

Institiúid Teicneolaíochta Cheatharlach



At the Heart of South Leinster

LifeReceitas

4th Year Project Functional Specification

Student Name: Fillipe Soares

Student ID: C00203202

Supervisor: Joseph Kehoe

Table of Contents

Introduction	2
Target Users	3
System Architecture	4
Use Case Diagrams	5
CRUD Account Use Case (Create)	6
CRUD Account Use Case (Update)	7
CRUD Account Use Case (Delete)	8
Login Use Case	9
Logout Use Case	10
CRUD Cookbook Use Case (Create)	11
CRUD Cookbook Use Case (Read/View)	12
CRUD Cookbook Use Case (Update)	13
CRUD Cookbook Use Case (Delete)	14
Search Recipe Use Case	15
CRUD Recipe Use Case (Create)	16
CRUD Recipe Use Case (Read/View)	17
CRUD Recipe Use Case (Update)	18
CRUD Recipe Use Case (Delete)	19
Rate Recipe Use Case	20
Save Recipe to Cookbook Use Case	21
Private Message Use Case	22
Metrics	23
Functionality	23
Usability	24
Performance	24
Supportability	25
+	25
Project Plan	26
Research	26
Framework	26
Functionality	26
Design	26
Software	26
References	27

Introduction

The goal of this document is to explain the project's functionality and requirements. The aim of this project is to develop a multi-platform social media like the application to allow people to discover food recipes of all types, from different nationalities and cultures. Once in the app, users can search through all the recipes that are already in the app database and the recipes that other users have added. When users create an account they are then given more permissions within the app, such as adding new recipes to the cookbook and sharing with other users, adding other user's recipes to the personal cookbook, rating other user's recipes, and many other small features that are going to be implemented as the app life cycle goes on. The purpose is to allow users from all kinds of cooking skill levels to find healthier alternatives to food recipes they like and love.

The first section of the research, Target Users, will outline the inspiration for this project, how people can discover and learn how to cook new dishes, and how having a mobile close on hand can aid in helping people make better choices when cooking a dish.

The System Architecture section of the research will display a diagram of the initial suggested concept of the architecture.

The Use Case Diagram section will show how the user will interact with the application.

The Metric section of the report will attempt to measure how successful the project development is becoming. This section will also outline FURPS+ as a guide to measuring the success rate. The last section of the research, the Project Plan, will outline the strategy for the project's various stages, which will be divided into three iterations.

Target Users

In today's day and age, anyone can easily order food online with just a few clicks. Due to this facility in ordering any kind of food, a lot of people struggle when it comes to cooking a good healthy substantial meal. Many people end up choosing the easier route, which is eating ready meals or ordering something online. Also, many are afraid of the term "healthy eating" because when hearing this term, people think it is all about eating food that they cannot enjoy but in reality, is all about "food that contains the right amount of nutrients to keep our body fit." (Healthy Food Essay for Students and Children, 2021). With that in mind, healthy eating is about eating a meal that is well balanced.

With the ease of ordering food through a mobile device in mind, an application that helps and teaches a person how to cook a well-balanced meal is a great idea, because many people do want to start eating better, but just don't know where to begin. Of course, anyone can just look up on the web for a dish recipe, but an app on a mobile device makes it much easier and personal to have a cookbook with favourite dishes saved on hand.

A user can access the app anywhere even without the need for an internet connection. With that in mind developing a Progressive Web Application is the best option. "A progressive web application takes advantage of the latest technologies to combine the best of web and mobile apps. Think of it as a website built using web technologies but that acts and feels like an app." (A Beginner's Guide To Progressive Web Apps — Smashing Magazine, 2021). Other advantages of designing a Progressive Web Application include the ability to install it, get push notifications, and use it offline. By caching information until an internet connection is re-established, service workers enable offline use.

The target audience for this project is any person that is looking to start cooking better, healthier substantial meals. The aim is to develop a Progressive Web Application that can help people make healthier meal choices, where a user can search through a variety of recipes and create a personal cookbook with favourite meals that can be brought anywhere.

