An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning.

By

VICKERN TAN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)

COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning

Academic Session: Y3S3

I VICKERN TAN

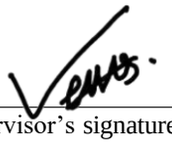
declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)



(Supervisor's signature)

Address:

No.9, Jalan Desa Botani
10, Taman Desa Botani,
83000, Batu Pahat, Johor

Ms. Tan Lyk Yin

Supervisor's name

Date: 12/9/2024

Date: 12/9/2024

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 12/9/2024

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that Vickern Tan (ID No: 21ACB05679) has completed this final year project entitled " An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning " under the supervision of Ms. Tan Lyk Yin (Supervisor).

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

DECLARATION OF ORIGINALITY

I declare that this report entitled “An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : Vickern Tan

Date : 12/9/2024

ACKNOWLEDGEMENTS

I would like to express my deep heartfelt thanks and appreciation to my supervisor, Ms. Tan Lyk Yin, who has given me this unique opportunity to engage in a flutter-based mobile food-recipe application system. The suggestions, guidance, and comments from you are precious to me. In addition, I would like to thanks my parents and family for their love, unconditional continuous support, and encouragement throughout the course. Greatly appreciates to you all.

ABSTRACT

This project develops a Flutter-based mobile application addressing challenges in recipe discovery and grocery management. Motivated by the need to reduce food wastage and improve culinary experiences, the application offers advanced recipe search features and a personalized grocery list function. Users can search recipes by name, ingredients, or video content, and seamlessly add ingredients to their grocery lists. The application aims to tackle disorganized shopping practices prevalent in households, which often lead to excessive food wastage and financial losses. By providing a comprehensive solution that combines recipe exploration with efficient grocery management, the project seeks to foster a more sustainable and health-conscious approach to meal preparation. To demonstrate potential integration with online grocery platforms, the project implements a simulated product storage system using a realtime database. This showcases how the application could streamline the shopping process in real-world scenarios, bridging the gap between meal planning and grocery acquisition. Developed using Visual Studio Code for coding, Android Studio for testing and deployment, and Firebase for backend services and database management, the application ensures robust functionality and user-friendly interface. It targets university students and busy individuals seeking to optimize their cooking experiences, providing wider accessibility to the user. The project's significance lies in its potential to leverage technology as a catalyst for positive change in the culinary domain. By streamlining the process from recipe selection to ingredient listing, it aims to reduce food wastage, promote culinary diversity, and improve overall efficiency in meal preparation. This approach not only enhances individual user experiences but also contributes to broader environmental sustainability efforts.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: PROJECT BACKGROUND

In this chapter, the background and motivation for the research are discussed, alongside the contributions made to the field. Additionally, an outline of the report is provided.

1.1 Introduction

The ubiquitous impact of technology in the contemporary world has transformed many elements of daily life, surpassing old boundaries and redefining conventional practices. The incorporation of a mobile application into this project, which sits at the nexus of this technology advancement and the culinary world, promises to redefine personalized grocery management and recipe research [1]. The Flutter-based food-recipe mobile application embodies this transformation, offering a comprehensive solution for recipe exploration and grocery management, with seamless integration to online shopping platforms.

The digital revolution that characterizes this time period has made mobile phones and applications crucial for coordinating routines, optimizing work, and accommodating personal preferences. Culinary arts continue the project's inspiration came from seeing the potential of this digital revolution to rethink culinary experiences, where the science of technology and the art of cooking come together [2].

At its core, the application offers advanced recipe exploration features, allowing users to search for recipes by name, available ingredients, or even video content. This functionality addresses the need for culinary diversity and inspiration, catering to various skill levels and dietary preferences. The integrated grocery list feature enables users to effortlessly add ingredients from recipes to their shopping list, with the flexibility to modify quantities and details as needed.

Furthermore, the application bridges the gap between recipe selection and grocery shopping by linking directly with external online grocery platforms. This innovative feature allows users to seamlessly add ingredients to their cart and make purchases without leaving the application, streamlining the process from recipe selection to ingredient acquisition.

This project aims to create an integrated mobile application that goes beyond customary culinary techniques and to enable the community to enter the field of mobile application development with an educational experience. The commitment to

developing a solution using the Flutter framework ensures a robust, user-friendly experience across Android devices. Beyond the application's functional utility, the project aspires to impart strategic methodologies, design principles, and technical intricacies pivotal to crafting such solutions in the mobile application development landscape.

1.2 Problem Statement and Motivation

In the contemporary fast-paced era, college students face significant meal preparation challenges. These include disorganized shopping practices that lead to wasted ingredients [3], and a limited culinary exploration due to reliance on inadequate recipe applications [4]. The Flutter-based food-recipe application directly addresses these issues by providing a comprehensive platform for recipe discovery and ingredient management.

The challenges are exacerbated by prolonged grocery shopping trips caused by the time spent selecting ingredients and the lack of precise measurements for required quantities, which not only reduces efficiency but also contributes to food wastage [3]. The application's grocery list feature allows users to add ingredients directly from recipes, ensuring accurate quantities and reducing over-purchasing. This addresses the issue highlighted in the study "Food Waste Behavior and Awareness of Malaysians" from PubMed Central, which found that 45.8% of Malaysians waste food due to overestimating their household skills, particularly in estimating the quantity of food needed [5].

Moreover, the audience's inclination towards a limited selection of recipes, driven by a lack of confidence in cooking, time constraints, and the financial barriers posed by subscription-based food-recipe applications, leads to culinary stagnation. The application combats this by offering a diverse range of recipes through its advanced search features, allowing users to discover new dishes based on ingredients they already have or by exploring video recipes. This functionality directly addresses the absence of a dedicated application for recipe exploration and sharing, enabling individuals to discover new and diverse dishes that cater to their dietary preferences and nutritional needs [6].

By providing a unified platform for recipe discovery, ingredient management, and grocery shopping, the Flutter-based food-recipe application aims to alleviate the stress related to meal preparation and promote more efficient, sustainable cooking practices among college students. The application's seamless integration with online grocery platforms further streamlines the process, addressing the inefficiencies in current shopping practices.

This project is driven by the recognition of a significant gap in how meals are prepared today, particularly for university students in a fast-paced environment where time is highly valued. By leveraging mobile technology to address cooking challenges, disorganized shopping practices, and culinary stagnation, the developers aim to reduce food wastage, improve dietary habits, and enhance the overall quality of home-cooked meals. Ultimately, this project demonstrates the transformative potential of mobile applications in the culinary domain, offering a comprehensive solution to the challenges of disorganized shopping and limited recipe exploration.

1.3 Project Objectives

The goal of this project is to conceive, develop, and deliver an integrated mobile application that redefines how individuals approach grocery list management and recipe research. This application is built upon the versatile Flutter framework and is meticulously designed to cater to the needs of university students, addressing the challenges they encounter in their culinary endeavors.

Objective 1: To Develop a Recipe Exploration Platform by Integrating a Culinary API

This project aims to revolutionize the culinary experience by seamlessly integrating grocery management and recipe exploration into a single application. By streamlining the meal preparation process, users can save valuable time and effort while navigating the complexities of cooking. The application enables users to add ingredients directly from their chosen recipes into personalized grocery lists, ensuring that only necessary items are purchased. This feature not only simplifies the shopping process but also significantly reduces food wastage by minimizing excess purchases.

