INSTITUTE of TECHNOLOGY CARLOW
Institiúid Teicneolaíochta Cheatharlach

Software Development Final Project

Driving Companion App

Design Document

Student: Chi Ieong Ng C00223421

Supervisor: Hisain Elshaafi

Submission Date: 17/12/2021

Table of Content

1. Introduction

This document is part of my final year project in ITCarlow Software Development Year 4. I decided to develop a mobile app called the "Driving companion app". The app will score and give feedback to drivers by monitoring their driving behavior.

Driving Companion App is a mobile phone application that analyses the user's driving behavior, obtains data from the monitoring process, and gives scores and feedback to improve driving behavior.

This document focuses on the design aspects of this application. The content will talk about the technology used to design this application, the sketches of each design screen, and the flow of functions. And application structure and so on. The details about the design of this application will bring a clear direction and a better understanding of future development.

2. Technologies

This section is to revisit the technology that will be used in the development of this application:

2.1 Flutter

Flutter is an open-source mobile application software development kit developed by Google. Its engine is mainly written in C++ and provides low-level rendering support using Google's Skia graphics library. Android and IOS also provide Platform-specific SDKs. It can use one programming language (Dart) to create cross-platform applications with Android and iOS and supported Web and desktop applications.

Flutter is a widget-based technology. It can apply object-oriented programming to any element and easily modify or customize widgets.

Flutter has numerous advantages over its competitors. These advantages are inherent in programming languages and development toolsets, allowing Flutter to solve problems that other languages cannot.

- One codebase for all platforms

- Widgets offer countless possibilities
- Rich libraries
- Fast testing with hot reload

Now Flutter is widely used to create apps for Alibaba, Yandex, Airbnb, Uber, eBay and other leading companies. Many excellent social media apps, service booking apps, productivity measurement apps, utility apps, product distribution apps and healthcare apps, all built with Flutter.

2.2 Dart

Dart is a programming language developed by Google. It is an Object-oriented language similar to JavaScript. It can compile to either native code or javascript, and its grammatical style is close to the C language. It is a programming language very suitable for web and mobile application development. Dart can be executed on a native virtual machine, converted Dart code to JavaScript, and then performed directly on the Javascript engine. Dart can quickly use the Library provided by Google, and users can also offer their self-written Library for other developers or other projects to use.

2.3 Firebase real-time database

Firebase is a back-end cloud platform (BaaS: Backend-as-a-Service) that provides real-time database service. It is a NoSQL cloud database and supports Android, iOS and Web apps. It can save data in JSON format and synchronize to the online client in real-time. And even if the user is offline, the data is still available.

2.4 Google Maps API / Roads API

Google Maps API is a Maps programming API provided by Google for developers. It allows developers to embed Google Maps data into web apps by using Javascript. Mainly used to get location, navigation, and map service. Roads API is part of Google Maps API. It maps GPS coordinates to the road and determines speed limits and road segments. It is available via a simple HTTPS interface to expose Snap to roads, Nearest roads, and Speed limits services. And it is compatible with Java and

Python. Its libraries make development more accessible by providing simple, native implementations of everyday tasks, such as authentication, request throttling and automatic retry. However, It is not entirely free to use. Google will charge it after the primary free usage is exceeded.

3. Screen Flow

This section is the application process, starting from the authentication to the “Main Menu” and the connection relationship of each functional interface.