The inspiration for this project came when searching for calories tracker applications that had the option for users to search for food recipes. After brief research was conducted, the decision was made to drop the calorie tracker function and focus the research on food recipe applications, so that the proposed application is well developed.

System Architecture

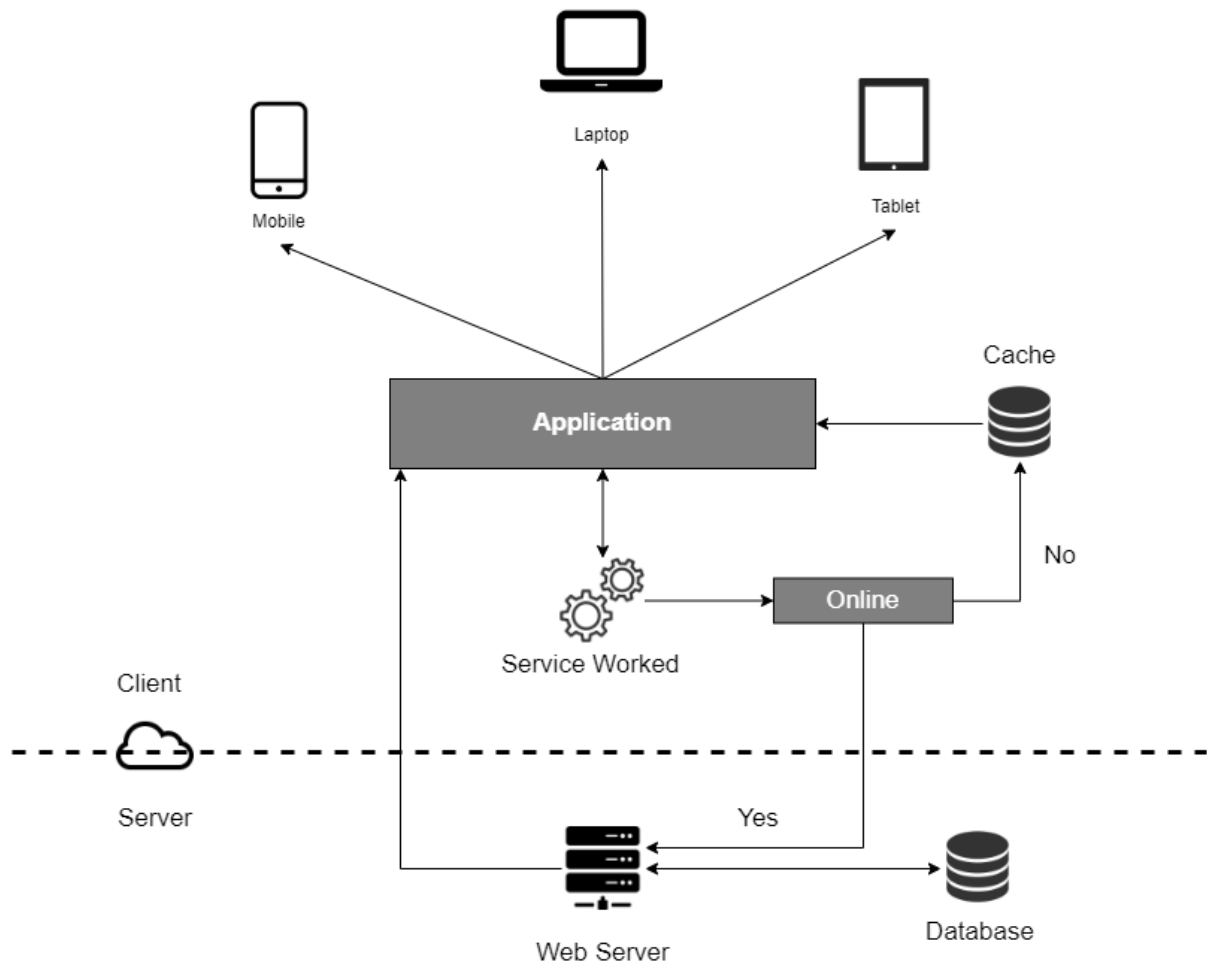
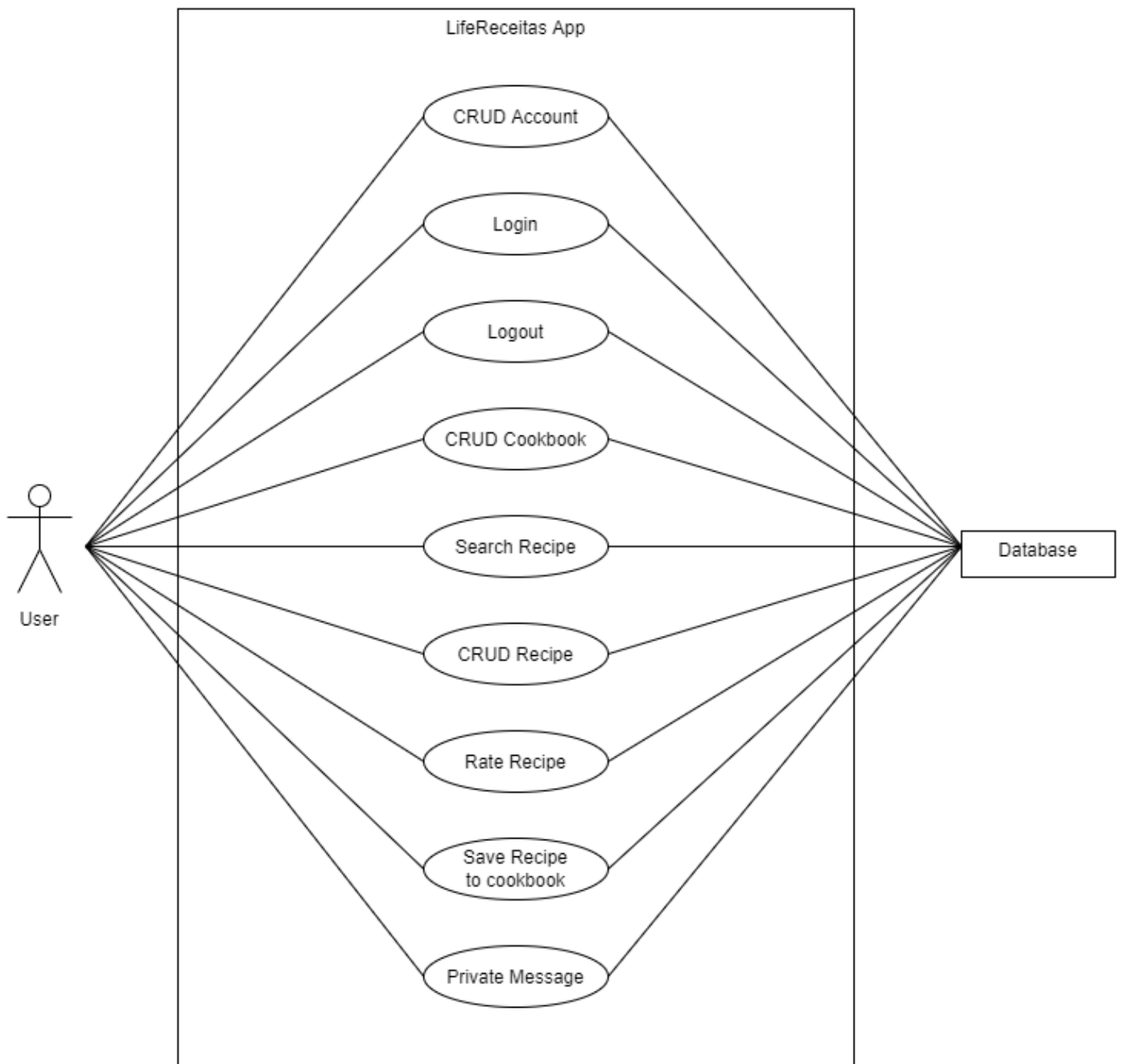


Fig. 1: Initial Architecture Design

Fig. 1 above shows the relationship between the application and all the devices. It also shows how service workers interact with the application to know if the mobile device has an internet connection or not. When the application is offline service workers will use the data saved on the cache until the application has an internet connection. Once an internet connection is established service workers will sync with the server and database.

