View Meal Activity

```
package com.vogella.android.diettrackerapplication;


// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import android.graphics.drawable.GradientDrawable;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
```

```
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import
com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;


public class ViewMealActivity extends AppCompatActivity {

   private FloatingActionButton fab;
   private String Title = "";
   private String Note = "";
   private HashMap<String, Object> Get_Note = new HashMap<>();
   private double Position = 0;
   private String fontName = "";
   private String typeace = "";
   private String Method = "";
   private String Ingredients = "";
   private String Calories = "";
   private String Hours = "";
   private String Minutes = "";
   private String classes = "";
   private String Serving = "";

   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();

   private LinearLayout toolbar;
   private ScrollView mainscroll;
   private ImageView back;
   private TextView title;
   private ImageView edit_img;
   private LinearLayout main;
   private LinearLayout linear4;
   private LinearLayout linear9;
   private LinearLayout linear1;
   private LinearLayout linear13;
   private TextView note;
   private LinearLayout linear16;
   private LinearLayout linear17;
   private LinearLayout linear5;
   private LinearLayout linear6;
   private TextView textview3;
   private TextView txtCookingMethod;
   private LinearLayout linear10;
   private LinearLayout linear11;
   private TextView textview5;
   private LinearLayout linear8;
```

```
    private EditText txtHours;
    private EditText txtMinutes;
    private LinearLayout linear2;
    private LinearLayout linear3;
    private TextView textview1;
    private TextView txtIngredients;
    private LinearLayout linear14;
    private LinearLayout linear15;
    private TextView textview7;
    private TextView txtCalories;
    private TextView textview8;
    private TextView txtClass;
    private TextView textview9;
    private TextView txtServing;

    private TextToSpeech NoteSpeak;
    private SharedPreferences Settings;
    private SharedPreferences AllMeals;
    private Intent Note_Edit = new Intent();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.view_meal);
        initialize(savedInstanceState);
        com.google.firebase.FirebaseApp.initializeApp(this);
        initializeLogic();
    }

    private void initialize(Bundle savedInstanceState) {
        fab = (FloatingActionButton) findViewById(R.id.fab);

        toolbar =  findViewById(R.id.toolbar);
        mainscroll =  findViewById(R.id.mainscroll);
        back =  findViewById(R.id.back);
        title = (TextView) findViewById(R.id.title);
        edit_img =  findViewById(R.id.edit_img);
        main =  findViewById(R.id.main);
        linear4 =  findViewById(R.id.linear4);
        linear9 =  findViewById(R.id.linear9);
        linear1 =  findViewById(R.id.linear1);
        linear13 =  findViewById(R.id.linear13);
        note = (TextView) findViewById(R.id.note);
        linear16 =  findViewById(R.id.linear16);
        linear17 =  findViewById(R.id.linear17);
        linear5 =  findViewById(R.id.linear5);
        linear6 =  findViewById(R.id.linear6);
        textview3 = (TextView) findViewById(R.id.textview3);
        txtCookingMethod = (TextView) findViewById(R.id.txtCookingMethod);
        linear10 =  findViewById(R.id.linear10);
        linear11 =  findViewById(R.id.linear11);
        textview5 = (TextView) findViewById(R.id.textview5);
        linear8 =  findViewById(R.id.linear8);
        txtHours = (EditText) findViewById(R.id.txtHours);
        txtMinutes = (EditText) findViewById(R.id.txtMinutes);
        linear2 =  findViewById(R.id.linear2);
        linear3 =  findViewById(R.id.linear3);
        textview1 = (TextView) findViewById(R.id.textview1);
        txtIngredients = (TextView) findViewById(R.id.txtIngredients);
        linear14 =  findViewById(R.id.linear14);
        linear15 =  findViewById(R.id.linear15);
```

```
      textview7 = (TextView) findViewById(R.id.textview7);
      txtCalories = (TextView) findViewById(R.id.txtCalories);
      textview8 = (TextView) findViewById(R.id.textview8);
      txtClass = (TextView) findViewById(R.id.txtClass);
      textview9 = (TextView) findViewById(R.id.textview9);
      txtServing = (TextView) findViewById(R.id.txtServing);
      NoteSpeak = new TextToSpeech(getApplicationContext(), null);
      Settings = getSharedPreferences("Settings", Activity.MODE_PRIVATE);
      AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);

      back.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            finish();
         }
      });

      edit_img.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            Note_Edit.setAction(Intent.ACTION_VIEW);
            Note_Edit.setClass(getApplicationContext(),
EditMealsActivity.class);
            Note_Edit.putExtra("Position",
getIntent().getStringExtra("Position"));
            startActivity(Note_Edit);
            finish();
         }
      });

      fab.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            if (NoteSpeak.isSpeaking()) {
               NoteSpeak.stop();
            }
            else {
               NoteSpeak.speak(txtCookingMethod.getText().toString(),
TextToSpeech.QUEUE_ADD, null);
            }
         }
      });
   }

   private void initializeLogic() {
      txtHours.setEnabled(false);
      txtMinutes.setEnabled(false);
      txtHours.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFFEEEEEE));
      txtMinutes.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFFEEEEEE));
      ONCREATE();
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {
```

```
            default:
            break;
         }
      }

      @Override
      public void onStart() {
         super.onStart();
         _GetNote();
      }
      public void Typeface () {

      }


      public void ONCREATE () {
         Position =
Double.parseDouble(getIntent().getStringExtra("Position"));
         Typeface();
         _GetNote();
         if (Settings.getString("Text Size", "").equals("Small")) {
            if (Title.length() > 35) {
               title.setText(Title.substring((int)(0),
(int)(35)).concat("..."));
            }
            else {
               title.setText(Title);
            }
         }
         else {
            if (Settings.getString("Text Size", "").equals("Medium")) {
               if (Title.length() > 32) {
                  title.setText(Title.substring((int)(0),
(int)(32)).concat("..."));
               }
               else {
                  title.setText(Title);
               }
            }
            else {
               if (Settings.getString("Text Size", "").equals("Large")) {
                  if (Title.length() > 29) {
                     title.setText(Title.substring((int)(0),
(int)(29)).concat("..."));
                  }
                  else {
                     title.setText(Title);
                  }
               }
               else {

               }
            }
         }
         note.setText(Note);
         txtCookingMethod.setText(Method);
         txtHours.setText(Hours.concat("hrs"));
         txtMinutes.setText(Minutes.concat("min"));
         txtIngredients.setText(Ingredients);
         txtCalories.setText(Calories);
```

```
      txtClass.setText(classes);
      getSettingsData();
   }


   public void Add (final String Colour, final ImageView Imageview) {
      Imageview.getDrawable().setColorFilter(Color.parseColor(Colour),
PorterDuff.Mode.SRC_IN);
   }


   public void getSettingsData () {
      if (Settings.getString("Detect Link", "").equals("True")) {
         _detectLinks(note);
      }
      else {

      }
      if (Settings.getString("Text Size", "").equals("Small")) {

      }
      else {
         if (Settings.getString("Text Size", "").equals("Medium")) {
            textSize(title, 18);
            textSize(note, 18);
         }
         else {
            if (Settings.getString("Text Size", "").equals("Large")) {
               textSize(title, 20);
               textSize(note, 20);
            }
            else {

            }
         }
      }
      if (Settings.getString("Theme", "").equals("Default")) {
         Window w =
ViewMealActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.F
LAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SY
STEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#FFFFFF"));
main.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
         Add("#3770FD", back);
         Add("#3770FD", edit_img);
      }
      else {
         if (Settings.getString("Theme", "").equals("Dark")) {
            Window w =
ViewMealActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.F
LAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SY
STEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#263238"));
            toolbar.setBackgroundColor(0xFF263238);
            mainscroll.setBackgroundColor(0xFF263238);
            title.setTextColor(0xFFFFFFFF);
            note.setTextColor(0xFFFFFFFF);

fab.setBackgroundTintList(android.content.res.ColorStateList.valueOf(Color.
parseColor("#263238")));
            Add("#FFFFFF", back);
            Add("#FFFFFF", edit_img);
         }
```

```
         else {
            if (Settings.getString("Theme", "").equals("Blue Grey")) {
               toolbar.setBackgroundColor(0xFFFFFFFF);
               mainscroll.setBackgroundColor(0xFFFFFFFF);
               title.setTextColor(0xFF607D8B);
               note.setTextColor(0xFF000000);
               Window w =
ViewMealActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.F
LAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SY
STEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#FFFFFF"));
main.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);

fab.setBackgroundTintList(android.content.res.ColorStateList.valueOf(Color.
parseColor("#607D8B")));
               Add("#607D8B", back);
               Add("#607D8B", edit_img);
            }
            else {
               if (Settings.getString("Theme", "").equals("Orange")) {
                  toolbar.setBackgroundColor(0xFFFFFFFF);
                  mainscroll.setBackgroundColor(0xFFFFFFFF);
                  title.setTextColor(0xFFFF5722);
                  note.setTextColor(0xFF000000);
                  Window w =
ViewMealActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.F
LAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SY
STEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#FFFFFF"));
main.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);

fab.setBackgroundTintList(android.content.res.ColorStateList.valueOf(Color.
parseColor("#FF5722")));
                  Add("#FF5722", back);
                  Add("#FF5722", edit_img);
               }
               else {
                  if (Settings.getString("Theme", "").equals("Indigo")) {
                     toolbar.setBackgroundColor(0xFFFFFFFF);
                     mainscroll.setBackgroundColor(0xFFFFFFFF);
                     title.setTextColor(0xFF3F51B5);
                     note.setTextColor(0xFF000000);
                     Window w =
ViewMealActivity.this.getWindow();w.clearFlags(WindowManager.LayoutParams.F
LAG_TRANSLUCENT_STATUS);w.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SY
STEM_BAR_BACKGROUNDS); w.setStatusBarColor(Color.parseColor("#FFFFFF"));
main.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);

fab.setBackgroundTintList(android.content.res.ColorStateList.valueOf(Color.
parseColor("#3F51B5")));
                     Add("#3F51B5", back);
                     Add("#3F51B5", edit_img);
                  }
                  else {

                  }
               }
            }
         }
      }
   }
```

```
   public void _detectLinks (final TextView _txt_linkify) {
      _txt_linkify.setClickable(true);
      android.text.util.Linkify.addLinks(_txt_linkify,
android.text.util.Linkify.ALL);
      _txt_linkify.setLinkTextColor(Color.parseColor("#FF3770FD"));
      _txt_linkify.setLinksClickable(true);
   }


   public void textSize (final TextView TextView1, final double size) {
      int j = (int) size;
      TextView1.setTextSize(j);
   }


   public void _GetNote () {
      if (!AllMeals.getString("Meals", "").equals("")) {
         Notes_List = new Gson().fromJson(AllMeals.getString("Meals", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
         Get_Note = Notes_List.get((int)Position);
         Title = Get_Note.get("Title").toString();
         Note = Get_Note.get("Note").toString();
         Method = Get_Note.get("Method").toString();
         Ingredients = Get_Note.get("Ingredients").toString();
         Calories = Get_Note.get("Calories").toString();
         Hours = Get_Note.get("Hours").toString();
         Minutes = Get_Note.get("Minutes").toString();
         classes = Get_Note.get("Class").toString();
         Serving = Get_Note.get("Serving").toString();
      }
   }


   public void changeActivityFont (final String fontname) {
      fontName = "fonts/".concat(fontname.concat(".ttf"));
      overrideFonts(this,getWindow().getDecorView());
   }
   private void overrideFonts(final android.content.Context context, final
View v) {

      try {
         Typeface
         typeace = Typeface.createFromAsset(getAssets(), fontName);;
         if ((v instanceof ViewGroup)) {
            ViewGroup vg = (ViewGroup) v;
            for (int i = 0;
            i < vg.getChildCount();
            i++) {
               View child = vg.getChildAt(i);
               overrideFonts(context, child);
            }
         }
         else {
            if ((v instanceof TextView)) {
               ((TextView) v).setTypeface(typeace);
            }
            else {
               if ((v instanceof EditText )) {
                  ((EditText) v).setTypeface(typeace);
               }
               else {
```

```
                    if ((v instanceof Button)) {
                        ((Button) v).setTypeface(typeace);
                    }
                }
            }
        }
        catch(Exception e)

        {

        };
    }


    @Deprecated
    public void showMessage(String msg) {
        Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
    }

    @Deprecated
    public int getLocationX(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
```

```
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

View Meal Requested Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.drawable.GradientDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.View;
import android.view.Window;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.bumptech.glide.Glide;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import com.sdsmdg.tastytoast.TastyToast;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Random;

import de.hdodenhof.circleimageview.CircleImageView;


public class ViewMealRequestedActivity extends AppCompatActivity {

    private HashMap<String, Object> AddFood = new HashMap<>();
```

```
   private String Date = "";
   private String Time = "";
   private HashMap<String, Object> AddRecipe = new HashMap<>();
   private String img = "";
   private String ingredients = "";
   private String kcal = "";
   private String carbo = "";
   private String protein = "";
   private String fat = "";
   private String sugar = "";
   private String salt = "";
   private String name = "";

   private ArrayList<HashMap<String, Object>> Food_List = new
ArrayList<>();
   private ArrayList<HashMap<String, Object>> Recipe_List = new
ArrayList<>();

   private LinearLayout toolbar;
   private LinearLayout linear4;
   private ScrollView vscroll1;
   private Button button1;
   private ImageView back;
   private LinearLayout linear1;
   private LinearLayout linrMain;
   private TextView textview7;
   private LinearLayout linrKcal;
   private TextView textview4;
   private LinearLayout linrCarbo;
   private LinearLayout linrProtn;
   private LinearLayout linrFats;
   private LinearLayout linrSugar;
   private LinearLayout linrSa;
   private LinearLayout linear5;
   private TextView txtIngredient;
   private CircleImageView circleimageview1;
   private LinearLayout linear3;
   private TextView txtFoodName;
   private TextView textview8;
   private TextView txtkCal;
   private TextView textview5;
   private TextView txtCarbohydate;
   private TextView textview10;
   private TextView txtProtein;
   private TextView textview12;
   private TextView txtFat;
   private TextView textview14;
   private TextView txtSugar;
   private TextView textview16;
   private TextView txtSalt;
   private TextView textview17;
   private LinearLayout linear6;
   private TextView textview18;
   private ImageView imageview1;

   private SharedPreferences Diary;
   private Calendar Cal = Calendar.getInstance();
   private SharedPreferences Recipe;
   private RequestNetwork barc;
   private RequestNetwork.RequestListener _barc_request_listener;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.viewmealrequested);
        initialize(savedInstanceState);
        com.google.firebase.FirebaseApp.initializeApp(this);
        initializeLogic();
    }

    private void initialize(Bundle savedInstanceState) {
        toolbar =  findViewById(R.id.toolbar);
        linear4 =  findViewById(R.id.linear4);
        vscroll1 =  findViewById(R.id.vscroll1);
        button1 = (Button) findViewById(R.id.button1);
        back =  findViewById(R.id.back);
        linear1 =  findViewById(R.id.linear1);
        linrMain =  findViewById(R.id.linrMain);
        textview7 = (TextView) findViewById(R.id.textview7);
        linrKcal =  findViewById(R.id.linrKcal);
        textview4 = (TextView) findViewById(R.id.textview4);
        linrCarbo =  findViewById(R.id.linrCarbo);
        linrProtn =  findViewById(R.id.linrProtn);
        linrFats =  findViewById(R.id.linrFats);
        linrSugar =  findViewById(R.id.linrSugar);
        linrSa =  findViewById(R.id.linrSa);
        linear5 =  findViewById(R.id.linear5);
        txtIngredient = (TextView) findViewById(R.id.txtIngredient);
        circleimageview1 = (CircleImageView)
findViewById(R.id.circleimageview1);
        linear3 =  findViewById(R.id.linear3);
        txtFoodName = (TextView) findViewById(R.id.txtFoodName);
        textview8 = (TextView) findViewById(R.id.textview8);
        txtkCal = (TextView) findViewById(R.id.txtkCal);
        textview5 = (TextView) findViewById(R.id.textview5);
        txtCarbohydate = (TextView) findViewById(R.id.txtCarbohydate);
        textview10 = (TextView) findViewById(R.id.textview10);
        txtProtein = (TextView) findViewById(R.id.txtProtein);
        textview12 = (TextView) findViewById(R.id.textview12);
        txtFat = (TextView) findViewById(R.id.txtFat);
        textview14 = (TextView) findViewById(R.id.textview14);
        txtSugar = (TextView) findViewById(R.id.txtSugar);
        textview16 = (TextView) findViewById(R.id.textview16);
        txtSalt = (TextView) findViewById(R.id.txtSalt);
        textview17 = (TextView) findViewById(R.id.textview17);
        linear6 =  findViewById(R.id.linear6);
        textview18 = (TextView) findViewById(R.id.textview18);
        imageview1 =  findViewById(R.id.imageview1);
        Diary = getSharedPreferences("Diary", Activity.MODE_PRIVATE);
        Recipe = getSharedPreferences("Recipe", Activity.MODE_PRIVATE);
        barc = new RequestNetwork(this);

        button1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                AddFood = new HashMap<>();
                AddFood.put("Food", txtFoodName.getText().toString());
                AddFood.put("Ingredt", txtIngredient.getText().toString());
                AddFood.put("kCal", txtkCal.getText().toString());
                AddFood.put("Carbo", txtCarbohydate.getText().toString());
                AddFood.put("Protein", txtProtein.getText().toString());
                AddFood.put("Fat", txtFat.getText().toString());
```

```
            AddFood.put("Salt", txtSalt.getText().toString());
            AddFood.put("Date", Date);
            AddFood.put("Time", Time);
            AddFood.put("Img",
getIntent().getStringExtra("image_frontmsgmall_url"));
            AddFood.put("Consumed", "False");
            Food_List.add(AddFood);
            Diary.edit().putString("Diary", new
Gson().toJson(Food_List)).commit();
            TastyToast.makeText(getApplicationContext(), "Your Meal Have
Been Successfully Added To Food Diary", TastyToast.LENGTH_LONG,
TastyToast.SUCCESS);
         }
      });

      linear6.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            AddRecipe = new HashMap<>();
            AddRecipe.put("Food", txtFoodName.getText().toString());
            AddRecipe.put("Ingredt", txtIngredient.getText().toString());
            AddRecipe.put("Date", Date);
            AddRecipe.put("Time", Time);
            Recipe_List.add(AddRecipe);
            Recipe.edit().putString("Recipe", new
Gson().toJson(Recipe_List)).commit();
            TastyToast.makeText(getApplicationContext(), "Recipe Have Been
Successfully Added", TastyToast.LENGTH_LONG, TastyToast.SUCCESS);
         }
      });

      _barc_request_listener = new RequestNetwork.RequestListener() {
         @Override
         public void onResponse(String param1, String param2,
HashMap<String, Object> param3) {
            final String tag = param1;
            final String response = param2;
            final HashMap<String, Object> _responseHeaders = param3;
            if (tag.equals("a")) {
               try {
                  org.json.JSONObject obj = new
org.json.JSONObject(response);
                  org.json.JSONObject product =
obj.getJSONObject("product");
                  if (product.has("product_name")) {
                     name = product.getString("product_name");
                      }
                  if (product.has("image_frontmsgmall_url")) {
                     img = product.getString("image_frontmsgmall_url");
                      }

                  if (product.has("ingredients_text")) {
                     ingredients = product.getString("ingredients_text");
                      }
                  if (product.getJSONObject("nutriments").has("energy-
kcal")) {
                     kcal =
product.getJSONObject("nutriments").getString("energy-kcal").toString(); }

                  if
(product.getJSONObject("nutriments").has("carbohydrates")) {
```

```
                    carbo =
product.getJSONObject("nutriments").getString("carbohydrates"); }
                if (product.getJSONObject("nutriments").has("proteins"))
{
                    protein =
product.getJSONObject("nutriments").getString("proteins"); }
                if (product.getJSONObject("nutriments").has("fat")) {
                    fat =
product.getJSONObject("nutriments").getString("fat"); }


                if (product.getJSONObject("nutriments").has("salt")) {
                    salt =
product.getJSONObject("nutriments").getString("salt");
                     }
                txtFoodName.setText(name);
                txtIngredient.setText(ingredients);
                txtkCal.setText(kcal);
                txtCarbohydate.setText(carbo);
                txtProtein.setText(protein);
                txtFat.setText(fat);
                txtSalt.setText(salt);
                _CoreProgressLoading(false);
             } catch(final org.json.JSONException ex) {
                ToastUtil.showMessage(getApplicationContext(),
ex.toString());
                _CoreProgressLoading(false);
             }
          }
       }

       @Override
       public void onErrorResponse(String param1, String param2) {
          final String tag = param1;
          final String message = param2;
          ToastUtil.showMessage(getApplicationContext(), message);
       }
    };
 }

 private void initializeLogic() {

//    onCreate
    try{
       Date = new SimpleDateFormat("dd MMM").format(Cal.getTime());
       Time = new SimpleDateFormat("hh:mm a").format(Cal.getTime());
       button1.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) { this.setCornerRadius(a);
this.setColor(b); return this; } }.getIns((int)15, 0xFF3F51B6));
       if (!Diary.getString("Diary", "").equals("")) {
          Food_List = new Gson().fromJson(Diary.getString("Diary", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
       }
       if (!Recipe.getString("Recipe", "").equals("")) {
          Recipe_List = new Gson().fromJson(Recipe.getString("Recipe",
""), new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
       }
       if (getIntent().getStringExtra("Type").equals("all")) {
          txtFoodName.setText(getIntent().getStringExtra("name"));

Glide.with(getApplicationContext()).load(Uri.parse(getIntent().getStringExt
```

```
ra("img"))).into(circleimageview1);

txtIngredient.setText(getIntent().getStringExtra("ingredients"));
            txtkCal.setText(getIntent().getStringExtra("kcal"));
            txtCarbohydate.setText(getIntent().getStringExtra("carbo"));
            txtProtein.setText(getIntent().getStringExtra("protein"));
            txtFat.setText(getIntent().getStringExtra("fat"));
            txtSugar.setText(getIntent().getStringExtra("sugar"));
            txtSalt.setText(getIntent().getStringExtra("salt"));
         }
         else {
            if (getIntent().getStringExtra("Type").equals("barcode")) {
               barc.startRequestNetwork(RequestNetworkController.GET,
getIntent().getStringExtra("barcode"), "a", _barc_request_listener);
            }
         }
      }catch(Exception e){

      }
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   public void _cal () {

   }


   public void _CoreProgressLoading (final boolean _ifShow) {
      if (_ifShow) {
         if (coreprog == null){
            coreprog = new ProgressDialog(this);
            coreprog.setCancelable(false);
            coreprog.setCanceledOnTouchOutside(false);

            coreprog.requestWindowFeature(Window.FEATURE_NO_TITLE);
coreprog.getWindow().setBackgroundDrawable(new
android.graphics.drawable.ColorDrawable(Color.TRANSPARENT));

         }

         android.graphics.drawable.GradientDrawable gads = new
android.graphics.drawable.GradientDrawable();
         gads.setColor(Color.parseColor("#FFFFFF"));
         gads.setCornerRadius(100);
         coreprog.getWindow().setBackgroundDrawable(gads);
         coreprog.setMessage(null);
         coreprog.show();
         coreprog.setContentView(R.layout.custom_dialog);
      }
      else {
         if (coreprog != null){
```

```
            coreprog.dismiss();
         }
      }
   }
   private ProgressDialog coreprog;
   {
   }


   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }

   @Deprecated
   public int getRandom(int min, int max) {
      Random random = new Random();
      return random.nextInt(max - min + 1) + min;
   }

   @Deprecated
   public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
      ArrayList<Double> result = new ArrayList<Double>();
      SparseBooleanArray arr = _list.getCheckedItemPositions();
      for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
         if (arr.valueAt(iIdx))
         result.add((double)arr.keyAt(iIdx));
      }
      return result;
   }

   @Deprecated
   public float getDip(int input) {
      return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
   }

   @Deprecated
   public int getDisplayWidthPixels() {
      return getResources().getDisplayMetrics().widthPixels;
   }

   @Deprecated
   public int getDisplayHeightPixels() {
      return getResources().getDisplayMetrics().heightPixels;
```

```
    }
}
```