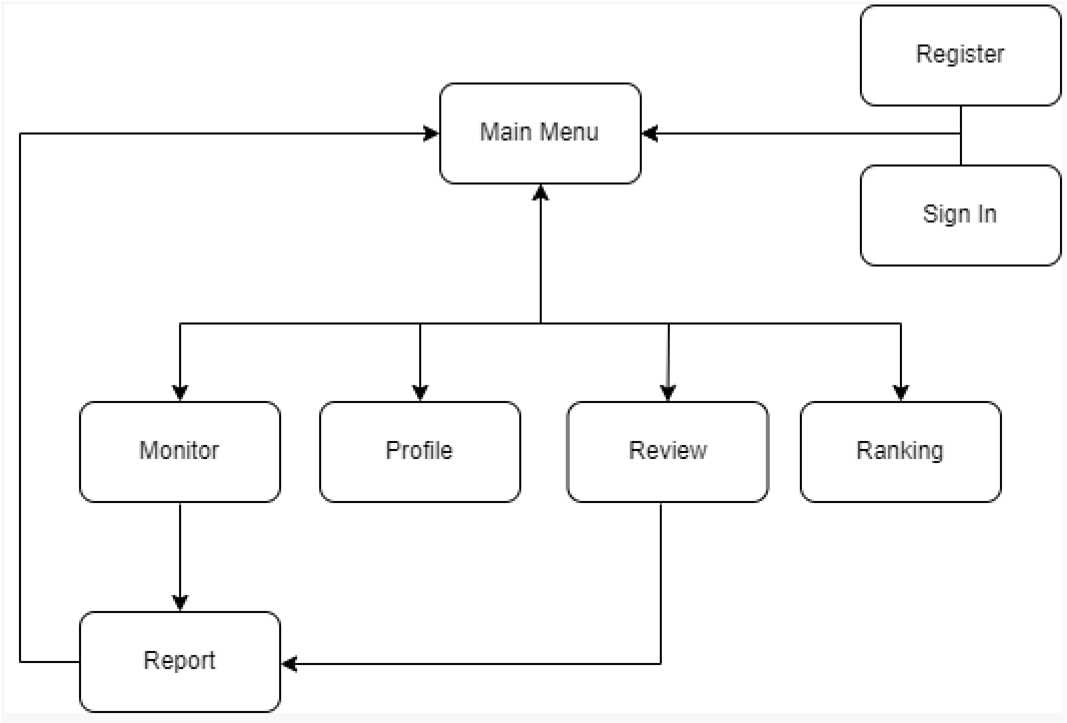


Fig 1 - Screen flow

4. Screen Sketch

This section is the screen design of the “Driving Companion Application”. The screens below are made with "Mockflow" design software, which may differ from the final product, but the developer should develop the project based on the following design drawings.

4.1 Main menu

It is the main screen of the application. The main menu is the first screen that the system automatically arrives at after the user opens the application and passes the authentication. Here are all of the main function links, namely “Monitor”, “Profile”, “Review”, and “Ranking”; The user presses these links to connect to another screen of the specified function.

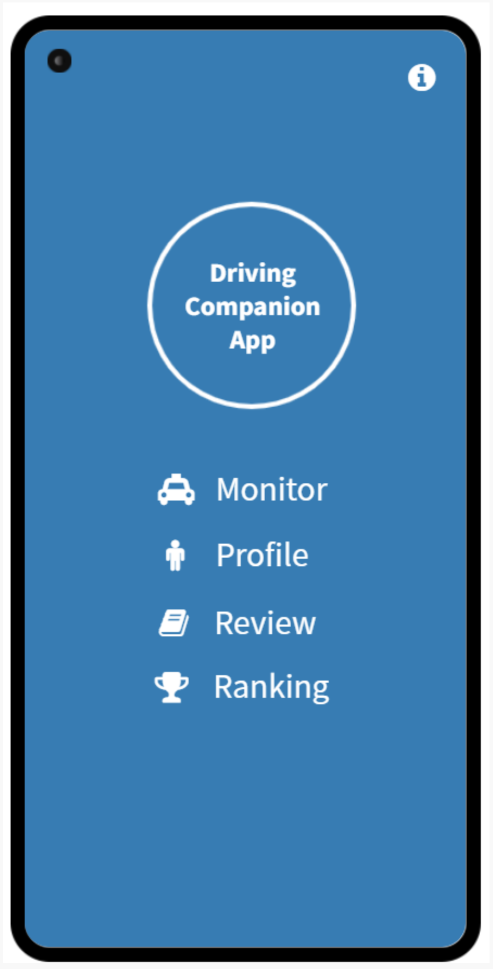


Fig 2 - Main menu screen

4.2 Monitor

It is the screen of the Monitor function. In the beginning, it has a green-style background and buttons. Initially, The application will display a real-time map and location. All data "Duration" and "Speed" are preset to zero value. The user needs to press the START button at the bottom of the screen to start the monitoring. After that, the screen will change to a red style, and the user can begin driving. The screen will start to collect and display the data during driving. When the user completes driving, press the "STOP" button to end the monitoring program.