Use Case Diagrams



CRUD Account Use Case (Create)

Name	CRUD Account
Actors	User, Database
Description	This use case describes the process of a user creating an account.
Main Scenario	<ol style="list-style-type: none">1. User chooses to create an account.2. User presses "Signup" on the main screen.3. User fills out the form with personal details such as name, email and password.4. User presses the "Signup" button at the end of the form.5. The information is sent to database6. The database stores the information.7. A message appears on the screen to the user saying "Signed up successfully".8. User is redirected to the login page.
Alternatives	<p>3a. User entered an email that has been taken.</p> <ol style="list-style-type: none">1. The app prompts the user that the email already exists in the system and goes back to step 3. <p>6a. The Database does not store the user information</p> <ol style="list-style-type: none">1. The app prompts a user that it cannot store the given information in the database.2. The user is prompted to try again later.3. User goes back to step 3.

CRUD Account Use Case (Update)

Name	CRUD Account
Actors	User, Database
Description	This use case describes the process of a user updating account information.
Activity	<ol style="list-style-type: none">1. User presses on “My Account” in the app menu.2. User presses “Edit Account Information”.3. User is then presented with all account personal information.4. User chooses the information that he/she would like to update.5. User presses the update button.6. The information is sent to database7. The database updates and stores the new information.8. A message appears on the screen to the user saying “Updated Successfully”.9. User is redirected to the “My Account” page.
Alternatives	<p>3a. User entered an email that has been taken.</p> <ol style="list-style-type: none">1. The app prompts the user that the email already exists in the system and goes back to step 3. <p>6a. The Database does not store the user information</p> <ol style="list-style-type: none">1. The app prompts a user that it cannot store the given information in the database.2. The user is prompted to try again later.3. User goes back to step 3.

CRUD Account Use Case (Delete)

Name	CRUD Account
Actors	User, Database
Description	This use case describes the process of a user deleting his/hers account.
Activity	<ol style="list-style-type: none">1. User presses on "My Account" in the app menu.2. User presses "Delete Account".3. User is asked to enter the account password.4. User is prompted with a message saying "Are you sure you want to permanently delete your account?"5. User presses the confirm button.6. The database marks the account as deleted.7. User is redirected to the application main page.
Alternatives	<p>4a. User does not wish to delete the account.</p> <ol style="list-style-type: none">1. User presses the cancel button and is redirected to step 1.

Login Use Case

Name	Login
Actors	User, Database
Description	This use case describes the process of a user login into the app.
Activity	<ol style="list-style-type: none">1. User presses "Login" on the app main page.2. User enters email and password and presses the login button.3. The application verifies user credentials in the database.4. User is redirected to the home page.
Alternatives	<p>2a. User enters the wrong email and/or password</p> <ol style="list-style-type: none">1. The user is prompted with a message saying "Wrong email and/or password".2. User is redirected to step 2.

Logout Use Case

Name	Logout
Actors	User
Description	This use case describes the process of a user logout of the app.
Activity	<ol style="list-style-type: none">1. User presses the menu button icon on the top of the page.2. User presses logout.3. User is prompted with a message saying "Are you sure you want to log out".4. User presses the confirm button.5. User is redirected to the application main page.
Alternatives	<p>3a. User does not wish to log out.</p> <ol style="list-style-type: none">1. User presses the cancel button and is redirected to step 1.

CRUD Cookbook Use Case (Create)

Name	CRUD Cookbook
Actors	User, Database
Description	This use case describes the process of a user creating a cookbook.
Main Scenario	<ol style="list-style-type: none">1. User presses the menu button icon.2. User presses on "My Cookbooks".3. User presses the "+" button to create a new cookbook and is prompted with a form.4. User fills out the form with the cookbook name and description.5. User presses the "Save" button at the end of the form.6. The information is sent to database7. The database stores the information.8. A message appears on the screen to the user saying "Cookbook created successfully".9. User is redirected to "My Cookbooks".
Alternatives	<p>5a. User entered a cookbook name that already exists.</p> <ol style="list-style-type: none">1. The app prompts the user that the cookbook name already exists in the system and goes back to step 3. <p>7a. The Database does not store the user information</p> <ol style="list-style-type: none">1. The app prompts a user that it cannot store the given information in the database.2. The user is prompted to try again later.3. User goes back to step 4.