View Recipe Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import androidx.appcompat.app.AppCompatActivity;
import androidx.annotation.*;
import android.app.*;
import android.os.*;
import android.view.*;
import android.view.View.*;
import android.widget.*;
import android.content.*;
import android.content.res.*;
import android.graphics.*;
import android.graphics.drawable.*;
import android.media.*;
import android.net.*;
import android.text.*;
import android.text.style.*;
import android.util.*;
import android.webkit.*;
import android.animation.*;
import android.view.animation.*;
import java.util.*;
import java.util.regex.*;
import java.text.*;
import org.json.*;
import java.util.HashMap;
import java.util.ArrayList;
import android.widget.LinearLayout;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.ScrollView;
import android.widget.ListView;
import android.widget.ArrayAdapter;
import android.widget.BaseAdapter;
import de.hdodenhof.circleimageview.*;
import android.app.Activity;
import android.content.SharedPreferences;
import android.content.Intent;
import android.net.Uri;
import android.view.View;
import com.google.gson.Gson;
import com.bumptech.glide.Glide;
import com.google.gson.reflect.TypeToken;
import com.budiyev.android.codescanner.*;
import com.jjoe64.graphview.*;
import com.google.android.gms.*;
import com.blogspot.atifsoftwares.animatoolib.*;
import com.sdsmdg.tastytoast.*;
import com.google.zxing.*;
```