Additionally, the application offers users a concise overview of the required ingredients, enabling them to make informed decisions about what to purchase and what to omit. This feature supports environmental sustainability by encouraging a more thoughtful and efficient approach to grocery shopping and meal preparation. By addressing the widespread issue of food wastage, this project aims to positively impact the user's experience while contributing to broader environmental conservation efforts.

Objective 2: To Implement an Advanced Search Feature for Accessing the Recipe Database by Name or Ingredients.

Inspiring culinary exploration among its users is also a key ambition of the project. With an extensive recipe database and advanced search features, including the ability to search by name, existing ingredients, and recipe videos, the application aims to encourage users to diversify their meals, experiment with new dishes, and embrace a more varied and nutritious diet. By providing access to a diverse range of recipes and culinary inspirations, the project seeks to foster a culture of creativity and experimentation in the kitchen [6].

Objective 3: To Implement a Digital Grocery List Feature that Integrates with Recipes to Streamline the Shopping Process

Effective shopping management is essential for navigating daily responsibilities. By providing a digital grocery list that can be populated directly from recipes and modified as needed, the project aims to assist users in streamlining their shopping process. This feature not only reduces stress associated with grocery shopping but also empowers users to make more efficient and informed purchasing decisions. The integration with external online grocery platforms further enhances this efficiency, allowing users to directly add ingredients to their online shopping carts, thereby saving time and effort in the grocery acquisition process.

1.4 Contributions

This project bestows a multitude of significant contributions that extend both to the immediate users of the application and to the broader community. The primary beneficiary is university students, yet the influence transcends this audience, embracing the domains of technology education and sustainability.

This application augments the user experience for university students through three pivotal features. Firstly, it introduces users to a paradigm of newfound efficiency in meal preparation and grocery shopping. Notably, its personalized grocery management functionality optimizes shopping expeditions, curtailing both the time required and the likelihood of food wastage. Secondly, with access to an extensive repertoire of diverse and inspirational recipes, users are encouraged to embark on journeys of culinary exploration and experimentation. This, in turn, leads to more diversified and nutritionally sound diets. Thirdly, the application serves as an enabler of efficient grocery shopping, affording users the opportunity to economize time and alleviate stress associated with ingredient acquisition.

This application profoundly contributes to the reduction of food wastage and its concomitant environmental ramifications through its precise grocery management feature. By significantly curtailing excessive ingredient purchases, it makes substantial strides toward diminishing food wastage and its associated environmental repercussions. Furthermore, by promoting efficient ingredient utilization and encouraging sustainable food practices, this application synergizes with broader environmental sustainability objectives.

The project's impact extends beyond individual users, as it showcases the potential for seamless integration between online grocery platforms and recipe management applications. By demonstrating the feasibility and benefits of such integration, this project paves the way for future collaborations between the food industry and technology sector. The application serves as a proof-of-concept, highlighting how technology can be leveraged to streamline the cooking experience, promote healthy eating habits, and reduce food waste on a larger scale. As more developers and businesses recognize the value of such integration, it has the potential to revolutionize the way people approach meal planning, grocery shopping, and cooking, ultimately contributing to a more sustainable and health-conscious society.

1.5 Project Scope

This project aims to develop a mobile food-recipe application designed for the Android operating system, focusing on revolutionizing recipe exploration, reducing culinary stress, and reducing food wastage. This application serves as a multifaceted tool that assists users in efficiently managing their grocery shopping and exploring a diverse range of food recipes according to dietary preferences. A key feature of the application is its ability to provide precise measurements for ingredients needed for recipes, allowing users to add specific recipe ingredients into their grocery list, helping the users to avoid the purchase of excess items and thus minimizing food waste. Additionally, the application offers advanced recipe exploration features, including the ability to search by name, existing ingredients, and recipe videos, enhancing the meal preparation experience for users with specific dietary needs or preferences.

The application incorporates a user-friendly database, powered by Google Firebase, for storing personal preferences and information. This enables the application to deliver tailored recommendations and fosters a community through a user rating and recommendation system for recipes, enhancing the reliability of culinary choices. To ensure compliance with online safety and privacy standards, the application adheres to the Children's Online Privacy Protection Act (COPPA) by restricting account registration to users above the age of 13 [8]. This measure safeguards the privacy of young users and aligns with legal requirements for online services [8].

Moreover, this application is designed to be accessible to all users free of charge, ensuring that individuals from diverse backgrounds can benefit from its features without financial constraints. The integration with external online grocery platforms further enhances the application's utility, allowing users to seamlessly transition from recipe selection to ingredient purchase, streamlining the entire cooking process from start to finish.

1.6 Report Organization

This project report is structured into seven chapters, each focusing on different aspects of the food-recipe mobile application. Chapter 1 serves as an introduction, providing background information, outlining the problem statement and motivation, defining the project scope, and detailing the project goals and contributions. Chapter 2 presents a literature review, examining relevant background information and highlighting current practices in existing systems. Chapter 3 delves into the systems approach, explaining the methodology, system requirements, functional requirements, project milestones, and estimated costs. Chapter 4 focuses on system design, utilizing diagrams to illustrate the system's architecture and describing functional modules in terms of input, processing, and output. Chapter 5 details the implementation and setup process of the system. Chapter 6 presents the test cases used to validate the system's functionality. Chapter 7 concludes the project, summarizing key findings and outcomes.

CHAPTER 2: LITERATURE REVIEW

2.1 Review of Operating System

Android and iOS are the dominant operating systems utilized in smartphone and tablet technology. Each system offers distinct features and specifications.

2.1.1 Android

Figure 2.1 – Android Logo

Android, a Linux-based operating system, was initially created by Android Inc. before being acquired by Google in 2005 [9]. While the core of the Android operating system (known as AOSP - Android Open-Source Project) is open source, certain components, such as Google Mobile Services (GMS) and Google Play Store, are closed source and under Google's control [10].

Today, Android stands as the most prevalent smartphone platform globally, finding utilization among numerous mobile manufacturers, including Samsung, Oppo, Vivo, Huawei, Xiaomi and Honor. Initially, Android application development necessitated the use of the Java programming language. However, since 2019, Google has endorsed Kotlin as an additional programming language fully interoperable with Java, gaining preference for Android application development. Additionally, Android now supports multiple programming languages, including Kotlin, Javascript, Dart, and more.

To support this, developers have the flexibility to choose from a range of programming languages and frameworks, such as Kotlin for native Android development, JavaScript for web-based applications using frameworks like React Native, Dart for cross-platform development with Flutter. With around 3.9 billion people worldwide own Android phones [11], Android reigns supreme as the operating system of choice for a vast majority of mobile users. Its open-source nature, coupled with flexible design guidelines and customization options, provides developers with unparallel freedom and adaptability.

Furthermore, publishing apps on the Google Play Store, Android's largest application marketplace, is a streamlined process, facilitating rapid deployment and widespread accessibility.

2.1.2 iOS

Figure 2.2 – iOS Logo

iOS, developed by Apple Inc., stands as a premier operating system in the realm of mobile technology. Renowned for its sleek design, seamless integration across Apple devices, and robust security features, iOS has firmly established itself as the preferred choice among users worldwide.

Notably, iOS operates as a closed system, which contributes to its reputation for smooth performance and enhanced security [12]. Unlike some of its competitors, iOS does not allow users to download and install applications from unknown sources, further bolstering its security measures [12]. However, this closed ecosystem does come with limitations in terms of customization options.

Despite this, iOS continues to captivate users with its user-friendly interface and reliable performance. As of 2024, Apple boasts an impressive 1.46 billion users, underscoring the widespread adoption and enduring popularity of iOS. Nevertheless, publishing an application on iOS App Store is generally more difficult due to high publishing standards for applications.