CRUD Cookbook Use Case (Read/View)

Name	CRUD Cookbook
Actors	User, Database
Description	This use case describes the process of when a user views a cookbook.
Main Scenario	<ol style="list-style-type: none">1. User presses the menu button icon.2. User presses "My Cookbooks".3. User is presented with all the cookbooks.4. User chooses one of the cookbooks.5. All saved recipes are displayed to the user.
Alternatives	<p>3a. Cookbooks could not be retrieved from the database.</p> <ol style="list-style-type: none">1. The app prompts the user that cookbooks could not be retrieved and goes back to step 2. <p>5a. recipes could not be retrieved from the database.</p> <ol style="list-style-type: none">2. The app prompts the user that recipes could not be retrieved and goes back to step 3.

CRUD Cookbook Use Case (Update)

Name	CRUD Cookbook
Actors	User, Database
Description	This use case describes the process of a user updating cookbook information.
Activity	<ol style="list-style-type: none"> 1. User presses the menu button icon. 2. User presses “My Cookbooks” in the app menu. 3. User is presented with all saved cookbooks. 4. User presses the setting icon beside the chosen cookbook. 5. User presses “Edit” and is presented with a form to edit the cookbook name, description and/or image. 6. User changes cookbook information and presses the update button. 7. The information is sent to database 8. The database updates and stores the new information. 9. A message appears on the screen to the user saying “Cookbook Updated Successfully”. 10. User is redirected to the “My Cookbooks” page.
Alternatives	<ol style="list-style-type: none"> 6a. User entered a name that already exists. <ol style="list-style-type: none"> 2. The app prompts the user that the name already exists in the system and goes back to step 5. 8a. The Database does not store the user information <ol style="list-style-type: none"> 4. The app prompts a user that it cannot store the given information in the database. 5. The user is prompted to try again later. 6. User goes back to step 5.

CRUD Cookbook Use Case (Delete)

Name	CRUD Cookbook
Actors	User, Database
Description	This use case describes the process of a user deleting a cookbook.
Activity	<ol style="list-style-type: none">1. User presses the menu button icon.2. User presses "My Cookbooks" in the app menu.3. User is presented with all saved cookbooks.4. User presses the setting icon beside the chosen cookbook.5. User presses "Delete".6. User is prompted with a message asking "Are you sure you want to permanently delete this?"7. User presses the confirm button.8. The database marks the cookbook as deleted.9. User is redirected to the "My Cookbooks" page.
Alternatives	<p>5a. User does not wish to delete the cookbook.</p> <ol style="list-style-type: none">1. User presses the cancel button and is redirected to step 3.

Search Recipe Use Case

Name	Search Recipe
Actors	User, Database
Description	This use case describes the process of a user searching for a recipe.
Activity	<ol style="list-style-type: none">1. User presses the search icon to search for recipes in the "Recipes" tab.2. User types a dish name or an ingredient into a search box.3. User is presented with related recipes.
Alternatives	<p>2a. Recipes not found and/or do not exist in the database.</p> <ol style="list-style-type: none">1. App prompts the user by saying that the searched recipes could not be found and/or do not exist.2. User is redirected to the "Recipe" tab.

CRUD Recipe Use Case (Create)

Name	CRUD Recipe
Actors	User, Database
Description	This use case describes the process of a user creating a new recipe.
Main Scenario	<ol style="list-style-type: none">1. User presses the menu button icon.2. User presses on "My Recipes".3. User presses the "+" button to create a new recipe and is prompted with a form.4. User is fills out the form with recipe details.5. User chooses if the recipe is going to be public or private.6. User presses the "Save" button at the end of the form.7. The information is sent to database8. The database stores the information.9. A message appears on the screen to the user saying "Recipe created successfully".10. User is redirected to "My Recipes".
Alternatives	<p>6a. Recipes could not be published.</p> <ol style="list-style-type: none">1. The app informs the user that recipes could not be published and to try again later.2. User is redirected back to step 4.