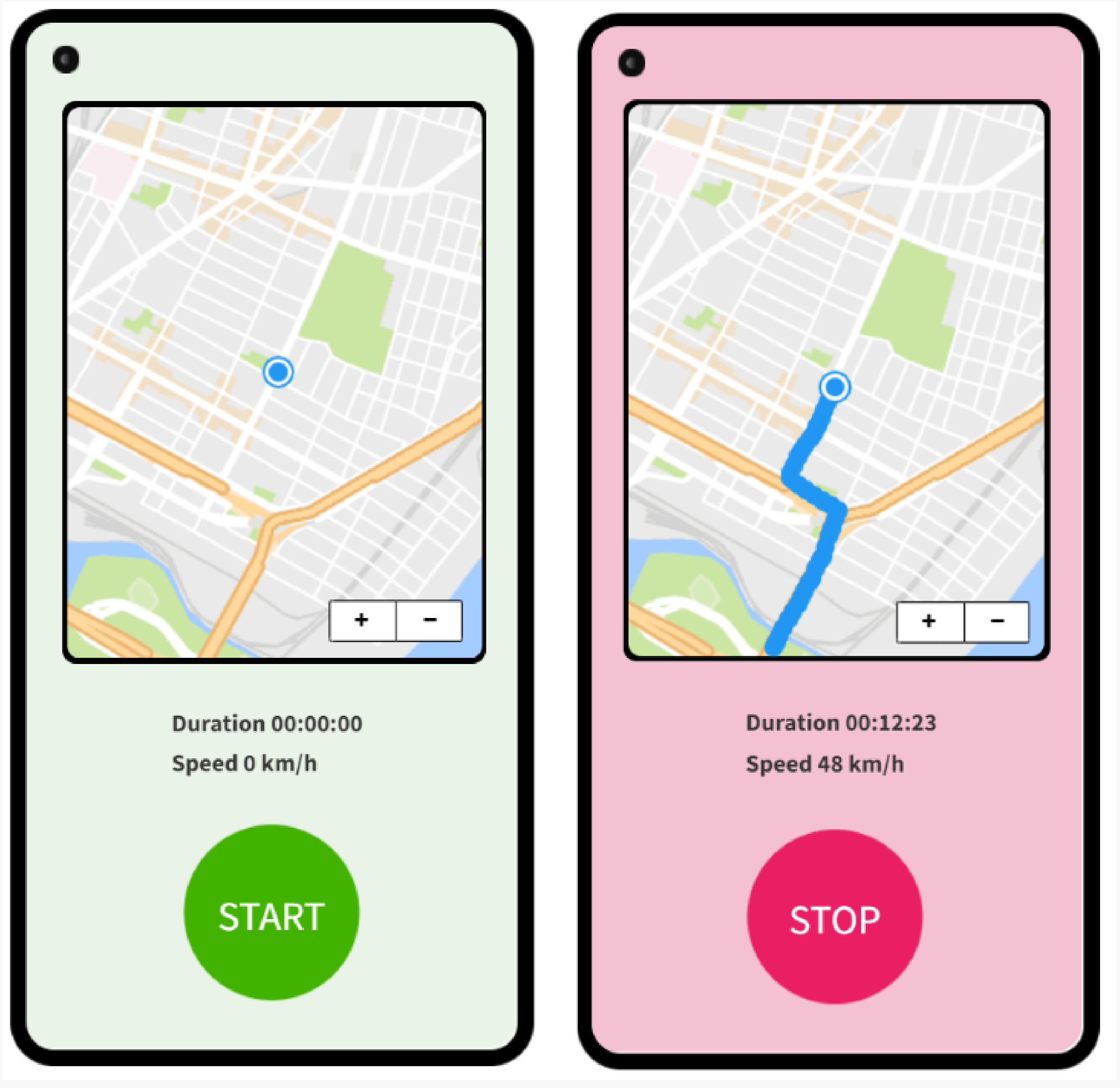
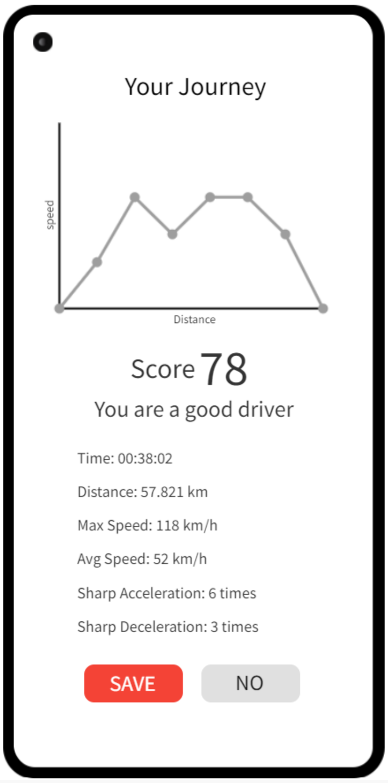


Fig 3 - Monitor screen

4.3 Report

This screen displays the results of the monitoring. The system will automatically connect to this result screen when the user ends the monitoring function. Here, the user can see the data analysis and statistics obtained from the previous driving journey, including the "speed change Linear chart", "Score", "Comment", "Time", "Distance", "Max speed", "Avg Speed", "Sharp Acceleration", and "Sharp Deceleration"; There are two buttons at the bottom of the screen: "Save" And "No" the user can choose whether to save and upload this record or leave without saving, and then the system will return to the main menu again.



Your Journey

Score 78

You are a good driver

Time: 00:38:02

Distance: 57.821 km

Max Speed: 118 km/h

Avg Speed: 52 km/h

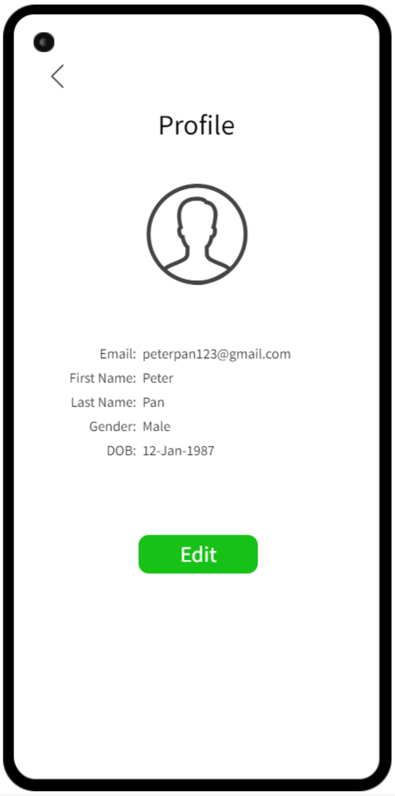
Sharp Acceleration: 6 times

Sharp Deceleration: 3 times

Fig 4 - Report screen

4.4 Profile

This screen displays the user's personal information, including profile picture, email address, First Name, Last Name, Gender, DOB, etc. Users can update their information by pressing the "Edit" button below. And the arrow in the upper left corner is the link back to the main menu



Profile

Email: peterpan123@gmail.com

First Name: Peter

Last Name: Pan

Gender: Male

DOB: 12-Jan-1987

Fig 5 - Profile screen

4.5 Register / Sign in

It is the login and registration screen. Users must sign in to continue using some functions, such as saving records, viewing personal profiles, etc. If the user is not logged in, the system will automatically jump to this login screen. The user must fill in the email address and password to log in. And New users can register a new account and then log in. After logging in, the system will automatically return to the function screen or main menu previously selected by the user. And the arrow in the upper left corner links back to the previous screen or main menu.