```
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.DialogFragment;


public class ViewrecipeActivity extends AppCompatActivity {

   private HashMap<String, Object> ingred = new HashMap<>();
   private double position = 0;
   private HashMap<String, Object> AddRecipe = new HashMap<>();

   private ArrayList<HashMap<String, Object>> recipe = new ArrayList<>();
   private ArrayList<HashMap<String, Object>> recip = new ArrayList<>();
   private ArrayList<HashMap<String, Object>> Recipe_List = new
ArrayList<>();

   private LinearLayout toolbar;
   private LinearLayout linear1;
   private ImageView back_img;
   private LinearLayout linear8;
   private LinearLayout linear7;
   private TextView textview18;
   private LinearLayout linear9;
   private ScrollView vscroll1;
   private LinearLayout linear10;
   private ListView listview1;
   private TextView textview19;
   private ImageView imageview1;
   private LinearLayout linear2;
   private LinearLayout linear4;
   private LinearLayout linear11;
   private TextView txtRecipe;
   private TextView textview5;
   private LinearLayout linrCarbo;
   private LinearLayout linrProtn;
   private LinearLayout linrFats;
   private LinearLayout linrSugar;
   private LinearLayout linrSa;
   private LinearLayout linear6;
   private CircleImageView circleimageview1;
   private LinearLayout linear5;
   private TextView txtMeal;
   private LinearLayout linear3;
   private TextView txtCalories;
   private TextView textview3;
   private ImageView imageview2;
   private TextView textview6;
   private TextView txtCarbohydate;
   private TextView textview10;
   private TextView txtProtein;
   private TextView textview12;
   private TextView txtFat;
   private TextView textview14;
   private TextView txtSugar;
   private TextView textview16;
   private TextView txtFiber;
   private TextView textview17;
   private TextView txtCholestrols;

   private SharedPreferences holdrec;
   private Intent img = new Intent();
```

```
    private SharedPreferences Recipe;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.viewrecipe);
        initialize(savedInstanceState);
        com.google.firebase.FirebaseApp.initializeApp(this);
        initializeLogic();
    }

    private void initialize(Bundle savedInstanceState) {
        toolbar =  findViewById(R.id.toolbar);
        linear1 =  findViewById(R.id.linear1);
        back_img =  findViewById(R.id.back_img);
        linear8 =  findViewById(R.id.linear8);
        linear7 =  findViewById(R.id.linear7);
        textview18 = (TextView) findViewById(R.id.textview18);
        linear9 =  findViewById(R.id.linear9);
        vscroll1 =  findViewById(R.id.vscroll1);
        linear10 =  findViewById(R.id.linear10);
        listview1 = (ListView) findViewById(R.id.listview1);
        textview19 = (TextView) findViewById(R.id.textview19);
        imageview1 =  findViewById(R.id.imageview1);
        linear2 =  findViewById(R.id.linear2);
        linear4 =  findViewById(R.id.linear4);
        linear11 =  findViewById(R.id.linear11);
        txtRecipe = (TextView) findViewById(R.id.txtRecipe);
        textview5 = (TextView) findViewById(R.id.textview5);
        linrCarbo =  findViewById(R.id.linrCarbo);
        linrProtn =  findViewById(R.id.linrProtn);
        linrFats =  findViewById(R.id.linrFats);
        linrSugar =  findViewById(R.id.linrSugar);
        linrSa =  findViewById(R.id.linrSa);
        linear6 =  findViewById(R.id.linear6);
        circleimageview1 = (CircleImageView)
findViewById(R.id.circleimageview1);
        linear5 =  findViewById(R.id.linear5);
        txtMeal = (TextView) findViewById(R.id.txtMeal);
        linear3 =  findViewById(R.id.linear3);
        txtCalories = (TextView) findViewById(R.id.txtCalories);
        textview3 = (TextView) findViewById(R.id.textview3);
        imageview2 =  findViewById(R.id.imageview2);
        textview6 = (TextView) findViewById(R.id.textview6);
        txtCarbohydate = (TextView) findViewById(R.id.txtCarbohydate);
        textview10 = (TextView) findViewById(R.id.textview10);
        txtProtein = (TextView) findViewById(R.id.txtProtein);
        textview12 = (TextView) findViewById(R.id.textview12);
        txtFat = (TextView) findViewById(R.id.txtFat);
        textview14 = (TextView) findViewById(R.id.textview14);
        txtSugar = (TextView) findViewById(R.id.txtSugar);
        textview16 = (TextView) findViewById(R.id.textview16);
        txtFiber = (TextView) findViewById(R.id.txtFiber);
        textview17 = (TextView) findViewById(R.id.textview17);
        txtCholestrols = (TextView) findViewById(R.id.txtCholestrols);
        holdrec = getSharedPreferences("holdrec", Activity.MODE_PRIVATE);
        Recipe = getSharedPreferences("Recipe", Activity.MODE_PRIVATE);

        linear7.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
```

```
            AddRecipe = new HashMap<>();
            AddRecipe.put("Food", txtMeal.getText().toString());
            AddRecipe.put("Carbo", txtCarbohydate.getText().toString());
            AddRecipe.put("Protein", txtProtein.getText().toString());
            AddRecipe.put("Fat", txtFat.getText().toString());
            AddRecipe.put("Sugar", txtSugar.getText().toString());
            AddRecipe.put("Fiber", txtFiber.getText().toString());
            AddRecipe.put("Cholestrol",
txtCholestrols.getText().toString());
            AddRecipe.put("Recipe", txtRecipe.getText().toString());
            Recipe_List.add(AddRecipe);
            Recipe.edit().putString("Recipe", new
Gson().toJson(Recipe_List)).commit();
            TastyToast.makeText(getApplicationContext(), "Recipe Have Been
Successfully Added", TastyToast.LENGTH_LONG, TastyToast.SUCCESS);
            linear7.setAlpha((float)(0.5d));
            linear7.setEnabled(false);
         }
      });

      imageview1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            vscroll1.setVisibility(View.VISIBLE);
            linear9.setVisibility(View.GONE);
         }
      });

      circleimageview1.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            img.setClass(getApplicationContext(), ViewimgActivity.class);
            img.putExtra("img", getIntent().getStringExtra("img"));
            startActivity(img);
         }
      });

      textview3.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            linear9.setVisibility(View.VISIBLE);
            vscroll1.setVisibility(View.GONE);
         }
      });

      imageview2.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            linear9.setVisibility(View.VISIBLE);
            vscroll1.setVisibility(View.GONE);
         }
      });
   }

   private void initializeLogic() {
      linear9.setVisibility(View.GONE);
      try{

Glide.with(getApplicationContext()).load(Uri.parse(getIntent().getStringExt
ra("img"))).into(circleimageview1);
         txtCalories.setText("Calories:
```

```
".concat(getIntent().getStringExtra("calories").substring((int)(0),
(int)(4))));
         txtMeal.setText(getIntent().getStringExtra("name"));

txtFat.setText(getIntent().getStringExtra("fat").substring((int)(0),
(int)(4)).concat(""));

txtSugar.setText(getIntent().getStringExtra("sugar").substring((int)(0),
(int)(4)).concat(""));

txtFiber.setText(getIntent().getStringExtra("fiber").substring((int)(0),
(int)(4)).concat(""));

txtCholestrols.setText(getIntent().getStringExtra("cholesterol").substring(
(int)(0), (int)(4)).concat(""));

txtCarbohydate.setText(getIntent().getStringExtra("carbo").substring((int)(
0), (int)(4)).concat(""));

txtProtein.setText(getIntent().getStringExtra("protein").substring((int)(0)
, (int)(4)).concat(""));
         position =
Double.parseDouble(getIntent().getStringExtra("position"));

//       To get the recipe for the particular viewed meal, since our
recipe(ingredients) is an ArrayList
//       we'll get the value of the recipe into a listview too.
//       The recipe data was saved into a Json and for that we'll be
calling the
//       recipe from the extra
         if (!getIntent().getStringExtra("ingredients").equals("")) {
            recipe = new
Gson().fromJson(getIntent().getStringExtra("ingredients"),
                  new TypeToken<ArrayList<HashMap<String,
Object>>>(){}.getType());
            listview1.setAdapter(new Listview1Adapter(recipe));
            ((BaseAdapter)listview1.getAdapter()).notifyDataSetChanged();
         }
         for(int i = 0; i < (int)(recipe.size()); i++) {
            txtRecipe.setText(getIntent().getStringExtra("text"));
         }
      }catch(Exception e){

      }
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   public class Listview1Adapter extends BaseAdapter {
      ArrayList<HashMap<String, Object>> data;
      public Listview1Adapter(ArrayList<HashMap<String, Object>> arr) {
         data = arr;
```

```
      }

      @Override
      public int getCount() {
         return data.size();
      }

      @Override
      public HashMap<String, Object> getItem(int index) {
         return data.get(index);
      }

      @Override
      public long getItemId(int index) {
         return index;
      }
      @Override
      public View getView(final int position, View v, ViewGroup container)
{
         LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
         View view = v;
         if (view == null) {
            view = inflater.inflate(R.layout.rec, null);
         }

         final TextView textview1 = (TextView)
view.findViewById(R.id.textview1);


textview1.setText(recipe.get((int)position).get("ingredients").toString());

         return view;
      }
   }

   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }

   @Deprecated
   public int getRandom(int min, int max) {
      Random random = new Random();
      return random.nextInt(max - min + 1) + min;
```

```
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

View Weight Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.animation.ObjectAnimator;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.ClipData;
import android.content.ClipboardManager;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.os.Build;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.MotionEvent;
import android.view.View;
import android.view.animation.AccelerateDecelerateInterpolator;
import android.widget.ImageView;
import android.widget.LinearLayout;
```

```
import android.widget.ListView;
import android.widget.ScrollView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import
com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;
import java.util.Timer;
import java.util.TimerTask;


public class ViewWeightActivity extends AppCompatActivity {
   private Timer timer = new Timer();

   private FloatingActionButton fab;
   private String colorPrimary = "";
   private String position = "";
   private String ID = "";
   private double n = 0;
   private boolean isDark = false;

   private ArrayList<HashMap<String, Object>> list = new ArrayList<>();

   private LinearLayout main;
   private LinearLayout fragment1;
   private LinearLayout action;
   private LinearLayout linear;
   private ImageView close;
   private TextView top_title;
   private ImageView delete;
   private ImageView copy;
   private ScrollView vscroll1;
   private LinearLayout linear1;
   private LinearLayout linear2;
   private TextView text;
   private ImageView imageview1;
   private TextView title;

   private Intent intent = new Intent();
   private SharedPreferences data;
   private AlertDialog.Builder deleteDialog;
   private TimerTask _timer;
   private SharedPreferences week;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.viewweight);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }
```

```
   private void initialize(Bundle savedInstanceState) {
      fab = (FloatingActionButton) findViewById(R.id.fab);

      main =  findViewById(R.id.main);
      fragment1 =  findViewById(R.id.fragment1);
      action =  findViewById(R.id.action);
      linear =  findViewById(R.id.linear);
      close =  findViewById(R.id.close);
      top_title = (TextView) findViewById(R.id.top_title);
      delete =  findViewById(R.id.delete);
      copy =  findViewById(R.id.copy);
      vscroll1 =  findViewById(R.id.vscroll1);
      linear1 =  findViewById(R.id.linear1);
      linear2 =  findViewById(R.id.linear2);
      text = (TextView) findViewById(R.id.text);
      imageview1 =  findViewById(R.id.imageview1);
      title = (TextView) findViewById(R.id.title);
      data = getSharedPreferences("data", Activity.MODE_PRIVATE);
      deleteDialog = new AlertDialog.Builder(this);
      week = getSharedPreferences("week", Activity.MODE_PRIVATE);

      close.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            finish();
         }
      });

      delete.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            final AlertDialog.Builder alert = new
AlertDialog.Builder(ViewWeightActivity.this);
            view =
getLayoutInflater().inflate(R.layout.dialog_default,null);
            TextView apply = (TextView)view.findViewById(R.id.apply);
            TextView cancel = (TextView)view.findViewById(R.id.cancel);
            TextView title = (TextView)view.findViewById(R.id.title);
            TextView message = (TextView)view.findViewById(R.id.message);
            ImageView image = view.findViewById(R.id.image);
            LinearLayout parent = view.findViewById(R.id.parent);
            alert.setView(view);
            final AlertDialog dialog = alert.create();
            setBackground(parent,(double)50,(double)5,colorPrimary,false);
            setBackground(apply,(double)50,(double)0,colorPrimary,true);
            setBackground(cancel,(double)50,(double)0,colorPrimary,true);
            if (true) {
               ObjectAnimator scaleX = new ObjectAnimator();
               scaleX.setTarget(parent);
               scaleX.setPropertyName("scaleX");
               scaleX.setFloatValues((float)0.8f,(float)1f);
               scaleX.setDuration(300);
               scaleX.setInterpolator(new
AccelerateDecelerateInterpolator());
               scaleX.start();
               ObjectAnimator scaleY = new ObjectAnimator();
               scaleY.setTarget(parent);
               scaleY.setPropertyName("scaleY");
               scaleY.setFloatValues((float)0.8f,(float)1f);
               scaleY.setDuration(300);
               scaleY.setInterpolator(new
```

```
AccelerateDecelerateInterpolator());
                scaleY.start();
                ObjectAnimator alpha = new ObjectAnimator();
                alpha.setTarget(parent);
                alpha.setPropertyName("alpha");
                alpha.setFloatValues((float)0f,(float)1);
                alpha.setDuration(300);
                alpha.setInterpolator(new
AccelerateDecelerateInterpolator());
                alpha.start();
            }
            if (true) {
                title.setText("Destroy");
                message.setText("You want to destroy");
                apply.setText("DESTROY");
                cancel.setText("CANCEL");
            }
            if (isDark) {
                apply.setTextColor(Color.parseColor("#ffffff"));
                cancel.setTextColor(Color.parseColor("#ffffff"));
                title.setTextColor(Color.parseColor("#ffffff"));
                message.setTextColor(Color.parseColor("#ffffff"));
                image.setImageResource(R.drawable.delete_white);
            }
            else {
                apply.setTextColor(Color.parseColor("#424242"));
                cancel.setTextColor(Color.parseColor("#424242"));
                title.setTextColor(Color.parseColor("#424242"));
                message.setTextColor(Color.parseColor("#424242"));
                image.setImageResource(R.drawable.delete_black);
            }
            dialog.getWindow().setBackgroundDrawable(new
android.graphics.drawable.ColorDrawable(android.graphics.Color.TRANSPARENT)
);
            apply.setOnClickListener(new View.OnClickListener() {
                @Override public void onClick(View v) {
                    if (true) {
                        ID = getIntent().getStringExtra("id");
                        list = new Gson().fromJson(data.getString("list", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
                        n = 0;
                        for(int _repeat120 = 0; _repeat120 <
(int)(list.size()); _repeat120++) {
                            if
(list.get((int)n).get("id").toString().equals(ID)) {
                                list.remove((int)(n));
                                data.edit().putString("list", new
Gson().toJson(list)).commit();
                                ToastUtil.showMessage(getApplicationContext(),
"Deleted");
                                finish();
                            }
                            n++;
                        }
                    }
                    dialog.dismiss();}});
            cancel.setOnClickListener(new View.OnClickListener() {
                @Override public void onClick(View v) {
                    if (true) {

                    }
```

```
                    dialog.dismiss();}});
                dialog.show();
            }
        });

        copy.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                ((ClipboardManager)
getSystemService(getApplicationContext().CLIPBOARD_SERVICE)).setPrimaryClip
(ClipData.newPlainText("clipboard", text.getText().toString()));
                ToastUtil.showMessage(getApplicationContext(), "Copied");
            }
        });