CRUD Recipe Use Case (Read/View)

Name	CRUD Recipe
Actors	User, Database
Description	This use case describes the process of a user viewing a personal recipe.
Main Scenario	<ol style="list-style-type: none">1. User presses the menu button icon.2. User presses on "My Recipes".3. User is presented with all saved private recipes.4. User chooses one of the recipes.5. Recipe is displayed to the user.
Alternatives	<p>3a. Recipes could not be retrieved from the database.</p> <ol style="list-style-type: none">1. The app prompts the user that recipes could not be retrieved and goes back to step 2.

CRUD Recipe Use Case (Update)

Name	CRUD Recipe
Actors	User, Database
Description	This use case describes the process of a user updating personal recipe information.
Activity	<ol style="list-style-type: none">1. User presses the menu button icon.2. User presses on "My Recipes" in the app menu.3. User is presented with all saved private recipes.4. User presses the setting icon beside the chosen recipe .5. User presses "Edit" and is presented with a form to edit recipe details.6. User changes recipe details and presses the update button.7. The information is sent to database8. The database updates and stores the new information.9. A message appears on the screen to the user saying "Recipe Updated Successfully".10. User is redirected to the "My Recipes" page.
Alternatives	<p>8a. The Database does not store the user information</p> <ol style="list-style-type: none">1. The app prompts a user that it cannot store the given information in the database.2. The user is prompted to try again later.3. User goes back to step 6.

CRUD Recipe Use Case (Delete)

Name	CRUD Cookbook
Actors	User, Database
Description	This use case describes the process of a user deleting a personal recipe.
Activity	<ol style="list-style-type: none">1. User presses the menu button icon.2. User presses on "My Recipes" in the app menu.3. User is presented with all saved personal recipes.4. User presses the setting icon beside the chosen recipe.5. User presses "Delete".6. User is prompted with a message asking "Are you sure you want to permanently delete this?"7. User presses the confirm button.8. The database marks the recipe as deleted.9. User is redirected to "My Recipes" page.
Alternatives	<ol style="list-style-type: none">5a. User does not wish to delete the recipe.2. User presses the cancel button and is redirected to step 3.

Rate Recipe Use Case

Name	Rate Recipe
Actors	User, Database
Description	This use case describes the process of a user rating and reviewing a public recipe.
Activity	<ol style="list-style-type: none">1. User presses the menu button icon.2. User presses "Recipes" in the app menu.3. User chooses the desired recipe.4. User presses "Rate and Review".5. User rates, types review and presses submit.6. App informs the user that the rating and review were submitted successfully and stored in the database.7. User is redirected to "Recipes".
Alternatives	<p>5a. Rating and review could be submitted.</p> <ol style="list-style-type: none">1. App informs users that ratings and reviews could not be sent and to try again later.2. User is redirected to step 4.

Save Recipe to Cookbook Use Case

Name	Save Recipe to Cookbook
Actors	User
Description	This use case describes the process of a user saving public recipes to private cookbooks.
Activity	<ol style="list-style-type: none">1. User presses the menu button icon.2. User presses "Recipes" in the app menu.3. User searches for a recipe and presses on the chosen recipe to open.4. User presses the "Save" icon and has the option to save to a specific cookbook.5. User saves the recipe and the app displays "Recipe Saved Successfully".6. User is redirected back to the chosen recipe.
Alternatives	<p>5a. Recipe could not be saved.</p> <ol style="list-style-type: none">1. The app informs the user that the recipe could not be saved and to try again later.2. User is redirected to step 4.

Private Message Use Case

Name	Start Private Message
Actors	User, Database
Description	This use case describes the process of a user sending private messages to other users.
Activity	<ol style="list-style-type: none"> 1. User presses the menu button icon. 2. User presses on "Recipes" in the app menu. 3. User chooses a recipe. 4. User presses the "Message" icon beside the recipe author name to start a private chat. 5. User sends a message and the message is securely delivered. 6. Messages are sent and stored in the database encrypted.
Alternatives	<p>5a. Messages could not be sent.</p> <ol style="list-style-type: none"> 1. The app informs the user that message could not be sent and to try again later. 2. User is redirected to step 4.