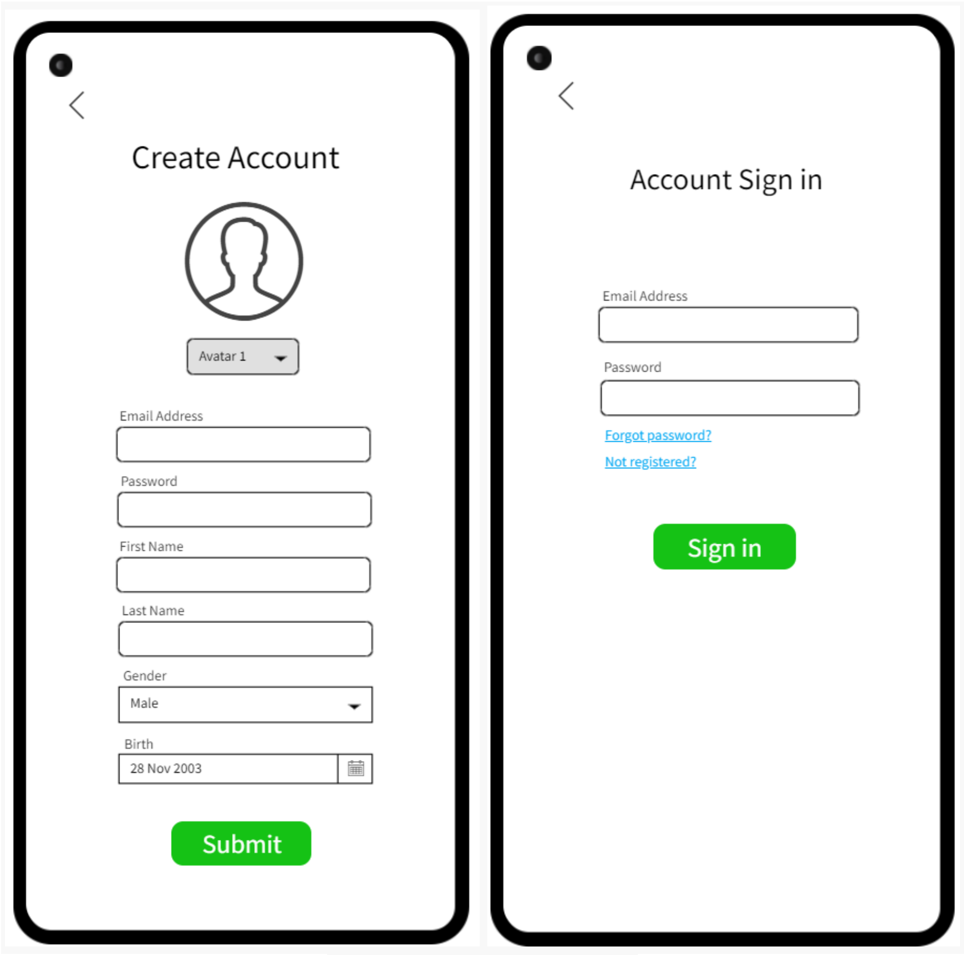
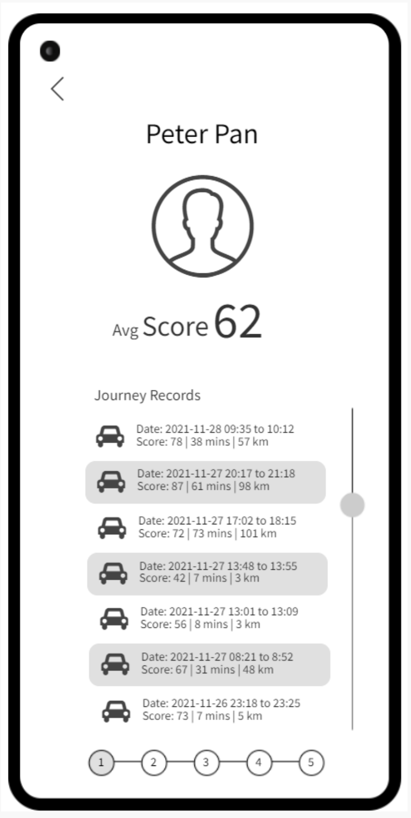


Fig 6 - Register / Sign In screens

4.6 Review

This screen allows users to review their driving records. The user’s name, avatar, and average driving score are displayed at the top. There is a drag menu at the bottom to check the previous driving records, including date and time, score, duration and distance, and the user can drag up and down to select these records and press to review the driving results of the time. And the arrow in the upper left corner links back to the main menu.