2.1.3 Operating System Comparison

Operating System	Android	iOS
Supported programming language	Java, Kotlin, Javascript, and more	Swift
Platform	Widely used by different phone manufacturers	Only on Apple devices
Numbers of active devices	3.9 billion	Over 1.4 billion
Customization	Yes	No
Freedom of user interface design	Yes	No
Provides user interface guidelines	Yes	Yes
Criteria for publishing applications	Low	High

Table 2.1 Operating System Comparison

2.2 Review of Framework

Flutter and React Native are two prominent frameworks in the realm of mobile application development. Each framework offers unique features and capabilities, catering to the diverse needs of developers and users alike.

2.2.1 Flutter

Figure 2.3 – Flutter Logo

Flutter, a mobile application development framework created by Google, has revolutionized the process of building high-quality, native-like apps for various platforms, including iOS and Android. Utilizing the Dart programming language, Flutter offers a rich set of pre-built widgets and tools that streamline the application development process [13].

One of the standout features of Flutter is its hot reload capability, enabling developers to instantly see code changes without restarting the application, thus enhancing efficiency and productivity [13]. Furthermore, Flutter adopts a widget-based architecture, simplifying the creation of complex and interactive user interfaces [13]. With a diverse collection of pre-built widgets for common UI elements and customization options, Flutter empowers developers to craft visually appealing and functional apps tailored to their specific requirements [13].

Additionally, Flutter provides seamless access to native device features and APIs through platform-specific plugins, ensuring compatibility and performance across different devices. Powered by Skia, a high-performance 2D rendering engine, Flutter delivers smooth and fast graphics rendering, enhancing the user experience. Moreover, Flutter offers built-in support for internationalization and localization, facilitating the development of apps that cater to diverse language and region preferences [13] . Apart from its technical prowess, Flutter addresses critical industry challenges, such as cost reduction, robust user experience, and increased productivity. By enabling the creation of apps with a single codebase, Flutter significantly reduces development time and costs, making it particularly appealing for startups and businesses with limited resources [14].

Moreover, Flutter's support for material design ensures a consistent and native-looking user interface across different devices, enhancing user satisfaction and

engagement [14]. The framework's productivity-boosting features, including hot reload and multi-faceted widgets, further expedite the application development process, enabling developers to build fluid and intuitive apps efficiently [14]. Flutter's comprehensive features, coupled with its ability to address industry challenges and boost developer productivity, have positioned it as a preferred choice for mobile application development [13] [14]. As businesses strive to deliver high-quality apps that meet user expectations, Flutter continues to play a pivotal role in shaping the future of mobile application development.

2.2.2 React Native

Figure 2.4 – React Native Logo

React Native, introduced by Facebook in 2015, stands out as a JavaScript-based mobile application development framework renowned for its cross-platform capabilities and user-centric approach [15]. Unlike traditional platforms that require knowledge of platform-specific languages like Java or Swift, React Native enables developers to build natively-rendered apps for iOS and Android using JavaScript, streamlining the development process [16]. At the heart of React Native lies its unique rendering mechanism, which eschews WebViews in favor of actual native views and components, resulting in superior performance and user experience [16].

By leveraging JavaScript to produce the application's interface, React Native facilitates seamless access to native views and components, ensuring compatibility with OS-specific features and delivering a near-native experience [15]. One of the hallmark features of React Native is its exceptional code reusability, allowing developers to use a single codebase for multiple platforms, thereby reducing development time, costs, and maintenance efforts [15]. Furthermore, React Native's hot reloading feature enables real-time code changes without the need for manual refresh, enhancing developer productivity and accelerating the development cycle [15]. In terms of performance, React Native's innovative "bridge" concept enables the incorporation of natively-written code, resulting in optimal performance and responsiveness that nearly same as native apps.

Moreover, React Native fosters a vibrant and growing developer community, supported by Facebook's continuous efforts to enhance the framework and address emerging challenges [15]. As React Native continues to evolve and improve, it remains a preferred choice for mobile application development, offering cost efficiency, rapid development cycles, and access to a rich ecosystem of libraries and resources [15]. With

its constant innovation and commitment to excellence, React Native embodies the future of mobile application development, empowering developers to create immersive, high-performance apps that cater to diverse user needs and preferences.

2.2.3 Framework Comparison

Features	Framework	
	Flutter	React Native
Language	Dart	JavaScript
Rendering	Uses Skia, a high-performance 2D rendering engine	Utilizes native views and components
Hot Reload	Supported	Supported
Performance	Near-native performance	Near-native performance
Code Reusability	Single codebase for multiple platforms	Single codebase for multiple platforms
Developer Community	Growing community with active development and support	Established community with extensive resources and libraries
Maintenance	Easier maintenance with streamlined development process	Streamlined development process with access to native APIs
Learning Curve	Moderate learning curve with Dart language	Moderate learning curve with JavaScript

Table 2.2 Framework Comparison

2.3 Traditional methods of Recipe Discovery and Grocery Management

The techniques used in the areas of food recipes and grocery management have evolved significantly over time. While mobile applications and other digital technologies have enabled innovative solutions, it is important to acknowledge the traditional methods as well. These conventional approaches often involve handwritten recipes and grocery lists. This context provides an opportunity to examine how these practices have transformed and how modern technology is reshaping the culinary landscape.

Historically, discovering recipes involved acquiring physical cookbooks through purchase or printing, or transcribing desired recipes by hand onto paper. Although physical copies of cookbooks are now increasingly rare in stores, they can still be purchased online through platforms such as Amazon. With the advent of the internet, accessing food recipes has become more convenient, allowing individuals to download their preferred recipes as e-books or digital files.

Effective grocery management traditionally involved creating a comprehensive shopping list. As experts suggest, "attempting to remember everything off the top of your head can lead to forgetting essential items." Therefore, it is advisable to make a list that corresponds with planned meals for the upcoming week. "Review your planned meals and jot down necessary food items." To avoid purchasing items already in the kitchen, crossing out those that are present is a practical approach. Traditionally, a simple pen and paper sufficed for this task[17].

However, these traditional methods often led to disorganized shopping practices, resulting in excessive food wastage and financial losses. The lack of integration between recipe discovery and grocery list creation made it challenging for individuals to efficiently manage their culinary needs.

The transition from these traditional methods to digital solutions marks a significant shift in how people approach cooking and grocery shopping. Modern applications, like the one developed in this project, aim to bridge the gap between recipe exploration and efficient grocery management, addressing the limitations of traditional methods while preserving their core principles.

2.3.1 Limitation of Traditional Methods

Traditional methods of recipe discovery and grocery management, while time-tested, present several drawbacks and limitations in the modern context.

Firstly, relying on physical cookbooks can be financially challenging, as some editions can be notably more expensive than their digital counterparts. Moreover, these cookbooks often contain a limited selection of recipes, restricting culinary exploration and variety. The physical nature of cookbooks also makes the users less portable compared to mobile devices, limiting their accessibility in various situations.

In contrast, modern individuals can use their smartphones or other digital devices to access an extensive range of food recipes. Popular search engines like Google or Bing and video-sharing platforms like YouTube offer an abundance of recipes and step-by-step tutorials, allowing users to explore a diverse array of dishes without significant cost barriers.