Name	Open Private Message
Actors	User, Database
Description	This use case describes the process of a user opening private messages.
Activity	<ol style="list-style-type: none"> 1. User presses the menu button icon. 2. User presses on "Messages" in the app menu. 3. User presses on the chosen message notification and the chatbox opens. 4. User replies to the message and closes the chat by pressing the "X" icon. 5. User is redirected to the "Messages" tab
Alternatives	<p>4a. Messages could not be sent.</p> <ol style="list-style-type: none"> 3. The app informs the user that message could not be sent and to try again later. 4. User is redirected to step 3.

Metrics

To establish whether or not this project is a success, it will be analysed against a set of goals or objectives. There will be some minimum standards that must be met, as well as extra goals that will be considered more ambitious. At the very least, this project must deliver a Progressive Web Application that can be utilised by many users.

For the application to be successful users should be to do the following:

- CRUD Account
- CRUD Recipe
- CRUD Private Cookbook
- Share Recipes to Other Users
- Search For Recipes
- Filter Search
- Rate and Save Recipes
- Send Private messages

An ambitious goal for this project is to allow users to add recipes straight from the web by copying the link from a website and pasting into the app. Another goal is to allow users to send private messages to each other, this will allow people to interact with each other and ask questions about recipes they may have.

To further establish whether or not this project is a success, the FURPS+ model will be used as a guideline. "FURPS is a technique to validate the prioritised requirements after an understanding of the client's needs and necessities. The acronym FURPS is Functionality, Usability, Reliability, Performance, and Supportability, the "+" of the FURPS+ acronym allows us to specify constraints, including design, implementation, interface, and physical constraints." (What is FURPS+?, 2021)

Functionality

"What the customer wants! Note that this includes security-related needs."(Ottinger and Langr, 2021).

To establish the success of this application the following functionalities will be required:

- The user must be able to register/log in.
- When there is no internet connection, the system will cache the user's session.
- To access the database, the app must be connected to the internet at all times.

- Internet connection is required to send private messages.
- The system will allow CRUD operations throughout the system.

Usability

“How effective is the product from the standpoint of the person who must use it? Is it aesthetically acceptable? Is the documentation accurate and complete?”(Ottinger and Langr, 2021).

To establish the success of this application the following usabilityes will be required:

- The app should be cross-platform and accessible on multiple mobile devices and browsers.
- The application should be aesthetically pleasing.
- Users should be able to easily log in after filling out credentials.
- All forms should be clearly labelled and users should be able to fill them with ease.
- Error messages should be clearly displayed.
- Users should be able to easily filter and search for recipes.
- Users should be able to quickly navigate through pages, with a navigation average of 10 seconds.
- Users should be able to successfully send and receive private messages.
- The application should be fully mobile responsive.

Reliability

“What is the maximum acceptable system downtime? Are failures predictable? Can we demonstrate the accuracy of results? How is the system recovered?”(Ottinger and Langr, 2021).

To establish the success of this application the following reliability will be required:

- As it is a progressive web application, the system must cache the user's session when there is no internet connection.
- The application should have a 99% uptime.
- The application must be bug-free and handle any errors in a way that it does not crash.
- All personal cookbooks should be accessible without an internet connection.

Performance

“How fast must it be? What's the maximum response time? What's the throughput? What's the memory consumption?”(Ottinger and Langr, 2021).

To establish the success of this application the following performance will be required:

- At any given moment, the system should be able to support a high volume of active users.
- Users should not have to wait more than 5 seconds for the application to load.
- Adding or editing recipes to the database should take no more than 2 seconds.

Supportability

“Is it testable, extensible, serviceable, installable, and configurable? Can it be monitored?”(Ottinger and Langr, 2021).