Peter Pan

Avg Score 62

Journey Records

Date: 2021-11-28 09:35 to 10:12
Score: 78 | 38 mins | 57 km

Date: 2021-11-27 20:17 to 21:18
Score: 87 | 61 mins | 98 km

Date: 2021-11-27 17:02 to 18:15
Score: 72 | 73 mins | 101 km

Date: 2021-11-27 13:48 to 13:55
Score: 42 | 7 mins | 3 km

Date: 2021-11-27 13:01 to 13:09
Score: 56 | 8 mins | 3 km

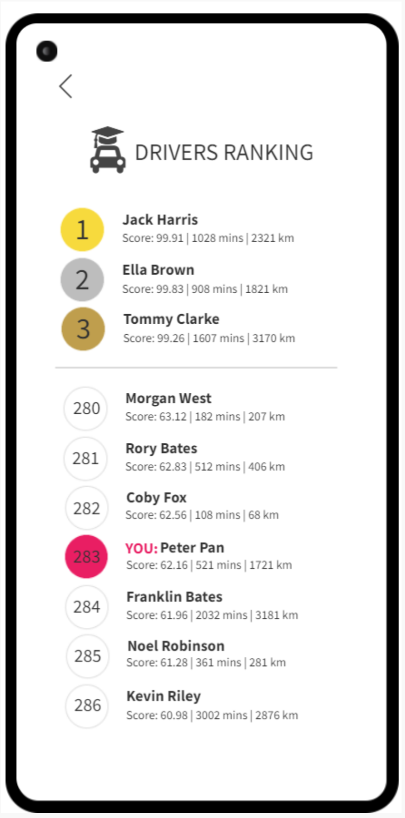
Date: 2021-11-27 08:21 to 8:52
Score: 67 | 31 mins | 48 km

Date: 2021-11-26 23:18 to 23:25
Score: 73 | 7 mins | 5 km

Fig 7 - Review screen

4.7 Ranking

This screen is a ranking list. In the top part, three drivers are shown with their "score", "total driving time", and "total driving distance", and the bottom part shows the user's ranking and the records of the three drivers near the user. The top three numbers are displayed in gold, silver and bronze, and the user's number is displayed in red, which is easy to distinguish. And the arrow in the upper left corner links back to the main menu.



DRIVERS RANKING

1 Jack Harris
Score: 99.91 | 1028 mins | 2321 km

2 Ella Brown
Score: 99.83 | 908 mins | 1821 km

3 Tommy Clarke
Score: 99.26 | 1607 mins | 3170 km

280 Morgan West
Score: 63.12 | 182 mins | 207 km

281 Rory Bates
Score: 62.83 | 512 mins | 406 km

282 Coby Fox
Score: 62.56 | 108 mins | 68 km

283 YOU: Peter Pan
Score: 62.16 | 521 mins | 1721 km

284 Franklin Bates
Score: 61.96 | 2032 mins | 3181 km

285 Noel Robinson
Score: 61.28 | 361 mins | 281 km

286 Kevin Riley
Score: 60.98 | 3002 mins | 2876 km

Fig 8 - Ranking screen

5. Class Diagram

This section is the class design of the application, their relationship, variables and functions in each class.

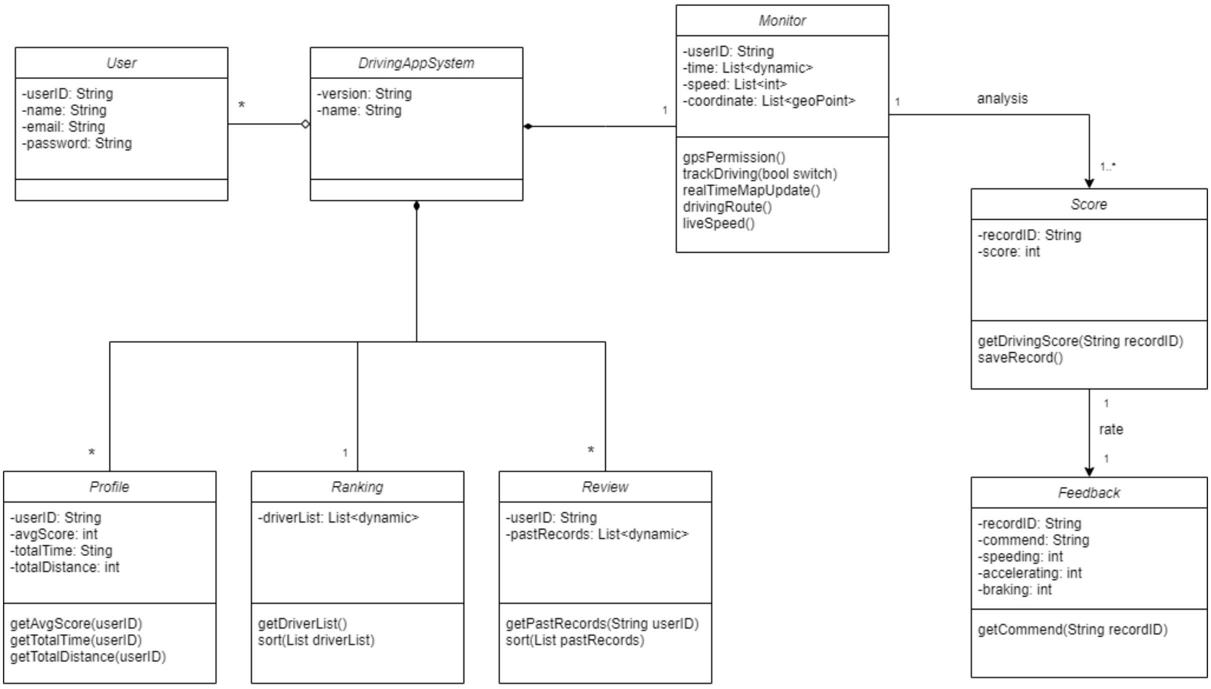


Fig 9 - Class Diagram

6. Sequence Diagram

This section is about modeling the interaction between the user and the object. They explain how the different parts of the system interact and perform functions.