        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                ID = getIntent().getStringExtra("id");
                list = new Gson().fromJson(data.getString("list", ""), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
                n = 0;
                for(int _repeat30 = 0; _repeat30 < (int)(list.size());
_repeat30++) {
                    if (list.get((int)n).get("id").toString().equals(ID)) {
                        intent.setClass(getApplicationContext(),
WeightEditorActivity.class);
                        intent.putExtra("activity", "Edit");
                        intent.putExtra("id",
list.get((int)n).get("id").toString());
                        intent.putExtra("type",
list.get((int)n).get("type").toString());
                        intent.putExtra("title",
list.get((int)n).get("title").toString());
                        intent.putExtra("text",
list.get((int)n).get("text").toString());
                        intent.putExtra("date",
list.get((int)n).get("date").toString());
                        intent.putExtra("updatedDate",
list.get((int)n).get("updatedDate").toString());
                        startActivity(intent);
                        break;
                    }
                    n++;
                }
            }
        });
    }

    private void initializeLogic() {
        isDark = data.getString("theme", "").equals("dark");
        UI(data.getString("accent", ""), isDark);
        startApp();
        overridePendingTransition(0, 0);
        text.setTextIsSelectable(true);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);
```

```
      switch (requestCode) {

         default:
         break;
      }
   }

   @Override
   public void onStart() {
      super.onStart();
      ID = getIntent().getStringExtra("id");
      list = new Gson().fromJson(data.getString("list", ""), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      n = 0;
      for(int _repeat16 = 0; _repeat16 < (int)(list.size()); _repeat16++) {
         if (list.get((int)n).get("id").toString().equals(ID)) {
            title.setText(list.get((int)n).get("title").toString());
            text.setText("Week: ".concat(week.getString("week", "")));

text.setMovementMethod(android.text.method.LinkMovementMethod.getInstance()
);

            android.text.util.Linkify.addLinks(text,
android.text.util.Linkify.ALL);
            break;
         }
         n++;
      }
   }

   @Override
   public void onPause() {
      super.onPause();
      overridePendingTransition(0, 0);
   }
   public void UI (final String accent, final boolean dark) {
      if (dark) {
         colorPrimary = "#202125";
         getWindow().setStatusBarColor(Color.parseColor(colorPrimary));

         getWindow().setNavigationBarColor(Color.parseColor(colorPrimary));
         View decor = getWindow().getDecorView();
         decor.setSystemUiVisibility(0);
         setBackground(main, 0, 0, colorPrimary, false);
         gradientDrawable(action, 0, 0, 10, colorPrimary, colorPrimary,
false, false, 0);
         gradientDrawable(close, 50, 0, 0, colorPrimary, colorPrimary,
true, true, 200);
         gradientDrawable(delete, 50, 0, 0, colorPrimary, colorPrimary,
true, true, 200);
         gradientDrawable(copy, 50, 0, 0, colorPrimary, colorPrimary, true,
true, 200);
         top_title.setTextColor(0xFFE0E1E5);
         title.setTextColor(0xFFE0E1E5);
         text.setTextColor(0xFFE0E1E5);
         close.setImageResource(R.drawable.close_black);
         delete.setImageResource(R.drawable.delete_white);
         copy.setImageResource(R.drawable.copy_black);
         close.setColorFilter(0xFFFFFFFF, PorterDuff.Mode.MULTIPLY);
         copy.setColorFilter(0xFFFFFFFF, PorterDuff.Mode.MULTIPLY);
      }
```

```
      else {
         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_
STATUS_BAR); }
         colorPrimary = "#ffffff";
         getWindow().setStatusBarColor(Color.parseColor(colorPrimary));

         getWindow().setNavigationBarColor(Color.parseColor(colorPrimary));
         setBackground(main, 0, 0, colorPrimary, false);
         gradientDrawable(action, 0, 0, 10, colorPrimary, colorPrimary,
false, false, 0);
         gradientDrawable(close, 50, 0, 0, colorPrimary, colorPrimary,
true, true, 200);
         gradientDrawable(delete, 50, 0, 0, colorPrimary, colorPrimary,
true, true, 200);
         gradientDrawable(copy, 50, 0, 0, colorPrimary, colorPrimary, true,
true, 200);
         top_title.setTextColor(0xFF424242);
         title.setTextColor(0xFF424242);
         text.setTextColor(0xFF424242);
         close.setImageResource(R.drawable.close_black);
         delete.setImageResource(R.drawable.delete_black);
         copy.setImageResource(R.drawable.copy_black);
      }
   }


   public void gradientDrawable (final View view, final double radius,
final double msgtroke, final double msghadow, final String color, final
String _borderColor, final boolean ripple, final boolean _clickAnim, final
double _animDuration) {
      if (ripple) {
         android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
         gd.setColor(Color.parseColor(color));
         gd.setCornerRadius((int)radius);
         gd.setStroke((int)msgtroke,Color.parseColor(_borderColor));
         if (Build.VERSION.SDK_INT >= 21){
            view.setElevation((int)msghadow);}
         android.content.res.ColorStateList clrb = new
android.content.res.ColorStateList(new int[][]{new int[]{}}, new
int[]{Color.parseColor("#9e9e9e")});
         android.graphics.drawable.RippleDrawable ripdrb = new
android.graphics.drawable.RippleDrawable(clrb , gd, null);
         view.setClickable(true);
         view.setBackground(ripdrb);
      }
      else {
         android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
         gd.setColor(Color.parseColor(color));
         gd.setCornerRadius((int)radius);
         gd.setStroke((int)msgtroke,Color.parseColor(_borderColor));
         view.setBackground(gd);
         if (Build.VERSION.SDK_INT >= 21){
            view.setElevation((int)msghadow);}
      }
      if (_clickAnim) {
         view.setOnTouchListener(new View.OnTouchListener() {
            @Override
            public boolean onTouch(View v, MotionEvent event) {
```

```
                switch (event.getAction()){
                    case MotionEvent.ACTION_DOWN:{
                        ObjectAnimator scaleX = new ObjectAnimator();
                        scaleX.setTarget(view);
                        scaleX.setPropertyName("scaleX");
                        scaleX.setFloatValues(0.9f);
                        scaleX.setDuration((int)_animDuration);
                        scaleX.start();

                        ObjectAnimator scaleY = new ObjectAnimator();
                        scaleY.setTarget(view);
                        scaleY.setPropertyName("scaleY");
                        scaleY.setFloatValues(0.9f);
                        scaleY.setDuration((int)_animDuration);
                        scaleY.start();
                        break;
                    }
                    case MotionEvent.ACTION_UP:{

                        ObjectAnimator scaleX = new ObjectAnimator();
                        scaleX.setTarget(view);
                        scaleX.setPropertyName("scaleX");
                        scaleX.setFloatValues((float)1);
                        scaleX.setDuration((int)_animDuration);
                        scaleX.start();

                        ObjectAnimator scaleY = new ObjectAnimator();
                        scaleY.setTarget(view);
                        scaleY.setPropertyName("scaleY");
                        scaleY.setFloatValues((float)1);
                        scaleY.setDuration((int)_animDuration);
                        scaleY.start();

                        break;
                    }
                }
                return false;
            }
        });
    }
}


   public void setBackground (final View view, final double radius, final
double msghadow, final String color, final boolean ripple) {
      if (ripple) {
         android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
         gd.setColor(Color.parseColor(color));
         gd.setCornerRadius((int)radius);
         view.setElevation((int)msghadow);
         android.content.res.ColorStateList clrb = new
android.content.res.ColorStateList(new int[][]{new int[]{}}, new
int[]{Color.parseColor("#9e9e9e")});
         android.graphics.drawable.RippleDrawable ripdrb = new
android.graphics.drawable.RippleDrawable(clrb , gd, null);
         view.setClickable(true);
         view.setBackground(ripdrb);
      }
      else {
         android.graphics.drawable.GradientDrawable gd = new
```

```
android.graphics.drawable.GradientDrawable();
            gd.setColor(Color.parseColor(color));
            gd.setCornerRadius((int)radius);
            view.setBackground(gd);
            view.setElevation((int)msghadow);
        }
    }


    public void startApp () {

    }


    @Deprecated
    public void showMessage(String msg) {
        Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
    }

    @Deprecated
    public int getLocationX(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
```

```
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Weight Editor Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.Manifest;
import android.animation.ObjectAnimator;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.ClipData;
import android.content.ClipboardManager;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.MotionEvent;
import android.view.View;
import android.view.animation.AccelerateDecelerateInterpolator;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import com.sdsmdg.tastytoast.TastyToast;

import java.text.SimpleDateFormat;
```

```
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.Random;
import java.util.Timer;
import java.util.TimerTask;


public class WeightEditorActivity extends AppCompatActivity {
   public final int REQ_CD_PICKER = 101;
   private Timer timer = new Timer();

   private String primaryColor = "";
   private HashMap<String, Object> map = new HashMap<>();
   private String filePath = "";
   private String generateID = "";
   private String ID = "";
   private String Date = "";
   private String UpdatedDate = "";
   private String type = "";
   private double n = 0;
   private double listLength = 0;
   private double test = 0;
   private boolean isDark = false;
   private double weeks = 0;
   private double weight = 0;

   private ArrayList<HashMap<String, Object>> list = new ArrayList<>();
   private ArrayList<String> path = new ArrayList<>();

   private LinearLayout parent;
   private LinearLayout fragment1;
   private LinearLayout action;
   private LinearLayout linear4;
   private ImageView close;
   private TextView textview1;
   private ImageView cut;
   private ImageView copy;
   private ImageView folder;
   private ImageView done;
   private EditText title;
   private EditText text;

   private SharedPreferences data;
   private Calendar calendar = Calendar.getInstance();
   private Intent picker = new Intent(Intent.ACTION_GET_CONTENT);
   private AlertDialog.Builder cutDialog;
   private Intent intent = new Intent();
   private TimerTask _timer;
   private SharedPreferences week;
   private SharedPreferences userdetail;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.weighteditor);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_DENIED) {
```

```
        ActivityCompat.requestPermissions(this, new String[]
{Manifest.permission.READ_EXTERNAL_STORAGE}, 1000);
      }
      else {
        initializeLogic();
      }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[]
permissions, int[] grantResults) {
      super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
      if (requestCode == 1000) {
        initializeLogic();
      }
    }

    private void initialize(Bundle savedInstanceState) {
      parent =  findViewById(R.id.parent);
      fragment1 =  findViewById(R.id.fragment1);
      action =  findViewById(R.id.action);
      linear4 =  findViewById(R.id.linear4);
      close =  findViewById(R.id.close);
      textview1 = (TextView) findViewById(R.id.textview1);
      cut =  findViewById(R.id.cut);
      copy =  findViewById(R.id.copy);
      folder =  findViewById(R.id.folder);
      done =  findViewById(R.id.done);
      title = (EditText) findViewById(R.id.title);
      text = (EditText) findViewById(R.id.text);
      data = getSharedPreferences("data", Activity.MODE_PRIVATE);
      picker.setType("*/*");
      picker.putExtra(Intent.EXTRA_ALLOW_MULTIPLE, true);
      cutDialog = new AlertDialog.Builder(this);
      week = getSharedPreferences("week", Activity.MODE_PRIVATE);
      userdetail = getSharedPreferences("userdetail",
Activity.MODE_PRIVATE);

      close.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
          android.view.inputmethod.InputMethodManager imm =
(android.view.inputmethod.InputMethodManager)
getSystemService(Activity.INPUT_METHOD_SERVICE);
imm.hideSoftInputFromWindow(text.getWindowToken(), 0);
          intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
          finish();
        }
      });

      cut.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
          final AlertDialog.Builder alert = new
AlertDialog.Builder(WeightEditorActivity.this);
          view =
getLayoutInflater().inflate(R.layout.neutral_dialog,null);
          TextView apply = (TextView)view.findViewById(R.id.apply);
          TextView cancel = (TextView)view.findViewById(R.id.cancel);
          TextView title = (TextView)view.findViewById(R.id.title);
```

```
            TextView message = (TextView)view.findViewById(R.id.message);
            TextView neutral = (TextView)view.findViewById(R.id.neutral);
            ImageView image = view.findViewById(R.id.image);
            LinearLayout parent = view.findViewById(R.id.parent);
            alert.setView(view);
            final AlertDialog dialog = alert.create();
            setBackground(parent,(double)50,(double)5,primaryColor,false);
            setBackground(apply,(double)50,(double)0,primaryColor,true);
            setBackground(cancel,(double)50,(double)0,primaryColor,true);
            setBackground(neutral,(double)50,(double)0,primaryColor,true);
            if (true) {
               ObjectAnimator scaleX = new ObjectAnimator();
               scaleX.setTarget(parent);
               scaleX.setPropertyName("scaleX");
               scaleX.setFloatValues((float)0.8f,(float)1f);
               scaleX.setDuration(300);
               scaleX.setInterpolator(new
AccelerateDecelerateInterpolator());
               scaleX.start();
               ObjectAnimator scaleY = new ObjectAnimator();
               scaleY.setTarget(parent);
               scaleY.setPropertyName("scaleY");
               scaleY.setFloatValues((float)0.8f,(float)1f);
               scaleY.setDuration(300);
               scaleY.setInterpolator(new
AccelerateDecelerateInterpolator());
               scaleY.start();
               ObjectAnimator alpha = new ObjectAnimator();
               alpha.setTarget(parent);
               alpha.setPropertyName("alpha");
               alpha.setFloatValues((float)0f,(float)1);
               alpha.setDuration(300);
               alpha.setInterpolator(new
AccelerateDecelerateInterpolator());
               alpha.start();
            }
            if (true) {
               title.setText("Clean");
               message.setText("Would you like to clean it all up?");
               apply.setText("CLEAN");
               cancel.setText("CANCEL");
               neutral.setText("Copy&Clean");
            }
            if (isDark) {
               apply.setTextColor(Color.parseColor("#"));
               cancel.setTextColor(Color.parseColor("#"));
               neutral.setTextColor(Color.parseColor("#ffffff"));
               title.setTextColor(Color.parseColor("#ffffff"));
               message.setTextColor(Color.parseColor("#ffffff"));
               image.setImageResource(R.drawable.cut_white);
            }
            else {
               apply.setTextColor(Color.parseColor("#424242"));
               cancel.setTextColor(Color.parseColor("#424242"));
               neutral.setTextColor(Color.parseColor("#424242"));
               title.setTextColor(Color.parseColor("#424242"));
               message.setTextColor(Color.parseColor("#424242"));
               image.setImageResource(R.drawable.cut_black);
            }
            dialog.getWindow().setBackgroundDrawable(new
android.graphics.drawable.ColorDrawable(android.graphics.Color.TRANSPARENT)
```