Traditional grocery management, often involving handwritten lists on sticky notes, in notebooks, or on scraps of paper, comes with a significant disadvantage: the higher likelihood of losing these lists. Individuals frequently need to carry these physical lists throughout their day, from classes to working hours, until they finally get the opportunity to shop for groceries. This method is particularly challenging for busy individuals like office workers and students who may find it difficult to keep track of these physical lists in the rush of their daily routines.

Furthermore, the disconnect between recipe selection and grocery list creation in traditional methods can lead to inefficient shopping practices. Users might forget to add all necessary ingredients or struggle to modify quantities based on serving sizes, potentially resulting in overbuying or underbuying ingredients.

In contrast, mobile applications designed for recipe exploration and grocery management offer a more secure, convenient, and integrated solution. These apps allow users to create and maintain digital shopping lists that are directly linked to chosen recipes, are accessible at any time, and eliminate the risk of misplacing or losing the list. This ease of use, accessibility, and integration makes digital applications a more reliable and efficient option for modern consumers, addressing many of the limitations associated with traditional methods.

2.4 Previous works

2.4.1 Epicurious

Figure 2.5 – Epicurious Logo

Epicurious is a prominent mobile recipe management application that offers an extensive array of recipes sourced from top cooking publishers, including Epicurious, Bon Appétit, Gourmet, and HarperCollins. With over 35,000 recipes available, users can explore a diverse range of culinary delights [18]. The platform also features recipe reviews and ratings from home cooks, providing valuable insights into each dish's taste and preparation. Additionally, Epicurious offers weekly content updates to keep users inspired with fresh culinary ideas. For those seeking visual guidance, the application includes demonstration videos showcasing step-by-step instructions for preparing various recipes, enhancing the cooking experience for users of all skill levels [18]. Epicurious boasts a clean UI design with easy navigation within the application (Figure 2.6 shown the Epicurious Home, Figure 2.7 shown the Epicurious Search Recipe, Figure 2.8 shown the Epicurious Saved Recipes). This application primarily focuses on providing users with a vast array of recipe choices, with content updated on a weekly basis. Given that Epicurious has been a brand focused on food and cooking-related topics since 1995, its extensive database of food recipes is considered more trustworthy by users [19]. As a result, it has garnered a significant user base, with 6.2 million visits recorded in March 2024 [20].

However, despite its strengths, Epicurious has some limitations. While the application is free to download, users are required to subscribe to access most of its features. This subscription-based model includes monthly and yearly options, granting users access to essential features such as viewing recipes, utilizing the shopping list feature, employing the smart kitchen timer, toggling ingredients, and more [18] [21]. This shift towards a subscription model may limit accessibility for some users and underscores the platform's emphasis on monetization.

Furthermore, the absence of a comprehensive grocery management feature in Epicurious presents a significant drawback for users seeking a holistic solution to organize their meals and manage their shopping effectively. This limitation can lead to several consequences for the user, such as increased time and effort spent on grocery shopping, inconsistent dietary habits, and potential food wastage [22].

To address these limitations, a more effective approach could involve partially restricting features for new users rather than limiting access to 90% of the features without a subscription. This approach is particularly beneficial for individuals who are unfamiliar with the app and may be hesitant to commit to a subscription without experiencing its reliability firsthand. People are generally reluctant to pay for services that lack transparency or that they haven't fully explored. By offering limited access to certain features, new users can gain a better understanding of the application's functionality and the quality of its content before deciding to subscribe.

Additionally, implementing comprehensive grocery management features, allowing users to seamlessly transition from recipe selection to grocery list creation, would significantly enhance the application's utility. This could include features such as automatic ingredient list generation from selected recipes, the ability to modify quantities based on serving sizes, and integration with online grocery platforms for easy shopping.

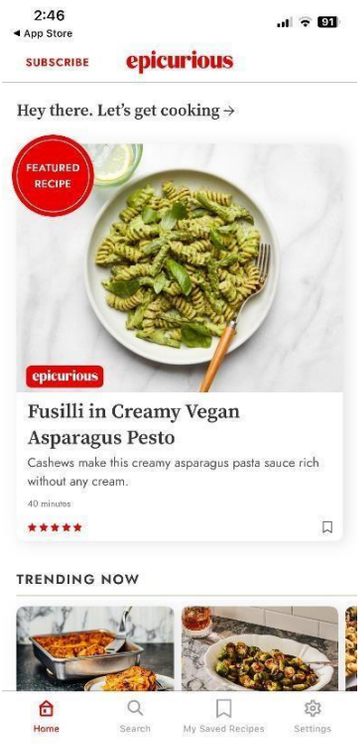


Figure 2.6 – Epicurious Home

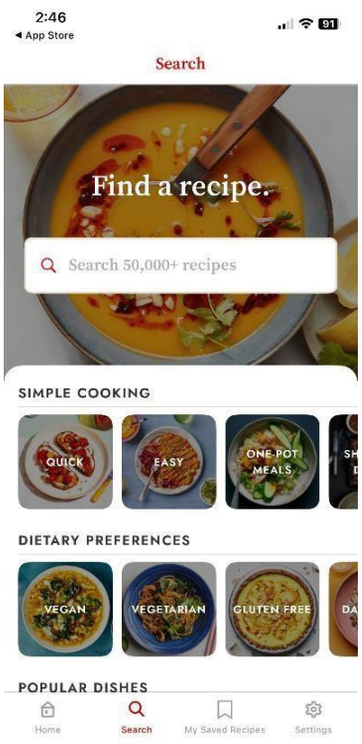


Figure 2.7 – Epicurious Search Recipe

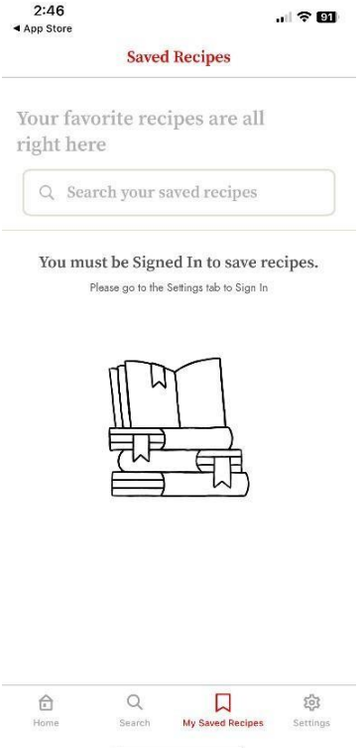


Figure 2.8 – Epicurious Saved Recipes

2.4.2 Cookpad

Figure 2.9 – Cookpad Logo

Cookpad is an application that enables users to share homemade recipes and discover culinary delights from around the globe [23]. Founded in Japan in 1997, Cookpad has become an invaluable resource for cooking enthusiasts, offering access to over 60,000 recipes curated by both professional chefs and amateur cooks [24]. Whether users are seeking inspiration for a special occasion or simply looking to expand their culinary repertoire.

One of Cookpad's notable features is the "Inspiration" tab on the home page (Figure 2.10 shown the Cookpad Home page), which showcases recipes recommended based on seasonal ingredients. This feature efficiently saves users time by providing recipes that align with the ingredients they have on hand, encouraging culinary exploration and reducing food waste. Additionally, Cookpad incorporates a "Follow" feature [23], as illustrated in Figure 2.11 shown the Cookpad Follow Feature, enabling users to subscribe to other users' profiles. This functionality sends notifications to users whenever the followed author publishes a new recipe post, serving as a source of inspiration for trying out new culinary ideas.