To establish the success of this application the following supportability will be required:

- With the responsive design, the application should work on practically every platform.
- The application should easily allow for upgrades and the addition of new features in the future.

+

“the “+” of the FURPS+ acronym allows us to specify constraints, including design, implementation, interface, and physical constraints.” (What is FURPS+?, 2021)

To establish the success of this application the plus section will cover the security side aspect.

- The application should be delivered via HTTPS
- After three unsuccessful attempts, an unauthenticated user should be locked out of the app for 30 minutes.
- Passwords should be at least 8 characters long, with one lowercase and uppercase letter, one number, and one special character included.
- To secure users' data, the system must use encryption.

Project Plan

This portion of the research will explain the strategy for the project's various stages. Each stage will provide tasks to be done in order to complete the application.

Research

The research part began by searching for existing applications to see what kind of features each application has. Further research was carried out on the features of each application to figure out what ways they could be improved. During the research stage, it was decided that building a Progressive Web Application was the best option as it suits various mobile devices.

Framework

After deciding that progressive was the way to go, the next step was to choose a framework. The first aim is to build the Progressive Web Application on top of HTML and CSS using React and Bootstrap. After thorough research, it was determined that the back end should be built using NoSQL rather than a SQL database. It was also determined that Visual Studio Code is the code editor that will be used to develop the application.

Functionality

The functionality of the Progressive Web Application had to be defined when the research phase of the project was completed, so the project's primary functionalities were decided at this time when writing the functional specification.

Design

The project must be designed after the functionality has been determined, so screens and general navigation must first be planned before any coding can begin. While no user interfaces are established at this time, it is critical that all functionalities have been chosen and that the whole application navigation is defined. This is also when the database tables will be created.

Software

Coding will begin at this point in the project with the goal of delivering an initial version of the application. The IDEs and frameworks must be set up in order to start studying the chosen languages and rapidly determine what will and will not work. This phase of the project will be when many technologies will be tested in order to determine which one is the best to develop a Progressive Web Application.

PWA Implementation

At this stage of the project It is expected that the Progressive Web Application will be developed and fully functional. It is hard to predict the number of features that are going to be completed at this point, but the application should at least be progressive and installable and users should be able to search through recipes. The signup and login system is also expected to be fully working at this point.

References

Business Analyst Training in Hyderabad - COEPD. 2021. *What is FURPS+?*. [online] Available at: <<https://businessanalysttraininghyderabad.wordpress.com/2014/08/05/what-is-furps/>> [Accessed 27 November 2021].

Choudhary, T., 2021. *Analysis of Software Quality Models for Organizations*. [online] D1wqtxts1xzle7.cloudfront.net. Available at: <https://d1wqtxts1xzle7.cloudfront.net/35742156/04-vol1issue2-with-cover-page-v2.pdf?Expires=1638027682&Signature=KV5GzVGnhfCu4jVJSFdI5Rfb0pEhjiop~iq0bSDtTEzPI0gHPsRYehsW-7K5Ihz7NUUPuUBeg2SnrfaJB5cffRUIPajFV6X9LTOFR3EU83fvKOTBkY2zzIEBY5gPzj2zeJK5w7OeUNuTHzJvJz82jFNUeYYY4brMRNF0Gqm6pJlegq1Xf1-w6Txdrr3qIMYRs7dNWiy3p1~vXOOZL4p2I~bRFt~3o-LfmJoiour9JyeHZ~IBdwceTIVRhtZ15r-JvlZtB8KyvXorJvTTjsAL62DLgv6IwuGDs7echDBuA5tJOdC9CXPIbkM~K6pVkxGIGKibRPbQKZD9v65dberhzQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA> [Accessed 27 November 2021].

Smashing Magazine. 2021. *A Beginner's Guide To Progressive Web Apps — Smashing Magazine*. [online] Available at: <<https://www.smashingmagazine.com/2016/08/a-beginners-guide-to-progressive-web-apps/>> [Accessed 25 November 2021].

Toppr. 2021. *Healthy Food Essay for Students and Children*. [online] Available at: <<https://www.toppr.com/guides/essays/healthy-food-essay/>> [Accessed 25 November 2021].