```
);
            apply.setOnClickListener(new View.OnClickListener() {
               @Override public void onClick(View v) {
                  if (true) {
                     text.setText("");
                     ToastUtil.showMessage(getApplicationContext(),
"Cleaned");
                  }
                  dialog.dismiss();}});
            cancel.setOnClickListener(new View.OnClickListener() {
               @Override public void onClick(View v) {
                  if (true) {

                  }
                  dialog.dismiss();}});
            dialog.show();
            neutral.setOnClickListener(new View.OnClickListener() {
               @Override public void onClick(View v) {
                  if (true) {
                     ((ClipboardManager)
getSystemService(getApplicationContext().CLIPBOARD_SERVICE)).setPrimaryClip
(ClipData.newPlainText("clipboard", text.getText().toString()));
                     text.setText("");
                     ToastUtil.showMessage(getApplicationContext(), "Copied
and cleaned");
                  }
                  dialog.dismiss();}});
            dialog.show();
         }
      });

      copy.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            ((ClipboardManager)
getSystemService(getApplicationContext().CLIPBOARD_SERVICE)).setPrimaryClip
(ClipData.newPlainText("clipboard", text.getText().toString()));
            ToastUtil.showMessage(getApplicationContext(), "Copied");
         }
      });

      folder.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            android.view.inputmethod.InputMethodManager imm =
(android.view.inputmethod.InputMethodManager)
getSystemService(Activity.INPUT_METHOD_SERVICE);
imm.hideSoftInputFromWindow(text.getWindowToken(), 0);
            startActivityForResult(picker, REQ_CD_PICKER);
         }
      });

      done.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            weeks++;
            text.setText(String.valueOf((long)(weeks)));
            list.clear();
            if (getIntent().getStringExtra("activity").equals("Edit")) {
               list = new Gson().fromJson(data.getString("list", ""), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
```

```
                test = 0;
                n = 0;
                listLength = list.size();
                for(int _repeat63 = 0; _repeat63 < (int)(listLength);
_repeat63++) {
                    if (list.get((int)n).get("id").toString().equals(ID)) {
                        calendar = Calendar.getInstance();
                        map = new HashMap<>();
                        map.put("id", ID);
                        map.put("type", "text");
                        map.put("title", title.getText().toString());
                        map.put("text", text.getText().toString());
                        map.put("date",
list.get((int)n).get("date").toString());
                        map.put("updatedDate", new SimpleDateFormat("E, dd MMM
yyyy HH:mm:ss").format(calendar.getTime()));
                        list.add((int)n, map);
                        list.remove((int)(n + 1));
                        week.edit().putString("week",
text.getText().toString()).commit();
                        data.edit().putString("list", new
Gson().toJson(list)).commit();
                        android.view.inputmethod.InputMethodManager imm =
(android.view.inputmethod.InputMethodManager)
getSystemService(Activity.INPUT_METHOD_SERVICE);
imm.hideSoftInputFromWindow(text.getWindowToken(), 0);
                        finish();
                    }
                    n++;
                }
            }
            else {
                if ((Double.parseDouble(title.getText().toString()) > 1000)
|| (title.getText().toString().trim().length() == 0)) {
                    ToastUtil.showMessage(getApplicationContext(), "Invalid
data entered");
                }
                else {
                    if (!data.getString("list", "").equals("")) {
                        list = new Gson().fromJson(data.getString("list", ""),
new TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
                    }
                    generateID =
String.valueOf((long)(ToastUtil.getRandom((int)(100000000),
(int)(999999999))));
                    calendar = Calendar.getInstance();
                    map = new HashMap<>();
                    map.put("id", generateID);
                    map.put("type", "text");
                    map.put("title", title.getText().toString());
                    map.put("text", text.getText().toString());
                    map.put("date", new SimpleDateFormat("E, dd MMM yyyy
HH:mm:ss").format(calendar.getTime()));
                    map.put("updatedDate", new SimpleDateFormat("E, dd MMM
yyyy HH:mm:ss").format(calendar.getTime()));
                    list.add(map);
                    data.edit().putString("list", new
Gson().toJson(list)).commit();
                    android.view.inputmethod.InputMethodManager imm =
(android.view.inputmethod.InputMethodManager)
getSystemService(Activity.INPUT_METHOD_SERVICE);
```

```
imm.hideSoftInputFromWindow(text.getWindowToken(), 0);
                if (Double.parseDouble(title.getText().toString()) <
weight) {
                    TastyToast.makeText(getApplicationContext(), "Your
Weight Is Reducing", TastyToast.LENGTH_LONG, TastyToast.SUCCESS);
                    finish();
                }
                else {
                    TastyToast.makeText(getApplicationContext(), "Your
Weight Has Increased", TastyToast.LENGTH_LONG, TastyToast.SUCCESS);
                    finish();
                }
            }
        }
    }
  });
  }

  private void initializeLogic() {
    text.setEnabled(false);
    isDark = data.getString("theme", "").equals("dark");
    UI(data.getString("accent", ""), isDark);
    overridePendingTransition(0, 0);
    _detectActivity(getIntent().getStringExtra("activity"));
    if (week.getString("week", "").equals("")) {
      weeks = 0;
    }
    else {
      text.setText(week.getString("week", ""));
      weeks = Double.parseDouble(text.getText().toString());
    }
    try{
      weight = Double.parseDouble(userdetail.getString("weight", ""));
    }catch(Exception e){
    }
  }

  @Override
  protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode) {
      case REQ_CD_PICKER:
      if (resultCode == Activity.RESULT_OK) {
        ArrayList<String> _filePath = new ArrayList<>();
        if (data != null) {
          if (data.getClipData() != null) {
            for (int index = 0; index <
data.getClipData().getItemCount(); index++) {
              ClipData.Item _item =
data.getClipData().getItemAt(index);

_filePath.add(FileUtil.convertUriToFilePath(getApplicationContext(),
_item.getUri()));
            }
          }
          else {

_filePath.add(FileUtil.convertUriToFilePath(getApplicationContext(),
data.getData()));
          }
```

```
            }
            filePath = _filePath.get((int)(0));
            title.setText(Uri.parse(filePath).getLastPathSegment());
            text.setText(FileUtil.readFile(filePath));
         }
         else {

         }
         break;
         default:
         break;
      }
   }

   @Override
   public void onPause() {
      super.onPause();
      overridePendingTransition(0, 0);
   }
   public void UI (final String accent, final boolean dark) {
      if (dark) {
         primaryColor = "#202125";
         getWindow().setStatusBarColor(Color.parseColor(primaryColor));

         getWindow().setNavigationBarColor(Color.parseColor(primaryColor));
         View decor = getWindow().getDecorView();
         decor.setSystemUiVisibility(0);
         gradientDrawable(parent, 0, 0, 0, primaryColor, primaryColor,
false, false, 200);
         gradientDrawable(action, 0, 0, 10, primaryColor, primaryColor,
false, false, 0);
         gradientDrawable(close, 50, 0, 0, primaryColor, primaryColor,
true, true, 200);
         gradientDrawable(cut, 50, 0, 0, primaryColor, primaryColor, true,
true, 200);
         gradientDrawable(copy, 50, 0, 0, primaryColor, primaryColor, true,
true, 200);
         gradientDrawable(folder, 50, 0, 0, primaryColor, primaryColor,
true, true, 200);
         gradientDrawable(done, 50, 0, 0, primaryColor, primaryColor, true,
true, 200);
         textview1.setTextColor(0xFFE0E1E5);
         title.setTextColor(0xFFE0E1E5);
         text.setTextColor(0xFFE0E1E5);
         close.setImageResource(R.drawable.backward_white);
         cut.setImageResource(R.drawable.cut_white);
         copy.setImageResource(R.drawable.copy_black);
         folder.setImageResource(R.drawable.openfile_black);
         done.setImageResource(R.drawable.done_black);
         title.setHintTextColor(Color.parseColor("#30ffffff"));
         text.setHintTextColor(Color.parseColor("#30ffffff"));
         copy.setColorFilter(0xFFFFFFFF, PorterDuff.Mode.MULTIPLY);
         folder.setColorFilter(0xFFFFFFFF, PorterDuff.Mode.MULTIPLY);
         done.setColorFilter(0xFFFFFFFF, PorterDuff.Mode.MULTIPLY);
      }
      else {
         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_
STATUS_BAR); }
         primaryColor = "#ffffff";
         getWindow().setStatusBarColor(Color.parseColor(primaryColor));
```

```
        getWindow().setNavigationBarColor(Color.parseColor(primaryColor));
        gradientDrawable(parent, 0, 0, 0, primaryColor, primaryColor,
false, false, 200);
        gradientDrawable(action, 0, 0, 10, primaryColor, primaryColor,
false, false, 0);
        gradientDrawable(close, 50, 0, 0, primaryColor, primaryColor,
true, true, 200);
        gradientDrawable(cut, 50, 0, 0, primaryColor, primaryColor, true,
true, 200);
        gradientDrawable(copy, 50, 0, 0, primaryColor, primaryColor, true,
true, 200);
        gradientDrawable(folder, 50, 0, 0, primaryColor, primaryColor,
true, true, 200);
        gradientDrawable(done, 50, 0, 0, primaryColor, primaryColor, true,
true, 200);
        textview1.setTextColor(0xFF424242);
        title.setTextColor(0xFF424242);
        text.setTextColor(0xFF424242);
        close.setImageResource(R.drawable.close_black);
        cut.setImageResource(R.drawable.cut_black);
        copy.setImageResource(R.drawable.copy_black);
        folder.setImageResource(R.drawable.openfile_black);
        done.setImageResource(R.drawable.done_black);
        title.setHintTextColor(Color.parseColor("#30000000"));
        text.setHintTextColor(Color.parseColor("#30000000"));
      }
   }


   public void gradientDrawable (final View view, final double radius,
final double msgtroke, final double msghadow, final String color, final
String _borderColor, final boolean ripple, final boolean _clickAnim, final
double _animDuration) {
      if (ripple) {
        android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
        gd.setColor(Color.parseColor(color));
        gd.setCornerRadius((int)radius);
        gd.setStroke((int)msgtroke,Color.parseColor(_borderColor));
        if (Build.VERSION.SDK_INT >= 21){
          view.setElevation((int)msghadow);}
        android.content.res.ColorStateList clrb = new
android.content.res.ColorStateList(new int[][]{new int[]{}}, new
int[]{Color.parseColor("#9e9e9e")});
        android.graphics.drawable.RippleDrawable ripdrb = new
android.graphics.drawable.RippleDrawable(clrb , gd, null);
        view.setClickable(true);
        view.setBackground(ripdrb);
      }
      else {
        android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
        gd.setColor(Color.parseColor(color));
        gd.setCornerRadius((int)radius);
        gd.setStroke((int)msgtroke,Color.parseColor(_borderColor));
        view.setBackground(gd);
        if (Build.VERSION.SDK_INT >= 21){
          view.setElevation((int)msghadow);}
      }
      if (_clickAnim) {
```

```
        view.setOnTouchListener(new View.OnTouchListener() {
           @Override
           public boolean onTouch(View v, MotionEvent event) {
              switch (event.getAction()){
                 case MotionEvent.ACTION_DOWN:{
                    ObjectAnimator scaleX = new ObjectAnimator();
                    scaleX.setTarget(view);
                    scaleX.setPropertyName("scaleX");
                    scaleX.setFloatValues(0.9f);
                    scaleX.setDuration((int)_animDuration);
                    scaleX.start();

                    ObjectAnimator scaleY = new ObjectAnimator();
                    scaleY.setTarget(view);
                    scaleY.setPropertyName("scaleY");
                    scaleY.setFloatValues(0.9f);
                    scaleY.setDuration((int)_animDuration);
                    scaleY.start();
                    break;
                 }
                 case MotionEvent.ACTION_UP:{

                    ObjectAnimator scaleX = new ObjectAnimator();
                    scaleX.setTarget(view);
                    scaleX.setPropertyName("scaleX");
                    scaleX.setFloatValues((float)1);
                    scaleX.setDuration((int)_animDuration);
                    scaleX.start();

                    ObjectAnimator scaleY = new ObjectAnimator();
                    scaleY.setTarget(view);
                    scaleY.setPropertyName("scaleY");
                    scaleY.setFloatValues((float)1);
                    scaleY.setDuration((int)_animDuration);
                    scaleY.start();

                    break;
                 }
              }
              return false;
           }
        });
     }
  }


  public void _detectActivity (final String msgtring) {
     if (msgtring.equals("Edit")) {
        ID = getIntent().getStringExtra("id");
        Date = getIntent().getStringExtra("date");
        UpdatedDate = getIntent().getStringExtra("updatedDate");
        type = getIntent().getStringExtra("type");
        title.setText(getIntent().getStringExtra("title"));
        text.setText(getIntent().getStringExtra("text"));
        text.requestFocus();
        android.view.inputmethod.InputMethodManager inputMethodManager =
(android.view.inputmethod.InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE); if (inputMethodManager !=
null) {
inputMethodManager.toggleSoftInput(android.view.inputmethod.InputMethodMana
ger.SHOW_FORCED, 0); }
```