Furthermore, Cookpad offers a "Bookmark" recipe feature (Figure 2.10 shown the Cookpad Bookmark Recipe), allowing users to save or favorite recipes for future reference [23], streamlining their experience with the application. Each recipe includes step-by-step cooking instructions and ingredient scaling, ensuring that users have all the necessary information before attempting the recipe. These features contribute to the user-friendly nature of Cookpad, making it a valuable tool for home cooks of all skill levels.

Nevertheless, Cookpad has some limitations that affect its overall utility. The application lacks a grocery listing feature, which prevents users from seamlessly adding ingredients directly from chosen recipes to a shopping list. This limitation impacts user experience and requires additional time and effort for managing grocery lists. Integrating a grocery shopping list feature would streamline the process from recipe selection to ingredient acquisition, significantly enhancing the application's functionality.

Another drawback is the lack of nutrition information for ingredients in the recipes. This absence can be a significant issue for users following specific diets or those concerned about their nutritional intake. By integrating external APIs that provide nutrition details for each ingredient, Cookpad could not only benefit users' health but also enrich the overall user experience by providing valuable nutritional insights.

Furthermore, while Cookpad excels in recipe sharing and discovery, it does not offer direct integration with online grocery shopping services. This gap in functionality means users cannot seamlessly transition from recipe selection to ingredient purchase within the application, creating a disconnect in the overall cooking preparation process.

To address these limitations, Cookpad could consider implementing a comprehensive grocery management system that allows users to add ingredients from recipes directly to a shopping list. This feature could be further enhanced by integrating with online grocery platforms, enabling users to purchase ingredients directly through the application. Additionally, incorporating nutritional information for recipes would provide users with a more holistic approach to meal planning and dietary management.

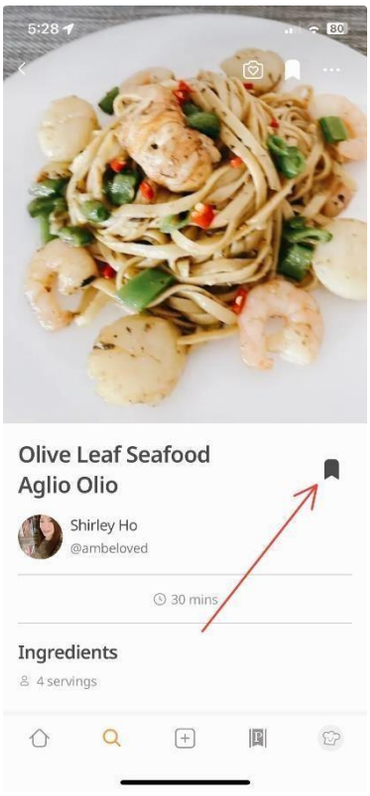


Figure 2.10 – Cookpad Bookmark Recipe

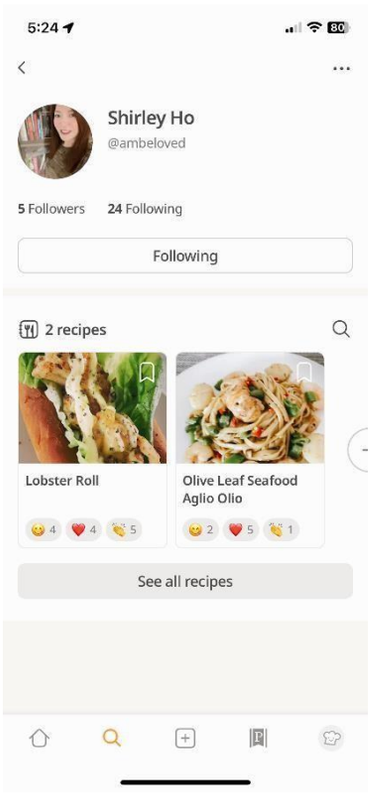


Figure 2.11 – Cookpad Follow Feature

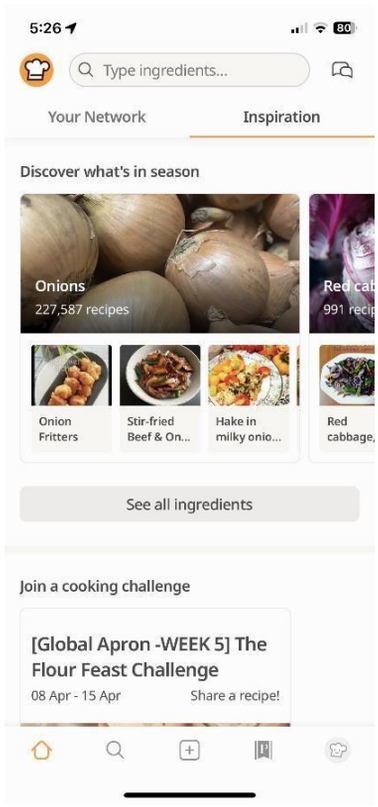


Figure 2.12 – Cookpad Home page

2.4.3 Cookbook

Figure 2.13 – CookBook Logo

CookBook offers a comprehensive suite of features designed to streamline the culinary journey. With four main functionalities - Add, Manage, Cook, and Plan - CookBook transforms recipe management and meal preparation into a seamless and delightful experience [25].

The “Add” feature enables effortless recipe input, allowing users to import recipes from various sources such as online platforms, cookbooks, handwritten notes, and magazines. Users can also customize recipes with personal notes and tips, creating a personalized culinary archive tailored to their tastes.

The “Manage” section allows users to efficiently organize their recipes into customizable categories and tags, facilitating easy navigation and retrieval. Whether it's creating folders for baking experiments or curating a collection of quick 30-minute meals, CookBook empowers users to curate their culinary library with precision and ease [25]. The “Cook” functionality elevates the cooking experience with intuitive tools and interactive features, from setting multiple timers for different recipe elements to having recipes read aloud while cooking. Moreover, the application seamlessly adjusts serving sizes, portions, and measurements, eliminating the hassle of manual conversions.

The “Plan” feature includes generating grocery lists, tracking pantry inventory, and identifying recipes with shared ingredients to optimize shopping trips and minimize waste. CookBook also allows users to export, share, and print recipes with ease, ensuring seamless integration into their culinary routine.

However, CookBook has some limitations. One of the main drawbacks is the manual process required to import recipes from external sources, which can be cumbersome and time-consuming for users, especially students with busy schedules. Having to copy URLs from supported websites or input recipes manually may deter

users from utilizing the application to its full potential. To address this issue, CookBook could implement a feature that allows users to create their own recipe library or database within the application, or enable one-click imports from popular recipe websites.

Another limitation of CookBook is its subscription-based plans, which may pose a barrier to access for users who are reluctant to commit to ongoing payments. While subscription models offer benefits such as regular updates and premium features, they may deter budget-conscious users or those hesitant to invest in long-term commitments. To mitigate this, CookBook could consider offering a freemium model with basic features available for free, supplemented by optional premium subscriptions for users seeking advanced functionalities. This approach would provide users with flexibility and choice, allowing the users to access essential features without financial constraints while offering premium options for users desiring additional benefits.

2.4.5 Features Comparison between Existing Cook Application and project deliverable

Cook Applications / Features	Epicurious	CookPad	Kitchen Stories	Project Deliverable
Recipe Browsing	✓	✓	✓	✓
Recipe Sharing	✗	✓	✓	✓
Shopping List	✗	✗	✗	✓
Personalized Shopping list	✓	✗	✗	✓
Cooking Instructions	✓	✓	✓	✓
Cooking demonstration(video)	✓	✗	✓	✓
Advertisements	✗	✓	✗	✗
Subscription	✓	✓	✓	✗
Free-of-charge	✗	✓	✓	✓

Table 2.3 Existing Application Features Comparison

The table provides a comparison of features between existing cooking applications (Epicurious, CookPad, and Kitchen Stories) and the project deliverable in this project.