6.1 Monitor Scenario

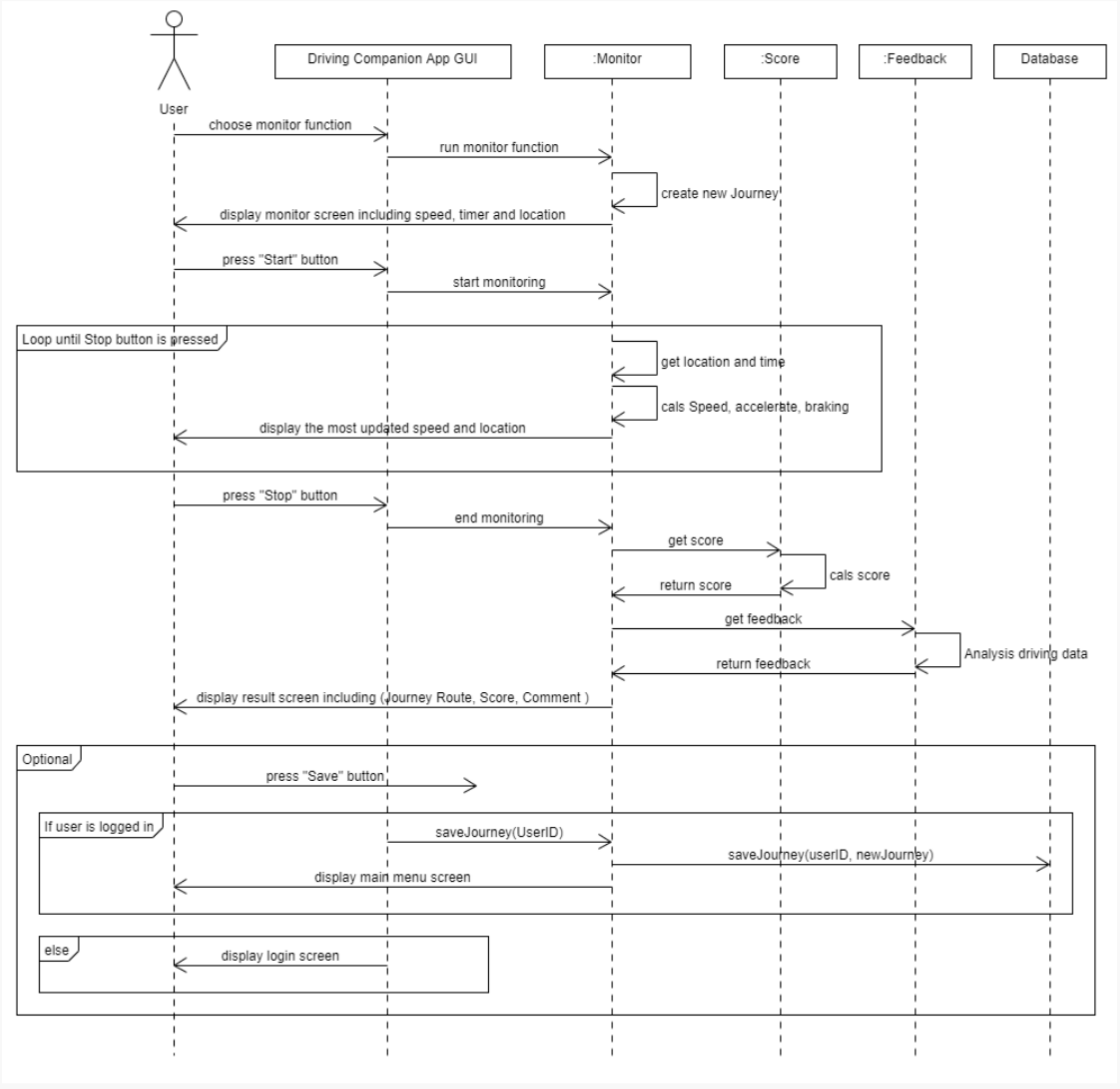


Fig 10 - Monitor sequence diagram

6.3 Score Scenario

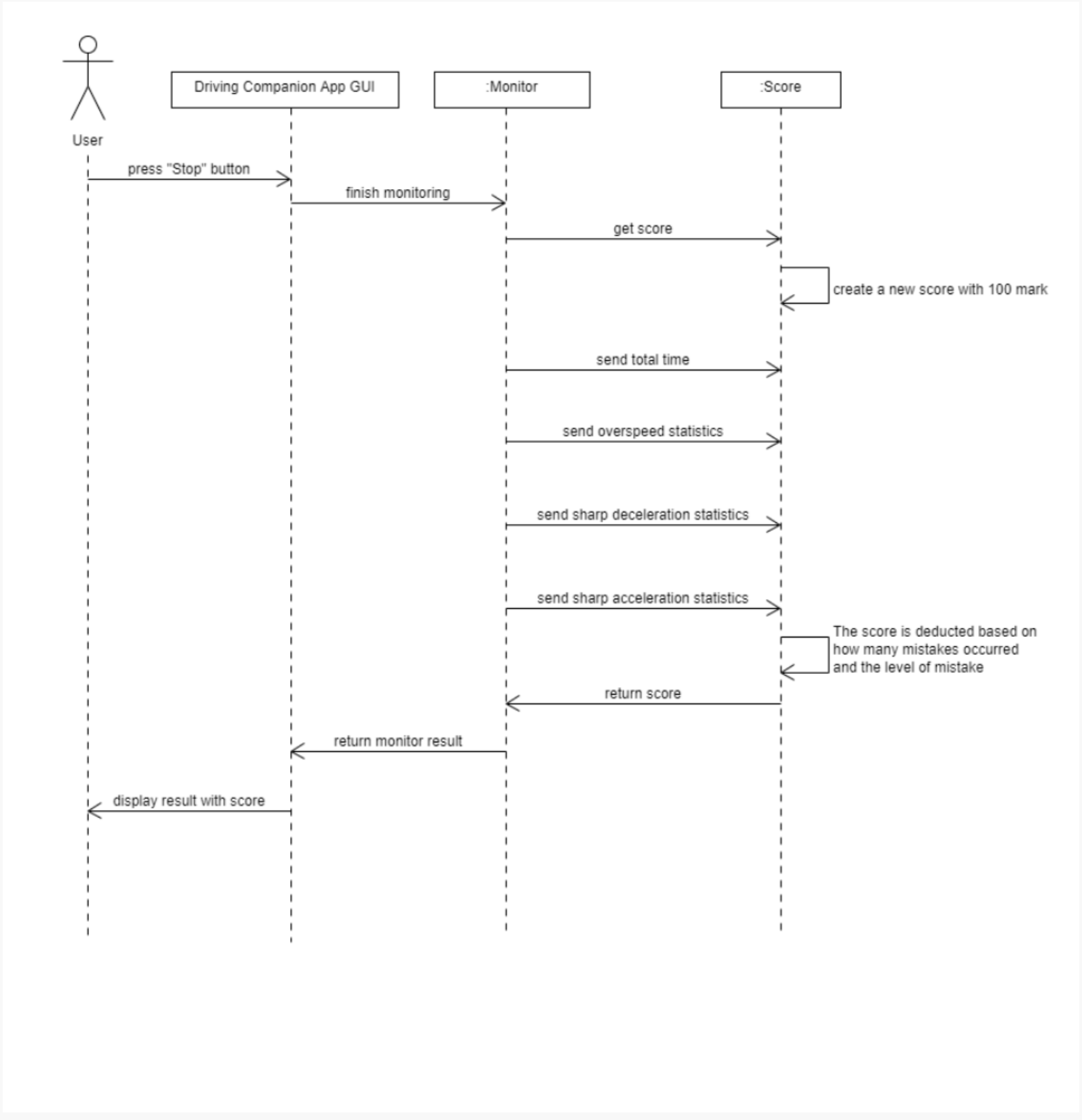

User
Driving Companion App GUI
:Monitor
:Score
press "Stop" button
finish monitoring
get score
create a new score with 100 mark
send total time
send overspeed statistics
send sharp deceleration statistics
send sharp acceleration statistics
The score is deducted based on how many mistakes occurred and the level of mistake
return score
return monitor result
display result with score