```
      }
      else {

      }
   }


   public void setBackground (final View view, final double radius, final
double msghadow, final String color, final boolean ripple) {
      if (ripple) {
         android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
         gd.setColor(Color.parseColor(color));
         gd.setCornerRadius((int)radius);
         view.setElevation((int)msghadow);
         android.content.res.ColorStateList clrb = new
android.content.res.ColorStateList(new int[][]{new int[]{}}, new
int[]{Color.parseColor("#9e9e9e")});
         android.graphics.drawable.RippleDrawable ripdrb = new
android.graphics.drawable.RippleDrawable(clrb , gd, null);
         view.setClickable(true);
         view.setBackground(ripdrb);
      }
      else {
         android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
         gd.setColor(Color.parseColor(color));
         gd.setCornerRadius((int)radius);
         view.setBackground(gd);
         view.setElevation((int)msghadow);
      }
   }


   @Deprecated
   public void showMessage(String msg) {
      Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
   }

   @Deprecated
   public int getLocationX(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[0];
   }

   @Deprecated
   public int getLocationY(View v) {
      int location[] = new int[2];
      v.getLocationInWindow(location);
      return location[1];
   }

   @Deprecated
   public int getRandom(int min, int max) {
      Random random = new Random();
      return random.nextInt(max - min + 1) + min;
   }

   @Deprecated
```

```
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
           if (arr.valueAt(iIdx))
           result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```