The comparison covers several key features:

- **Recipe Browsing:** All four applications, including the project deliverable, offer recipe browsing functionality.
- **Recipe Sharing:** CookPad and Kitchen Stories provide recipe sharing features, which are also included in the project deliverable. However, Epicurious does not offer this feature.
- **Shopping List:** The project deliverable includes a shopping list feature, which is not available in the other three applications.
- **Personalized Shopping list:** The project deliverable offers a personalized shopping list, a feature not available in the other applications.

- **Cooking Instructions:** All four applications provide cooking instructions for their recipes.
- **Cooking demonstration (video):** Epicurious and Kitchen Stories include cooking demonstration videos, a feature also available in the project deliverable. However, CookPad does not offer this feature.
- **Advertisements:** CookPad includes advertisements, while Epicurious and Kitchen Stories do not. The presence of advertisements in the project deliverable is not specified.
- **Subscription:** Epicurious and CookPad require a subscription, while Kitchen Stories does not. The project deliverable does not require a subscription.
- **Free-of-charge:** Kitchen Stories is free-of-charge, while the project deliverable is also available without any cost. The pricing models of Epicurious and CookPad are based on monthly subscription.

In summary, the table highlights the unique features of the project deliverable, such as the shopping list and personalized shopping list, which are not available in the other cooking applications. It also demonstrates that the project deliverable offers a comprehensive set of features, including recipe browsing, sharing, cooking instructions, and video demonstrations, while remaining free-of-charge for users.

CHAPTER 3: System Approach

The Software Development Life Cycle (SDLC) is a structured, systematic process designed to deliver software applications efficiently and effectively while ensuring high quality and minimizing risk [26]. It outlines a series of phases that break down the software development process into manageable tasks, facilitating assignment, completion, and progress tracking. The SDLC framework encompasses several stages, including planning, design, implementation, testing, deployment, and maintenance, each contributing to the overall success of the project.

- **Planning:** In this initial phase, activities such as cost-benefit analysis, scheduling, resource estimation, and allocation are performed. The focus is on estimating project costs, creating schedules, and developing a comprehensive plan to achieve project objectives.
- **Design:** The design phase involves analyzing requirements and identifying optimal solutions for software creation. This includes creating detailed architectural designs, defining data structures, and designing user interfaces. The goal is to establish a clear roadmap for implementing the software solution effectively.
- **Implementation:** During this phase, the development team begins coding the software based on the designs and specifications from the previous phases. The process is broken down into smaller, manageable tasks, allowing for incremental progress. Regular feedback and adjustments are made to ensure alignment with the desired outcome.
- **Testing:** In the testing phase, both automated and manual methods are used to identify and fix any bugs in the software. This quality assurance process ensures the software meets customer requirements and functions smoothly. Testing is often conducted concurrently with development to maintain software integrity throughout its lifecycle.
- **Deployment:** The deployment phase involves transitioning the software from testing environments to the production environment where users access it. This includes packaging the latest build, configuring the environment, and installing the software. Maintaining separate build and production environments ensures users can continue using the software uninterrupted while updates are made.

- **Maintenance:** In the maintenance phase, the development team addresses tasks such as bug fixes, resolving customer issues, and managing software changes. Monitoring system performance, security, and user experience helps identify opportunities for further enhancement, ensuring the software remains reliable and effective over time.

Various SDLC models offer distinct methodologies for achieving software development success. Examples include the Agile Model, Prototype Model, Waterfall Model, and others, each tailored to different project needs, team dynamics, and organizational goals. These models provide specific advantages and approaches suited to various development scenarios.

3.1.1 Agile Model

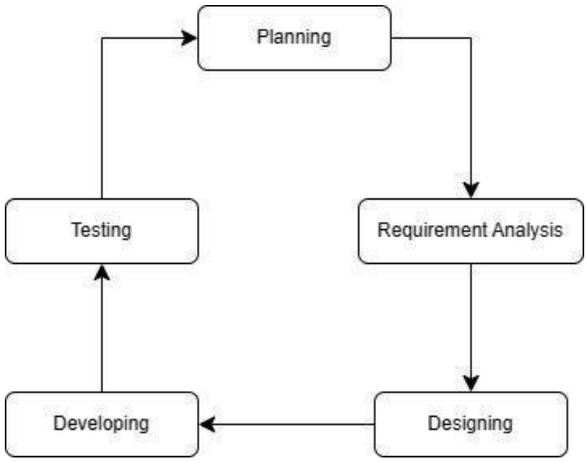


Figure 3.1.1 – Agile Model

The Agile Model is characterized by its iterative and incremental approach to software development. It emphasizes collaboration between cross-functional teams, frequent customer involvement, and the delivery of working software in short development cycles known as sprints. Agile methodologies, such as Scrum and Kanban, prioritize flexibility, adaptability, and responsiveness to change.

In the Agile Model, requirements and solutions evolve through the collaborative effort of self-organizing teams. Regular feedback from stakeholders, including customers and end-users, informs the prioritization of features and guides subsequent development iterations. This iterative process allows for continuous improvement and refinement of the software product.

One of the key strengths of the Agile Model is its ability to accommodate changing requirements and priorities, making it well-suited for projects with evolving or uncertain business needs [27]. By breaking down complex projects into manageable increments, Agile enables teams to deliver value to customers more frequently, reducing the time to market and increasing overall project transparency.

However, while the Agile Model offers many benefits, including increased customer satisfaction and faster delivery of software, it also presents challenges. The iterative nature of Agile development can sometimes lead to scope creep if not managed effectively. Moreover, maintaining a balance between delivering new features and

maintaining technical excellence requires discipline and ongoing communication within the development team.

3.1.2 Prototype Model

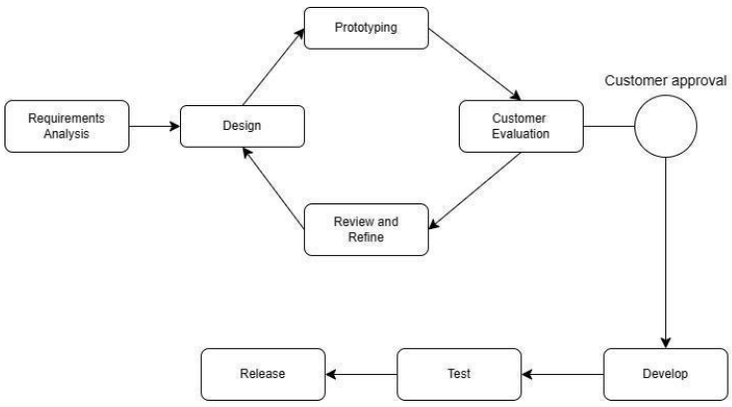


Figure 3.1.2 – Prototype Model

The Software Prototyping Model involves creating an initial prototype of the software product before developing the final version. This prototype serves as a working model to gather feedback and refine the design. It's particularly useful for projects with undefined or changing requirements, reducing the risk of developing a product that doesn't meet user needs [28]. Prototyping can use methods like throwaway, evolutionary, or incremental, and can be combined with other methodologies like Agile. Prototyping focuses on refining working models based on user feedback, making it popular for systems with complex user interfaces. It captures customer requirements early, allowing for valuable feedback and aiding in understanding user expectations directly from sales. However, prototypes may not contain the exact logic of the final software.