Fig 11 - Score sequence diagram

6.3 Login Scenario

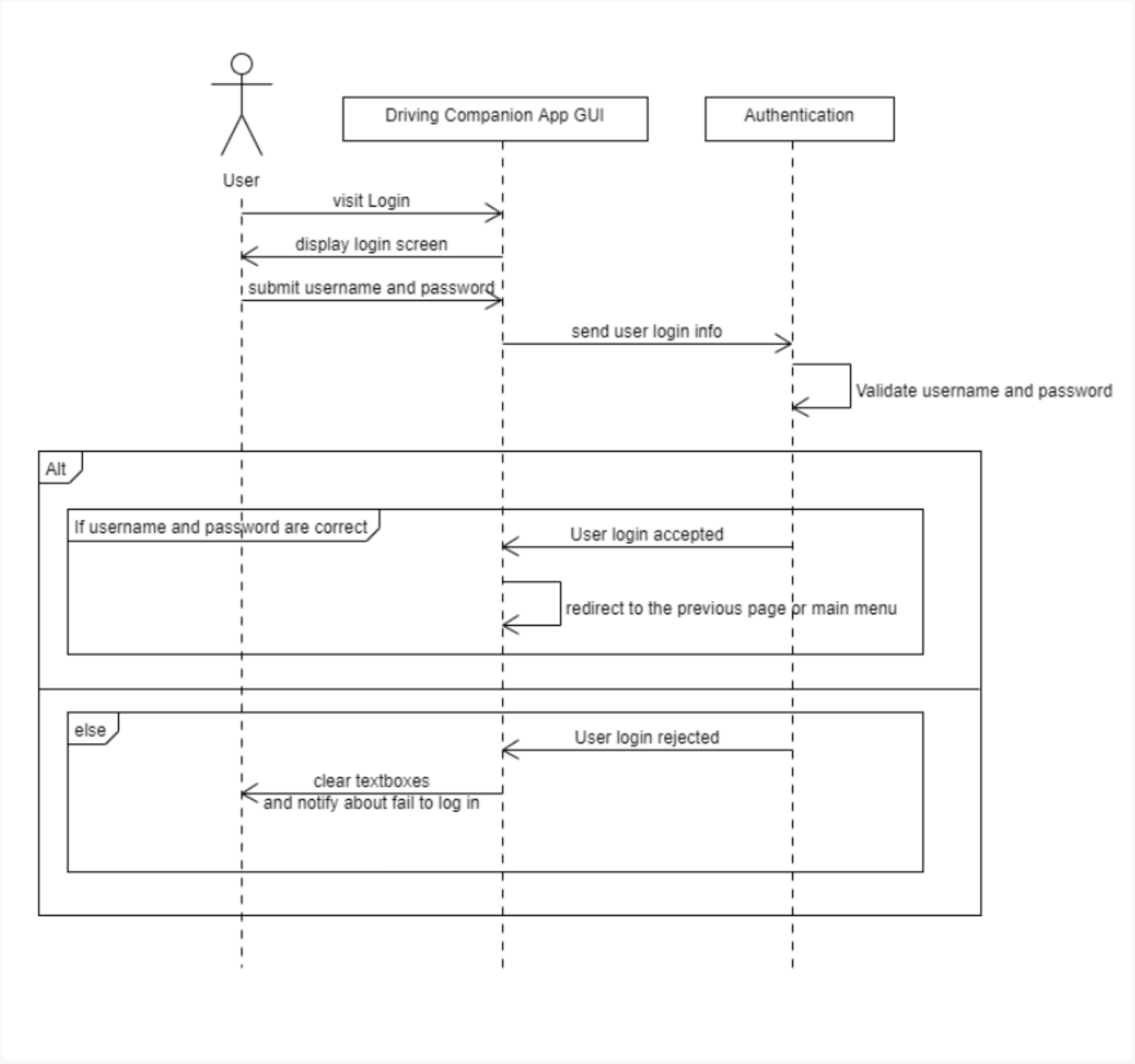

User
Driving Companion App GUI
Authentication
visit Login
display login screen
submit username and password
send user login info
Validate username and password
Alt
If username and password are correct
User login accepted
redirect to the previous page or main menu
else
User login rejected
clear textboxes
and notify about fail to log in

Fig 12 - Login sequence diagram

7. Database Schema

The data is in JSON format based on the need to be compatible with Firebase.

7.1 Journey data example:

```
{
        “journeys” : {
                “journey1”: {
                        “.driver” : “peterpan123@gmail.com”,
                        “.linearChart” :  { {0,0},{10,56},{20,118},{30,86},……{100,0} },
                        “.score” : 78,
                        “.comment” : 3,
                        “.time” : 2282,
                        “.distance” : 57.821,
                        “.maxSpeed” : 118,
                        “.avgSpeed” : 52,
                        “.sharpAcceleration”: 6,
                        “.sharpDeceleration” : 3,
                        “.deleteflag” : 0
                },
                “journey2”: {
                        ...
                }
        }
}
```

7.2 Driver data example:

```
{
        “drivers” : {
                “driver1”: {
                        “.email” : “peterpan123@gmail.com”,
                        “.password” : “DCAfcea920f7412b5a7be0cf42b8c93759”,
                        “.firstname” : “peter”,
                        “.lastname” : “pan”,
                        “.gender” : 1,
                        “.dob” : “12-01-1987”,
                        “.active” : true,
                        “.deleteflag” : 0
                },
                “driver2”: {
                        ...
                },
                “driver3”: {
                        ...
                }
        }
}
```

8. Bibliography

Elitech 2021, Flutter App advantages and disadvantages, available on
https://www.elitechsystems.com/start-with-flutter-basics-the-good-and-the-bad/

Dart 2021, Dart overview, available on
https://dart.dev/tutorials

Back4App 2021, Top10 Advantages of Firebase, available on
https://blog.back4app.com/advantages-of-firebase/

Firebase 2021, Configure hosting behavior, available on
https://firebase.google.com/docs/hosting/full-config

Google 2021, Road api overview, available on
https://developers.google.com/maps/documentation/roads/overview

Google 2021, Snapto Roads, available on
https://developers.google.com/maps/documentation/roads/snap

Google 2021, Speed limits, available on
https://developers.google.com/maps/documentation/roads/speed-limits

Google 2021, Write your first flutter app, available on
https://codelabs.developers.google.com/codelabs/first-flutter-app-pt1#0

Souvik Biswas 2019, Implementing Google Sign In, available on
https://medium.com/flutter-community/flutter-implementing-google-sign-in-71888bca24ed

Google 2021, Adding Google Maps to a Flutter app, available on
https://codelabs.developers.google.com/codelabs/google-maps-in-flutter#0

Peter Martinez, How to use flutter UI Design Tool Easily, available on
https://mockitt.wondershare.com/ui-ux-design/flutter-ui-design.html

Flutter 2021, Building layouts, available on
https://docs.flutter.dev/development/ui/layout/tutorial

Mockflow 2021, video tutorials, available on
https://support.mockflow.com/category/92-video-tutorials

rnfirebase 2021, Firebase JSON config, available on
https://rnfirebase.io/app/json-config