Weight Editor Activity

```
package com.vogella.android.diettrackerapplication;

// Diet Tracker Application
// Name: Emmanuel Ayelabola
// Student Number: C00242748

import android.animation.ArgbEvaluator;
import android.animation.ObjectAnimator;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.drawable.GradientDrawable;
import android.graphics.text.LineBreaker;
import android.os.Build;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.SparseBooleanArray;
import android.util.TypedValue;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.View.OnLongClickListener;
import android.view.ViewGroup;
import android.view.WindowManager;
```

```
import android.view.animation.AccelerateDecelerateInterpolator;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;

import
com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Random;
import java.util.Timer;
import java.util.TimerTask;


public class WeightLogsActivity extends AppCompatActivity {
   private Timer timer = new Timer();

   private FloatingActionButton fab;
   private boolean isSearch = false;
   private String colorPrimary = "";
   private double n = 0;
   private double m = 0;
   private boolean isLeftBar = false;
   private String colorPrimaryDark = "";
   private boolean isDark = false;
   private String strvalue = "";
   private double viewIndex = 0;
   private double viewTop = 0;

   private ArrayList<HashMap<String, Object>> list = new ArrayList<>();
   private ArrayList<HashMap<String, Object>> Notes_List = new
ArrayList<>();

   private LinearLayout main;
   private LinearLayout fragment1;
   private LinearLayout left_bar;
   private LinearLayout mask;
   private LinearLayout action;
   private LinearLayout linear4;
   private ImageView menu;
   private TextView title;
   private EditText search;
   private ImageView search_icon;
   private TextView txtRegWeight;
   private ListView listview;
   private LinearLayout empty_layout;
   private ImageView imageview1;
   private TextView textview1;
   private LinearLayout linear6;
   private LinearLayout linear7;
```

```
   private LinearLayout linear9;
   private LinearLayout linear8;
   private ImageView cancel;
   private ImageView features;
   private ImageView plugins;
   private ImageView settings;
   private ImageView darker;

   private Intent intent = new Intent();
   private SharedPreferences data;
   private TimerTask _timer;
   private SharedPreferences AllMeals;
   private Intent toNotes = new Intent();
   private SharedPreferences userdetail;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.weightlogs);
      initialize(savedInstanceState);
      com.google.firebase.FirebaseApp.initializeApp(this);
      initializeLogic();
   }

   private void initialize(Bundle savedInstanceState) {
      fab = (FloatingActionButton) findViewById(R.id.fab);

      main =  findViewById(R.id.main);
      fragment1 =  findViewById(R.id.fragment1);
      left_bar =  findViewById(R.id.left_bar);
      mask =  findViewById(R.id.mask);
      action =  findViewById(R.id.action);
      linear4 =  findViewById(R.id.linear4);
      menu =  findViewById(R.id.menu);
      title = (TextView) findViewById(R.id.title);
      search = (EditText) findViewById(R.id.search);
      search_icon =  findViewById(R.id.search_icon);
      txtRegWeight = (TextView) findViewById(R.id.txtRegWeight);
      listview = (ListView) findViewById(R.id.listview);
      empty_layout =  findViewById(R.id.empty_layout);
      imageview1 =  findViewById(R.id.imageview1);
      textview1 = (TextView) findViewById(R.id.textview1);
      linear6 =  findViewById(R.id.linear6);
      linear7 =  findViewById(R.id.linear7);
      linear9 =  findViewById(R.id.linear9);
      linear8 =  findViewById(R.id.linear8);
      cancel =  findViewById(R.id.cancel);
      features =  findViewById(R.id.features);
      plugins =  findViewById(R.id.plugins);
      settings =  findViewById(R.id.settings);
      darker =  findViewById(R.id.darker);
      data = getSharedPreferences("data", Activity.MODE_PRIVATE);
      AllMeals = getSharedPreferences("All Meals", Activity.MODE_PRIVATE);
      userdetail = getSharedPreferences("userdetail",
Activity.MODE_PRIVATE);

      mask.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            isLeftBar = false;
            openCloseLeftBar(isLeftBar);
```

```
         }
      });

      menu.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            if (isLeftBar) {
               isLeftBar = false;
               openCloseLeftBar(isLeftBar);
               search.setEnabled(false);
            }
            else {
               isLeftBar = true;
               openCloseLeftBar(isLeftBar);
               search.setEnabled(true);
            }
            android.view.inputmethod.InputMethodManager imm =
(android.view.inputmethod.InputMethodManager)
                  getSystemService(Activity.INPUT_METHOD_SERVICE);
            imm.hideSoftInputFromWindow(search.getWindowToken(), 0);
         }
      });

      search.addTextChangedListener(new TextWatcher() {
         @Override
         public void onTextChanged(CharSequence param1, int param2, int
param3, int param4) {
            final String _charSeq = param1.toString();
            getAllData(list);
            if (_charSeq.length() > 0) {
               n = list.size() - 1;
               m = list.size();
               for(int i = 0; i < (int)(m); i++) {
                  if
(list.get((int)n).get("title").toString().toLowerCase().contains(_charSeq.t
oLowerCase())) {

                  }
                  else {
                     list.remove((int)(n));
                  }
                  n--;
               }
            }
            listview.setAdapter(new ListviewAdapter(list));
            ((BaseAdapter)listview.getAdapter()).notifyDataSetChanged();
         }

         @Override
         public void beforeTextChanged(CharSequence param1, int param2, int
param3, int param4) {

         }

         @Override
         public void afterTextChanged(Editable param1) {

         }
      });

      search_icon.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View view) {
           transitionManager(action, 100);
           if (isSearch) {
              isSearch = false;
              title.setVisibility(View.VISIBLE);
              search.setVisibility(View.GONE);
              menu.setVisibility(View.VISIBLE);
              search.setText("");
              android.view.inputmethod.InputMethodManager imm =
(android.view.inputmethod.InputMethodManager)
getSystemService(Activity.INPUT_METHOD_SERVICE);
imm.hideSoftInputFromWindow(search.getWindowToken(), 0);
              if (isDark) {
                 search_icon.setImageResource(R.drawable.default_image);
              }
              else {
                 search_icon.setImageResource(R.drawable.search_black);
              }
              _timer = new TimerTask() {
                 @Override
                 public void run() {
                    runOnUiThread(new Runnable() {
                       @Override
                       public void run() {
                          fabAnimator("alpha", 1, 200);
                          fabAnimator("translationY", 0, 200);
                       }
                    });
                 }
              };
              timer.schedule(_timer, (int)(500));
           }
           else {
              isSearch = true;
              title.setVisibility(View.GONE);
              search.setVisibility(View.VISIBLE);
              menu.setVisibility(View.GONE);
              search.requestFocus();
              android.view.inputmethod.InputMethodManager
inputMethodManager = (android.view.inputmethod.InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
              if (inputMethodManager != null) {
inputMethodManager.toggleSoftInput(android.view.inputmethod.InputMethodMana
ger.SHOW_FORCED, 0); }
              if (isDark) {
                 search_icon.setImageResource(R.drawable.default_image);
              }
              else {
                 search_icon.setImageResource(R.drawable.close_black);
              }
              if (timer != null) {
                 timer.cancel();
              }
              fabAnimator("alpha", 0, 200);
              fabAnimator("translationY", 30, 200);
           }
        }
     });

     listview.setOnItemLongClickListener(new
```

```
AdapterView.OnItemLongClickListener() {
         @Override
         public boolean onItemLongClick(AdapterView<?> param1, View param2,
int param3, long param4) {
            final int position = param3;

            return true;
         }
      });

      cancel.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            isLeftBar = false;
            openCloseLeftBar(isLeftBar);
         }
      });

      features.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            toNotes.setAction(Intent.ACTION_VIEW);
            toNotes.setClass(getApplicationContext(),
MealsListActivity.class);
            toNotes.putExtra("Data", new Gson().toJson(Notes_List));
            toNotes.putExtra("Type", "All");
            startActivity(toNotes);
         }
      });

      darker.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            if (isDark) {
               isDark = false;
               data.edit().putString("theme", "white").commit();
               UI(data.getString("accent", ""), isDark);
            }
            else {
               isDark = true;
               data.edit().putString("theme", "dark").commit();
               UI(data.getString("accent", ""), isDark);
            }
            ((BaseAdapter)listview.getAdapter()).notifyDataSetChanged();
         }
      });

      fab.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            data.edit().putString("viewIndex",
String.valueOf((long)(listview.getFirstVisiblePosition()))).commit();
            final View vi = listview.getChildAt(0);
            data.edit().putString("viewTop", String.valueOf((long)((vi ==
null) ? 0 : (vi.getTop() - listview.getPaddingTop())))).commit();
            android.view.inputmethod.InputMethodManager imm =
(android.view.inputmethod.InputMethodManager)
getSystemService(Activity.INPUT_METHOD_SERVICE);
imm.hideSoftInputFromWindow(search.getWindowToken(), 0);
            intent.setClass(getApplicationContext(),
WeightEditorActivity.class);
```

```
            intent.putExtra("activity", "create");
            intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
            startActivity(intent);
         }
      });
   }

   private void initializeLogic() {
      startApp();
      txtRegWeight.setText("Current Weight Registered:
".concat(userdetail.getString("weight", "")));
      if (data.getString("accent", "").equals("")) {
         data.edit().putString("accent", "#008dcd").commit();
      }
      else {

      }
      if (data.getString("theme", "").equals("")) {
         data.edit().putString("theme", "white").commit();
         UI(data.getString("accent", ""), false);
         isDark = false;
      }
      else {
         if (data.getString("theme", "").equals("dark")) {
            UI(data.getString("accent", ""), true);
            isDark = true;
         }
         else {
            UI(data.getString("accent", ""), false);
            isDark = false;
         }
      }
      txtRegWeight.setBackground(new GradientDrawable() { public
GradientDrawable getIns(int a, int b) {
         this.setCornerRadius(a); this.setColor(b); return this; }
}.getIns((int)10, 0xFF));
      mask.setTranslationX((float)((0 -
ToastUtil.getDisplayWidthPixels(getApplicationContext())) -
ToastUtil.getDip(getApplicationContext(),
            (int)(61))));
      Animator(mask, "alpha", 0.0d, 200);
      left_bar.setTranslationZ(5);
      listview.setFastScrollEnabled(false);
      ONCREATE_LIST();
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
      super.onActivityResult(requestCode, resultCode, data);
      switch (requestCode) {

         default:
         break;
      }
   }

   @Override
   public void onStart() {
      super.onStart();
      if (isSearch) {
```

```
         getAllData(list);
         if (search.getText().toString().length() > 0) {
            android.view.inputmethod.InputMethodManager inputMethodManager
= (android.view.inputmethod.InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
            if (inputMethodManager != null) {
inputMethodManager.toggleSoftInput(android.view.inputmethod.InputMethodMana
ger.SHOW_FORCED, 0); }
            n = list.size() - 1;
            m = list.size();
            for(int i = 0; i < (int)(m); i++) {
               if
(list.get((int)n).get("title").toString().trim().toLowerCase().contains(sea
rch.getText().toString().trim().toLowerCase())) {

               }
               else {
                  list.remove((int)(n));
               }
               n--;
            }
         }
         refresh();
      }
      else {
         getAllData(list);
         refresh();
      }
      if (data.getString("list", "").equals("")) {
         listview.setVisibility(View.GONE);
         empty_layout.setVisibility(View.VISIBLE);
      }
      else {
         listview.setVisibility(View.VISIBLE);
         empty_layout.setVisibility(View.GONE);
      }
   }

   @Override
   public void onBackPressed() {
      transitionManager(action, 100);
      fabAnimator("alpha", 1, 200);
      fabAnimator("translationY", 0, 200);
      if (isLeftBar) {
         isLeftBar = false;
         openCloseLeftBar(isLeftBar);
      }
      else {
         if (isSearch) {
            isSearch = false;
            isSearch = false;
            title.setVisibility(View.VISIBLE);
            search.setVisibility(View.GONE);
            menu.setVisibility(View.VISIBLE);
            if (isDark) {
               search_icon.setImageResource(R.drawable.default_image);
            }
            else {
               search_icon.setImageResource(R.drawable.search_black);
            }
            search.setText("");
```

```
            android.view.inputmethod.InputMethodManager imm =
(android.view.inputmethod.InputMethodManager)
getSystemService(Activity.INPUT_METHOD_SERVICE);
            imm.hideSoftInputFromWindow(search.getWindowToken(), 0);
         }
         else {
            finish();
         }
      }
   }

   @Override
   public void onPause() {
      super.onPause();
      data.edit().putString("viewIndex",
String.valueOf((long)(listview.getFirstVisiblePosition()))).commit();
      final View vi = listview.getChildAt(0);
      data.edit().putString("viewTop", String.valueOf((long)((vi == null) ?
0 : (vi.getTop() - listview.getPaddingTop())))).commit();
   }
   public void startApp () {
      search.setVisibility(View.GONE);
      left_bar.setTranslationX((float)((-
ToastUtil.getDisplayWidthPixels(getApplicationContext())) -
ToastUtil.getDip(getApplicationContext(), (int)(61))));
      LayoutParams(mask,
ToastUtil.getDisplayWidthPixels(getApplicationContext()),
ToastUtil.getDisplayHeightPixels(getApplicationContext()));
      mask.setTranslationX((float)((-
ToastUtil.getDisplayWidthPixels(getApplicationContext())) -
ToastUtil.getDip(getApplicationContext(), (int)(61))));
      mask.setVisibility(View.GONE);
      mask.setAlpha((float)(0));
      listview.setFastScrollEnabled(false);
   }


   public void UI (final String accent, final boolean dark) {
      if (dark) {
         colorPrimary = "#202125";
         colorPrimaryDark = "#1a1a1e";
         if (isLeftBar) {
            View decor = getWindow().getDecorView();
            decor.setSystemUiVisibility(0);

getWindow().setStatusBarColor(Color.parseColor(colorPrimaryDark));

getWindow().setNavigationBarColor(Color.parseColor(colorPrimaryDark));
         }
         else {
            View decor = getWindow().getDecorView();
            decor.setSystemUiVisibility(0);
            getWindow().setStatusBarColor(Color.parseColor(colorPrimary));

getWindow().setNavigationBarColor(Color.parseColor(colorPrimary));
         }
         setBackground(main, 0, 0, colorPrimary, false);
         gradientDrawable(action, 0, 0, 10, colorPrimary, colorPrimary,
false, false, 0);
         gradientDrawable(search_icon, 50, 0, 0, colorPrimary,
colorPrimary, true, true, 200);
```

```
            corners(left_bar, colorPrimary, colorPrimary, 0, 0, 30, 30, 0);
            gradientDrawable(cancel, 50, 0, 0, colorPrimary, colorPrimary,
true, true, 200);
            gradientDrawable(features, 50, 0, 0, colorPrimary, colorPrimary,
true, true, 200);
            gradientDrawable(plugins, 50, 0, 0, colorPrimary, colorPrimary,
true, true, 200);
            gradientDrawable(settings, 50, 0, 0, colorPrimary, colorPrimary,
true, true, 200);
            gradientDrawable(darker, 50, 0, 0, colorPrimary, colorPrimary,
true, true, 200);
            gradientDrawable(menu, 50, 0, 0, colorPrimary, colorPrimary, true,
true, 200);
            menu.setImageResource(R.drawable.menu_white);
            search_icon.setImageResource(R.drawable.search_white);
            cancel.setImageResource(R.drawable.backward_white);
            features.setImageResource(R.drawable.circle_white);
            plugins.setImageResource(R.drawable.plugin_white);
            settings.setImageResource(R.drawable.settings_white);
            darker.setImageResource(R.drawable.light_mode_white);
            title.setTextColor(0xFFF5F5F5);
            search.setTextColor(0xFFF5F5F5);
            search.setHintTextColor(Color.parseColor("#30"));
         }
         else {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_
STATUS_BAR); }
            colorPrimary = "#3F51B6";
            colorPrimaryDark = "#f0f0f0";
            if (isLeftBar) {
               getWindow().setStatusBarColor(Color.parseColor("#cccccc"));
               getWindow().setNavigationBarColor(Color.parseColor("#cccccc"));
            }
            else {
               getWindow().setStatusBarColor(Color.parseColor(colorPrimary));

getWindow().setNavigationBarColor(Color.parseColor(colorPrimary));
            }

            gradientDrawable(action, 0, 0, 10, colorPrimary, colorPrimary,
false, false, 0);
            gradientDrawable(search_icon, 50, 0, 0, colorPrimary,
colorPrimary, true, true, 200);
            corners(left_bar, colorPrimary, colorPrimary, 0, 0, 30, 30, 0);
            gradientDrawable(cancel, 50, 0, 0, colorPrimary, colorPrimary,
true, true, 200);
            gradientDrawable(features, 50, 0, 0, colorPrimary, colorPrimary,
true, true, 200);
            gradientDrawable(plugins, 50, 0, 0, colorPrimary, colorPrimary,
true, true, 200);
            gradientDrawable(settings, 50, 0, 0, colorPrimary, colorPrimary,
true, true, 200);
            gradientDrawable(darker, 50, 0, 0, colorPrimary, colorPrimary,
true, true, 200);
            gradientDrawable(menu, 50, 0, 0, colorPrimary, colorPrimary, true,
true, 200);
            menu.setImageResource(R.drawable.menu_black);
            search_icon.setImageResource(R.drawable.search_black);
            cancel.setImageResource(R.drawable.backward);
            features.setImageResource(R.drawable.dinner);
```

```
        plugins.setImageResource(R.drawable.fast_food);
        settings.setImageResource(R.drawable.todo);
        darker.setImageResource(R.drawable.dark_mode_black);
        title.setTextColor(0xFF424242);
        search.setTextColor(0xFF424242);
        search.setHintTextColor(Color.parseColor("#30000000"));
      }
   }


   public void gradientDrawable (final View view, final double radius,
final double stroke, final double shadow,
                           final String color, final String _borderColor,
final boolean ripple,
                           final boolean _clickAnim, final double
_animDuration) {
      if (ripple) {
         android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
         gd.setColor(Color.parseColor(color));
         gd.setCornerRadius((int)radius);
         gd.setStroke((int)stroke,Color.parseColor(_borderColor));
         if (Build.VERSION.SDK_INT >= 21){
            view.setElevation((int)shadow);}
         android.content.res.ColorStateList clrb = new
android.content.res.ColorStateList(new int[][]{new int[]{}}, new
int[]{Color.parseColor("#9e9e9e")});
         android.graphics.drawable.RippleDrawable ripdrb = new
android.graphics.drawable.RippleDrawable(clrb , gd, null);
         view.setClickable(true);
         view.setBackground(ripdrb);
      }
      else {
         android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
         gd.setColor(Color.parseColor(color));
         gd.setCornerRadius((int)radius);
         gd.setStroke((int)stroke,Color.parseColor(_borderColor));
         view.setBackground(gd);
         if (Build.VERSION.SDK_INT >= 21){
            view.setElevation((int)shadow);}
      }
      if (_clickAnim) {
         view.setOnTouchListener(new View.OnTouchListener() {
            @Override
            public boolean onTouch(View v, MotionEvent event) {
               switch (event.getAction()){
                  case MotionEvent.ACTION_DOWN:{
                     ObjectAnimator scaleX = new ObjectAnimator();
                     scaleX.setTarget(view);
                     scaleX.setPropertyName("scaleX");
                     scaleX.setFloatValues(0.9f);
                     scaleX.setDuration((int)_animDuration);
                     scaleX.start();

                     ObjectAnimator scaleY = new ObjectAnimator();
                     scaleY.setTarget(view);
                     scaleY.setPropertyName("scaleY");
                     scaleY.setFloatValues(0.9f);
                     scaleY.setDuration((int)_animDuration);
                     scaleY.start();
```

```
                    break;
                }
                case MotionEvent.ACTION_UP:{

                    ObjectAnimator scaleX = new ObjectAnimator();
                    scaleX.setTarget(view);
                    scaleX.setPropertyName("scaleX");
                    scaleX.setFloatValues((float)1);
                    scaleX.setDuration((int)_animDuration);
                    scaleX.start();

                    ObjectAnimator scaleY = new ObjectAnimator();
                    scaleY.setTarget(view);
                    scaleY.setPropertyName("scaleY");
                    scaleY.setFloatValues((float)1);
                    scaleY.setDuration((int)_animDuration);
                    scaleY.start();

                    break;
                }
            }
            return false;
        }
    });
  }
}


  public void _ItemText (final TextView textView, final double position,