3.1.3 Iterative Model

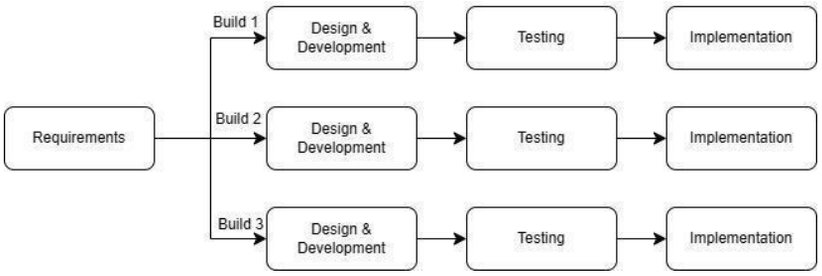


Figure 3.1.3 – Iterative Model

In the Iterative model, the development process commences with the implementation of a basic version of a small subset of software requirements. This initial version is then iteratively refined and enhanced until the entire system is fully implemented and prepared for deployment.

Unlike traditional models that require a complete specification of requirements upfront, the Iterative model starts with a partial specification [29]. Development begins by implementing a portion of the software, which is then reviewed to uncover additional requirements. This iterative cycle is repeated, resulting in a new version of the software at the conclusion of each iteration.

The Iterative and Incremental Software Development Life Cycle (SDLC) Model offers several advantages and disadvantages in the realm of software development. One notable advantage is the early detection of flaws in the system [29]. By providing a working model at an early stage, this approach facilitates the identification of functional or design flaws, allowing for corrective measures to be taken within budget constraints. This early detection helps mitigate risks and ensures a higher-quality end product.

However, alongside its benefits, the Iterative and Incremental SDLC Model also presents challenges. One significant disadvantage is its resource requirements. This model may demand more resources due to its iterative nature, requiring additional management attention. As the development progresses through multiple iterations, the need for skilled resources and ongoing management oversight becomes more pronounced, potentially increasing project complexity and resource demands.

3.1.4 Methodology Chosen – Prototype Model

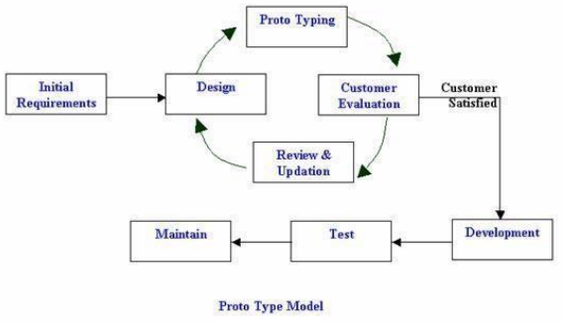


Figure 3.1.4 – Chosen Model

This project will be implemented based on the Prototype Model as its Software Development Life Cycle (SDLC) methodology. The Prototype Model involves creating an initial prototype of the software product before developing the final version. This approach aligns perfectly with the objectives and requirements of this Flutter food-recipe application project.

To support this, implementing the Prototype Model is particularly suitable for this project because it allows for addressing the dynamic nature of user requirements in the realm of food recipes. By creating prototypes that simulate key functionalities and user interactions, valuable feedback can be gathered early in the development process and incorporated into subsequent iterations.

One of the main benefits of adopting the Prototype Model for this project is its ability to reduce the risk of developing a product that doesn't meet user needs [28]. Through iterative prototyping, the design decisions can be validated, ensuring alignment with user expectations, and the application can be refined to enhance usability and functionality.

Additionally, the Prototype Model enables a focus on refining working models based on user feedback, which is crucial for a system with complex user interfaces like a recipe application. By embracing an iterative approach to development, the user experience can be iteratively improved, potential issues identified, and necessary adjustments made to deliver a high-quality application [28].

While the Prototype Model may have its limitations, such as prototypes not containing the exact logic of the final software, its benefits far outweigh the drawbacks for this project. By implementing the Prototype Model, the project can leverage its iterative nature to iteratively refine the application, gather valuable feedback from users, and ultimately deliver a user-centric Flutter food-recipe application that meets the needs of the target audience.

3.2 Software and Hardware Involved

3.2.1 Hardware Components

The hardware involved in this project includes various components essential for the development and deployment of the mobile application. This comprises a computer equipped with the necessary software development tools and environments, serving as the primary workstation for mobile application development using Flutter. The computer should meet the minimum system requirements for the Flutter framework and associated tools. Additionally, the application is designed to run specifically on Android mobile devices. Therefore, the project will utilize hardware devices such as a laptop for development purposes. The laptop will play a crucial role in developing the application optimized for Android devices.

Description	Specifications
Model	Lenovo Legion Slim 5
Processor	AMD Ryzen 7 7840HS
Operating System	Windows 11 Pro
Graphic	NVIDEA GeForce RTX 4060 16GB
Memory	16GB DDR5 RAM
Storage	1TB NVMe SSD

Table 3.1 Specifications of laptop

Description	Specifications
Model	Galaxy Nexus
Processor	Intel Atom (x86_64)
Operating System	Android 6.0 (API level 23)
Resolution	1080 x 1920
Memory	2GB
Storage	1GB

Table 3.2 Specifications of Android Emulator

Requirement	Minimum	Recommended
x86_64 CPU Cores	4	8
Memory in GB	8	16
Display resolution in pixels	WXGA (1366 x 768)	FHD (1920 x 1080)
Free disk space in GB	4.0	52.0

Table 3.3 Hardware Requirements of Flutter Framework

3.2.2 Software Components

The software component that involved in this project are Android Studio, Visual Studio Code, and Firebase.

Android Studio, developed by Google, is a powerful integrated development environment (IDE) tailored for Android application development [30]. It features a powerful code editor, an Android emulator for testing across different devices, support for Gradle and Maven build systems, project setup wizards, and a Dracula theme for an enhanced coding experience. These tools enable developers to create high-quality Android applications efficiently and effectively [31].

Other than that, Visual Studio Code, developed by Microsoft, is a popular and versatile source code editor that has gained widespread adoption since its launch in 2015 [32]. Known for its lightweight yet powerful features, it offers extensive customization options, including IntelliSense and built-in Git integration, which enhance productivity across different platforms. The rich ecosystem of extensions available in the VS Code marketplace further extends its functionality, covering areas

such as language support, debugging, themes, and productivity tools. This extensibility allows developers to tailor their coding environment to their needs and workflows, making Visual Studio Code a valuable tool for software development.

Additionally, Firebase, developed by Google, is a comprehensive Backend-as-a-Service (BaaS) platform [33]. It provides a wide range of features and services to support mobile and web application development, including real-time databases, authentication, hosting, and cloud messaging. Firebase enables developers to build and deploy scalable applications quickly without managing infrastructure, allowing them to focus on creating exceptional user experiences.