final String _key, final ArrayList<HashMap<String, Object>> _list) {
    if (list.get((int)position).containsKey(_key)) {
      textView.setText(_list.get((int)position).get(_key).toString());
    }
  }


  public void getAllData (final ArrayList<HashMap<String, Object>> _list)
{
    if (!data.getString("list", "").equals("")) {
      list = new Gson().fromJson(data.getString("list", ""), new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
    }
  }


  public void setBackground (final View view, final double radius, final
double msghadow, final String color, final boolean ripple) {
    if (ripple) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
      gd.setColor(Color.parseColor(color));
      gd.setCornerRadius((int)radius);
      view.setElevation((int)msghadow);
      android.content.res.ColorStateList clrb = new
android.content.res.ColorStateList(new int[][]{new int[]{}}, new
int[]{Color.parseColor("#9e9e9e")});
      android.graphics.drawable.RippleDrawable ripdrb = new
android.graphics.drawable.RippleDrawable(clrb , gd, null);
      view.setClickable(true);
      view.setBackground(ripdrb);
    }
```

```
      else {
         android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();
         gd.setColor(Color.parseColor(color));
         gd.setCornerRadius((int)radius);
         view.setBackground(gd);
         view.setElevation((int)msghadow);
      }
   }


   public void openCloseLeftBar (final boolean _open) {
      if (_open) {
         Animator(left_bar, "translationX", 0 -
ToastUtil.getDisplayWidthPixels(getApplicationContext()), 200);
         mask.setEnabled(true);
         Animator(mask, "alpha", 0.2d, 200);
         mask.setVisibility(View.VISIBLE);
         if (isDark) {
             @android.annotation.SuppressLint("RestrictedApi")
                    android.animation.ValueAnimator colorAnimation =
android.animation.ValueAnimator.ofObject(
                             new ArgbEvaluator(),
Color.parseColor("#202125"), Color.parseColor("#1a1a1e"));
                    colorAnimation.addUpdateListener(new
android.animation.ValueAnimator.AnimatorUpdateListener() {
                            @Override
                            public void
onAnimationUpdate(android.animation.ValueAnimator animator) {
                                    int color = (int)
animator.getAnimatedValue();
                                    if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.LOLLIPOP) {

getWindow().setStatusBarColor(color);
                     getWindow().setNavigationBarColor(color);
                                        }
                                }

                        });
                    colorAnimation.setDuration(200);
                    colorAnimation.start();



         }
         else {
             @android.annotation.SuppressLint("RestrictedApi")
                    android.animation.ValueAnimator colorAnimation =
android.animation.ValueAnimator.ofObject(
                             new ArgbEvaluator(),
Color.parseColor("#ffffff"), Color.parseColor("#cccccc"));
                    colorAnimation.addUpdateListener(new
android.animation.ValueAnimator.AnimatorUpdateListener() {
                            @Override
                            public void
onAnimationUpdate(android.animation.ValueAnimator animator) {
                                    int color = (int)
animator.getAnimatedValue();
                                    if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.LOLLIPOP) {
```

```
getWindow().setStatusBarColor(color);
                    getWindow().setNavigationBarColor(color);
                                }
                            }

                        });
                    colorAnimation.setDuration(200);
                    colorAnimation.start();



            }
            fabAnimator("alpha", 0, 200);
            fabAnimator("translationY", 30, 200);
        }
        else {
            Animator(left_bar, "translationX", (0 -
ToastUtil.getDisplayWidthPixels(getApplicationContext())) -
ToastUtil.getDip(getApplicationContext(), (int)(61)), 200);
            mask.setEnabled(false);
            Animator(mask, "alpha", 0, 200);
            _timer = new TimerTask() {
                @Override
                public void run() {
                    runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            mask.setVisibility(View.GONE);
                        }
                    });
                }
            };
            timer.schedule(_timer, (int)(200));
            if (isDark) {
                @android.annotation.SuppressLint("RestrictedApi")
                    android.animation.ValueAnimator colorAnimation =
android.animation.ValueAnimator.ofObject(
                            new ArgbEvaluator(),
Color.parseColor("#1a1a1e"), Color.parseColor("#202125"));
                    colorAnimation.addUpdateListener(new
android.animation.ValueAnimator.AnimatorUpdateListener() {
                        @Override
                        public void
onAnimationUpdate(android.animation.ValueAnimator animator) {
                            int color = (int)
animator.getAnimatedValue();
                            if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.LOLLIPOP) {

getWindow().setStatusBarColor(color);
                    getWindow().setNavigationBarColor(color);
                                }
                            }

                        });
                    colorAnimation.setDuration(200);
                    colorAnimation.start();
```

```
        }
        else {
            @android.annotation.SuppressLint("RestrictedApi")
                    android.animation.ValueAnimator colorAnimation =
android.animation.ValueAnimator.ofObject(
                             new ArgbEvaluator(),
Color.parseColor("#cccccc"), Color.parseColor("#ffffff"));
                    colorAnimation.addUpdateListener(new
android.animation.ValueAnimator.AnimatorUpdateListener() {
                            @Override
                            public void
onAnimationUpdate(android.animation.ValueAnimator animator) {
                                    int color = (int)
animator.getAnimatedValue();
                                    if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.LOLLIPOP) {

getWindow().setStatusBarColor(color);
                     getWindow().setNavigationBarColor(color);
                                        }
                                }

                       });
                    colorAnimation.setDuration(200);
                    colorAnimation.start();



        }
        fabAnimator("alpha", 1, 200);
        fabAnimator("translationY", 0, 200);
      }
   }


   public void Animator (final View view, final String _propertyName, final
double value, final double duration) {
      ObjectAnimator anim = new ObjectAnimator();
      anim.setTarget(view);
      anim.setPropertyName(_propertyName);
      anim.setFloatValues((float)value);
      anim.setDuration((long)duration);
      anim.start();
   }


   public void corners (final View view, final String _color1, final String
_color2, final double msgtr, final double _n1, final double _n2, final
double _n3, final double _n4) {
      android.graphics.drawable.GradientDrawable gd = new
android.graphics.drawable.GradientDrawable();

      gd.setColor(Color.parseColor(_color1));

      gd.setStroke((int)msgtr, Color.parseColor(_color2));

      gd.setCornerRadii(new
float[]{(int)_n1,(int)_n1,(int)_n2,(int)_n2,(int)_n3,(int)_n3,(int)_n4,(int
)_n4});

      view.setBackground(gd);
```

```
      view.setElevation(2);
   }


   public void LayoutParams (final View view, final double _width, final
double _height) {
      view.setLayoutParams(new LinearLayout.LayoutParams((int)_width,(int)
_height));
   }


   public void statusbarColorAnim (final String _color1, final String
_color2, final double duration) {
       @android.annotation.SuppressLint("RestrictedApi")
              android.animation.ValueAnimator colorAnimation =
android.animation.ValueAnimator.ofObject(
                      new ArgbEvaluator(), Color.parseColor(_color1),
Color.parseColor(_color2));
              colorAnimation.addUpdateListener(new
android.animation.ValueAnimator.AnimatorUpdateListener() {
                     @Override
                     public void
onAnimationUpdate(android.animation.ValueAnimator animator) {
                            int color = (int) animator.getAnimatedValue();
                            if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.LOLLIPOP) {
                                   getWindow().setStatusBarColor(color);
                                }
                         }

                });
              colorAnimation.setDuration((int)duration);
              colorAnimation.start();
   }


   public void _classMe () {

   }


   public void transitionManager (final View view, final double duration) {
      LinearLayout viewgroup = (LinearLayout) view;

      android.transition.AutoTransition autoTransition = new
android.transition.AutoTransition();
      autoTransition.setDuration((long)duration);

android.transition.TransitionManager.beginDelayedTransition(viewgroup,
autoTransition);
   }


   @RequiresApi(api = Build.VERSION_CODES.O)
   public void custDialog (final String _Title, final String msgubtitle,
final String _Image) {


      final AlertDialog dia = new
AlertDialog.Builder(WeightLogsActivity.this).create();
```

```
      View inflate = getLayoutInflater().inflate(R.layout.dialog_default,
null);
      if (true) {


dia.getWindow().clearFlags(WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE|Wi
ndowManager.LayoutParams.FLAG_ALT_FOCUSABLE_IM);

dia.getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STAT
E_ALWAYS_HIDDEN);

dia.getWindow().setBackgroundDrawableResource(android.R.color.transparent);
         dia.setContentView(R.layout.dialog_default);
         dia.getWindow().setDimAmount( 0.5f );
         dia.getWindow().setGravity( Gravity.CENTER);
         dia.setCanceledOnTouchOutside(true);
         dia.setCancelable(true);
         dia.setView(inflate);
         dia.show();
      }
      if (true) {

         android.graphics.drawable.GradientDrawable gd_linear1 = new
android.graphics.drawable.GradientDrawable();
         LinearLayout cust_linear1 = inflate.findViewById(R.id.linear1);
         gd_linear1.setColor(Color.parseColor("#C0C0C0"));
         gd_linear1.setCornerRadius(40);
         gd_linear1.setStroke(5,Color.parseColor("#000000"));
         cust_linear1.setBackground(gd_linear1);
      }
      if (true) {

         TextView cust_textview1 = (TextView)
inflate.findViewById(R.id.title);
         android.graphics.drawable.GradientDrawable gd_textview1 = new
android.graphics.drawable.GradientDrawable();
         gd_textview1.setStroke(5,Color.parseColor("#00ff00"));
         gd_textview1.setColor(Color.parseColor("#B666DE"));
         gd_textview1.setCornerRadius(20);
         cust_textview1.setElevation(16);
         cust_textview1.setBackground(gd_textview1);
         cust_textview1.setText(_Title);
         cust_textview1.setTextColor(Color.BLACK);
         cust_textview1.setShadowLayer(10,5,5,Color.parseColor("#999666"));

cust_textview1.setJustificationMode(LineBreaker.JUSTIFICATION_MODE_INTER_WO
RD);
         cust_textview1.setTextSize(18);
      }
      if (true) {

         TextView cust_textview2 = (TextView)
inflate.findViewById(R.id.message);
         android.graphics.drawable.GradientDrawable gd_textview2 = new
android.graphics.drawable.GradientDrawable();
         gd_textview2.setStroke(5,Color.parseColor("#ffff00"));
         gd_textview2.setColor(Color.parseColor("#B0C4ff"));
         gd_textview2.setCornerRadius(20);
         cust_textview2.setElevation(16);
         cust_textview2.setBackground(gd_textview2);
         cust_textview2.setText(msgubtitle);
```

```
        cust_textview2.setTextColor(Color.BLACK);
        cust_textview2.setShadowLayer(10,5,5,Color.parseColor("#999666"));

cust_textview2.setJustificationMode(LineBreaker.JUSTIFICATION_MODE_INTER_WO
RD);
        cust_textview2.setTextSize(12);
      }
      if (true) {

        ImageView cust_imageview1 =  inflate.findViewById (R.id.image);

cust_imageview1.setImageResource(getResources().getIdentifier(_Image,
"drawable", getPackageName()));
        LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(100, 100);
        cust_imageview1.setLayoutParams(layoutParams);
        cust_imageview1.requestLayout();
        cust_imageview1.setScaleType(ImageView.ScaleType.CENTER_CROP);
      }
      if (true) {

        TextView cust_button1 = (TextView)
inflate.findViewById(R.id.apply);
        cust_button1.setText("OK");
        cust_button1.setShadowLayer(10,5,5,Color.parseColor("#999666"));
        cust_button1.setOnClickListener(new OnClickListener() { public
void onClick(View view) {
              dia.dismiss();

            }});
      }
   }


   public void refresh () {
      listview.setAdapter(new ListviewAdapter(list));
      ((BaseAdapter)listview.getAdapter()).notifyDataSetChanged();
      if (!data.getString("viewIndex", "").equals("")) {
        viewIndex = Double.parseDouble(data.getString("viewIndex", ""));
        viewTop = Double.parseDouble(data.getString("viewTop", ""));
        listview.setSelectionFromTop((int)viewIndex,(int) viewTop);
      }
   }


   public void fabAnimator (final String _propertyName, final double value,
final double duration) {
      ObjectAnimator anim = new ObjectAnimator();
      anim.setTarget(fab);
      anim.setPropertyName(_propertyName);
      anim.setFloatValues((float)value);
      anim.setDuration((long)duration);
      anim.start();
   }


   public void ONCREATE_LIST () {
      if (!"".equals("")) {
        Notes_List = new Gson().fromJson("", new
TypeToken<ArrayList<HashMap<String, Object>>>(){}.getType());
      }
```

```
    }


    public class ListviewAdapter extends BaseAdapter {
        ArrayList<HashMap<String, Object>> data;
        public ListviewAdapter(ArrayList<HashMap<String, Object>> arr) {
            data = arr;
        }

        @Override
        public int getCount() {
            return data.size();
        }

        @Override
        public HashMap<String, Object> getItem(int index) {
            return data.get(index);
        }

        @Override
        public long getItemId(int index) {
            return index;
        }
        @Override
        public View getView(final int position, View v, ViewGroup container)
{
            LayoutInflater inflater =
(LayoutInflater)getBaseContext().getSystemService(Context.LAYOUT_INFLATER_S
ERVICE);
            View view = v;
            if (view == null) {
                view = inflater.inflate(R.layout.custom, null);
            }

            final LinearLayout background =
view.findViewById(R.id.background);
            final TextView textview1 = (TextView)
view.findViewById(R.id.textview1);
            final TextView textview3 = (TextView)
view.findViewById(R.id.textview3);
            final TextView textview2 = (TextView)
view.findViewById(R.id.textview2);

            _ItemText(textview1, position, "title", list);
            _ItemText(textview2, position, "text", list);
            if (isDark) {
                setBackground(background, 0, 0, "#202125", true);
                textview1.setTextColor(0xFFFAFAFA);
            }
            else {
                setBackground(background, 0, 0, "#ffffff", true);
                textview1.setTextColor(0xFF616161);
            }
            background.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                android.view.inputmethod.InputMethodManager imm =
(android.view.inputmethod.InputMethodManager)
getSystemService(Activity.INPUT_METHOD_SERVICE);
                imm.hideSoftInputFromWindow(search.getWindowToken(), 0);
                   intent.setClass(getApplicationContext(),
```

```
ViewWeightActivity.class);
                intent.putExtra("id",
data.get((int)position).get("id").toString());
                intent.putExtra("type",
data.get((int)position).get("type").toString());
                intent.putExtra("title",
data.get((int)position).get("title").toString());
                intent.putExtra("text",
data.get((int)position).get("text").toString());
                intent.putExtra("date",
data.get((int)position).get("date").toString());
                intent.putExtra("updatedDate",
data.get((int)position).get("updatedDate").toString());
                intent.addFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
                startActivity(intent);
            }
         });
         if (true) {
            background.setOnLongClickListener(new OnLongClickListener() {
@Override public boolean onLongClick(View v) {
                  if (true) {
                     final AlertDialog.Builder alert = new
AlertDialog.Builder(WeightLogsActivity.this);
                     View view =
getLayoutInflater().inflate(R.layout.dialog_default,null);
                     TextView apply =
(TextView)view.findViewById(R.id.apply);
                     TextView cancel =
(TextView)view.findViewById(R.id.cancel);
                     TextView title =
(TextView)view.findViewById(R.id.title);
                     TextView message =
(TextView)view.findViewById(R.id.message);
                     ImageView image = view.findViewById(R.id.image);
                     LinearLayout parent = view.findViewById(R.id.parent);
                     alert.setView(view);
                     final AlertDialog dialog = alert.create();

setBackground(parent,(double)50,(double)5,colorPrimary,false);

setBackground(apply,(double)50,(double)0,colorPrimary,true);

setBackground(cancel,(double)50,(double)0,colorPrimary,true);
                     if (true) {
                        ObjectAnimator scaleX = new ObjectAnimator();
                        scaleX.setTarget(parent);
                        scaleX.setPropertyName("scaleX");
                        scaleX.setFloatValues((float)0.8f,(float)1f);
                        scaleX.setDuration(300);
                        scaleX.setInterpolator(new
AccelerateDecelerateInterpolator());
                        scaleX.start();
                        ObjectAnimator scaleY = new ObjectAnimator();
                        scaleY.setTarget(parent);
                        scaleY.setPropertyName("scaleY");
                        scaleY.setFloatValues((float)0.8f,(float)1f);
                        scaleY.setDuration(300);
                        scaleY.setInterpolator(new
AccelerateDecelerateInterpolator());
                        scaleY.start();
                        ObjectAnimator alpha = new ObjectAnimator();
```

```
                        alpha.setTarget(parent);
                        alpha.setPropertyName("alpha");
                        alpha.setFloatValues((float)0f,(float)1);
                        alpha.setDuration(300);
                        alpha.setInterpolator(new
AccelerateDecelerateInterpolator());
                        alpha.start();
                     }
                     if (true) {
                        title.setText("DESTROY");
                        message.setText("You want to destroy this log");
                        apply.setText("DESTROY");
                        cancel.setText("CANCEL");
                     }
                     if (isDark) {
                        apply.setTextColor(Color.parseColor("#ffffff"));
                        cancel.setTextColor(Color.parseColor("#ffffff"));
                        title.setTextColor(Color.parseColor("#ffffff"));
                        message.setTextColor(Color.parseColor("#ffffff"));
                        image.setImageResource(R.drawable.delete_white);
                     }
                     else {
                        apply.setTextColor(Color.parseColor("#424242"));
                        cancel.setTextColor(Color.parseColor("#424242"));
                        title.setTextColor(Color.parseColor("#424242"));
                        message.setTextColor(Color.parseColor("#424242"));
                        image.setImageResource(R.drawable.delete_black);
                     }
                     dialog.getWindow().setBackgroundDrawable(new
android.graphics.drawable.ColorDrawable(android.graphics.Color.TRANSPARENT)
);
                     apply.setOnClickListener(new View.OnClickListener() {
                        @Override public void onClick(View v) {
                           if (true) {
                              list.remove((int)(position));
                              refresh();

ToastUtil.showMessage(getApplicationContext(), "Removed");
                           }
                           dialog.dismiss();}});
                     cancel.setOnClickListener(new View.OnClickListener() {
                        @Override public void onClick(View v) {
                           if (true) {

                           }
                           dialog.dismiss();}});
                     dialog.show();
                  }
                  return true; } });
            }

            return view;
         }
      }

      @Deprecated
      public void showMessage(String msg) {
         Toast.makeText(getApplicationContext(), msg,
Toast.LENGTH_SHORT).show();
      }
```

```
    @Deprecated
    public int getLocationX(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[0];
    }

    @Deprecated
    public int getLocationY(View v) {
        int location[] = new int[2];
        v.getLocationInWindow(location);
        return location[1];
    }

    @Deprecated
    public int getRandom(int min, int max) {
        Random random = new Random();
        return random.nextInt(max - min + 1) + min;
    }

    @Deprecated
    public ArrayList<Double> getCheckedItemPositionsToArray(ListView _list)
{
        ArrayList<Double> result = new ArrayList<Double>();
        SparseBooleanArray arr = _list.getCheckedItemPositions();
        for (int iIdx = 0; iIdx < arr.size(); iIdx++) {
            if (arr.valueAt(iIdx))
            result.add((double)arr.keyAt(iIdx));
        }
        return result;
    }

    @Deprecated
    public float getDip(int input) {
        return TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, input,
getResources().getDisplayMetrics());
    }

    @Deprecated
    public int getDisplayWidthPixels() {
        return getResources().getDisplayMetrics().widthPixels;
    }

    @Deprecated
    public int getDisplayHeightPixels() {
        return getResources().getDisplayMetrics().heightPixels;
    }
}
```