3.3 Functional Requirements

Function ID	Functions	Description	Accessible Role
F001	Sign Up	To register a new account.	User
F002	Login	To access the system using email and password or google login.	User
F003	Reset Password	To reset the password if they forget their password.	User
F004	Logout	To ensure user safely end their session by logging out of the application.	User
F005	Browse Recipes	To browse food recipes in different categories.	User
F006	Save Recipe and Ingredients	To save specific recipes and ingredients to the grocery list.	User
F007	Search Recipe by Name	To search recipe using by different search mode.	User
F008	View and Modify Saved Recipe and Ingredient	To view and edit details of saved recipe and ingredient in grocery list.	User
F009	View Shopping List	To view the items added from the grocery list in the shopping list.	User
F010	View History	To view removed recipes and ingredients history.	User
F011	Add to Shopping List	To add ingredients to the shopping list.	User
F012	Purchase	To purchase items in shopping list.	User
F013	Modify and Delete Shopping List	To modify the quantity of the items and delete items in shopping list.	User

F014	Edit Profile Picture	To change their profile picture in profile page.	User

Table 3.4 System Function

3.4 Project Milestone

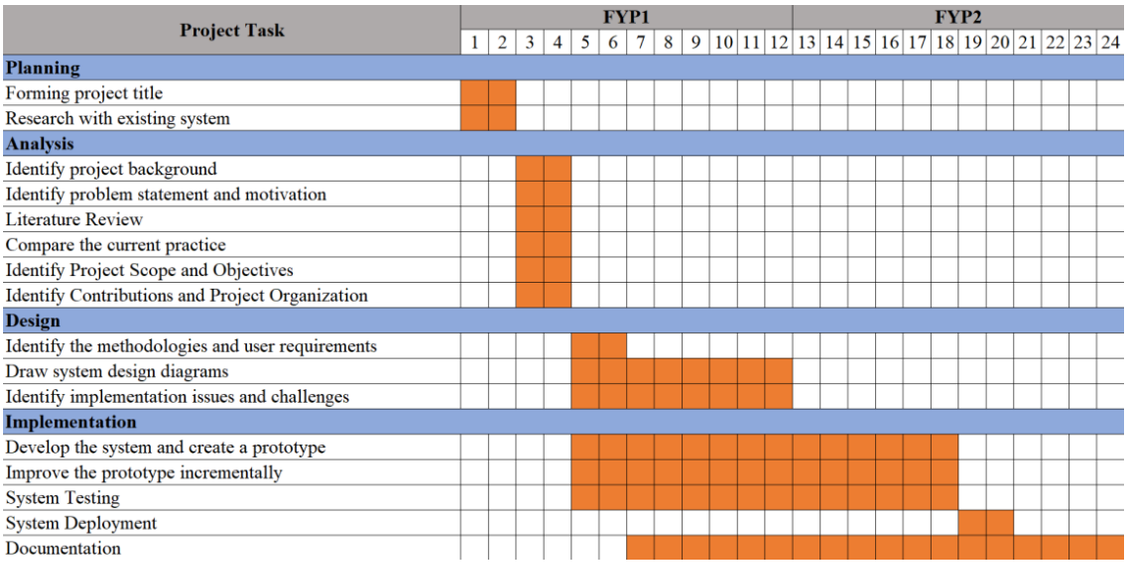


Figure 3.4 Project Milestone

The project timeline, presented as a Gantt chart, outlines the tasks and their durations for a project spanning two years, labeled as FYP1 and FYP2. The chart is divided into several key phases:

- **Planning:** This phase involves forming the project title, conducting research with existing systems, analyzing the project background, identifying the problem statement and motivation, and performing a literature review.
- **Design:** In this phase, the current practices are compared, project scope and objectives are identified, contributions and project organization are defined, and methodologies and user requirements are determined. System design diagrams are created, and implementation issues and challenges are identified.
- **Implementation:** This phase focuses on developing the system and creating a prototype, followed by incremental improvements to the prototype.
- **System Testing:** After implementation, the system undergoes thorough testing to ensure it meets the specified requirements and functions as intended.
- **System Deployment:** Once the system has been tested and validated, it is deployed to the production environment for end-users to access and utilize.
- **Documentation:** Throughout the project, comprehensive documentation is created, which includes all the necessary information about the system, its design, implementation, and usage.

The timeline indicates that the majority of the planning and design tasks are scheduled for completion in the first year (FYP1), while the implementation, testing, deployment, and documentation tasks are primarily scheduled for the second year (FYP2).

3.5 Estimated Cost

The estimated cost for the entire project lifecycle is zero. The software tools used for developing the application are Android Studio and Visual Studio Code. The project utilizes a personal laptop and an Android emulator as hardware. The Firebase Spark Plan is used for the online database service. All of these tools, hardware, and services are available free of charge.

3.6 Concluding Remark

The purpose of this chapter is to ensure the project is developed systematically by outlining the framework used to build the system and detailing its operations. It covers the prototype model utilized, the hardware and software involved, the 14 system functions, project milestones for FYP1 and FYP2, and the estimated project cost.

CHAPTER 4: System Design

4.1 System Architecture Diagram

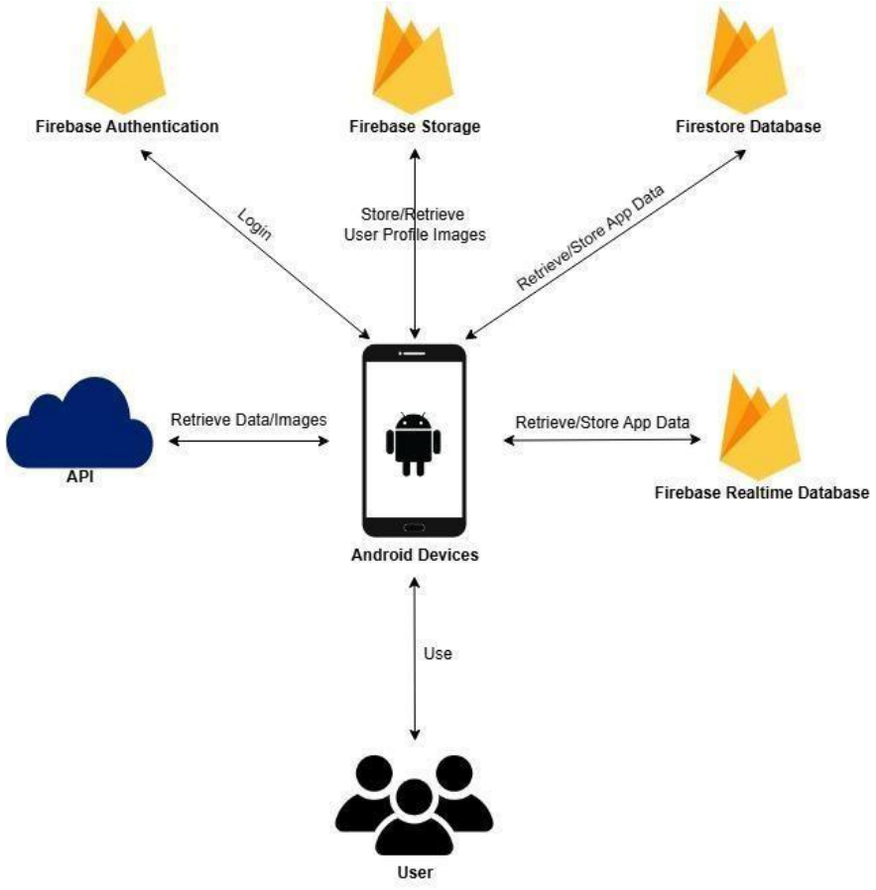


Figure 4.1 System Architecture Design

This project utilizes online services including Firebase Spark plans [34] and an Application Programming Interface (API). Firebase Authentication is implemented to handle user registration, login, logout, and password reset functionalities [35]. The authentication methods employed are email and password, as well as Google sign-in, ensuring user verification before accessing the application. The Firestore Database and Firebase Realtime Database is used for storing and retrieving system data [36]. Additionally, the API is employed to fetch the necessary data and images for the users.

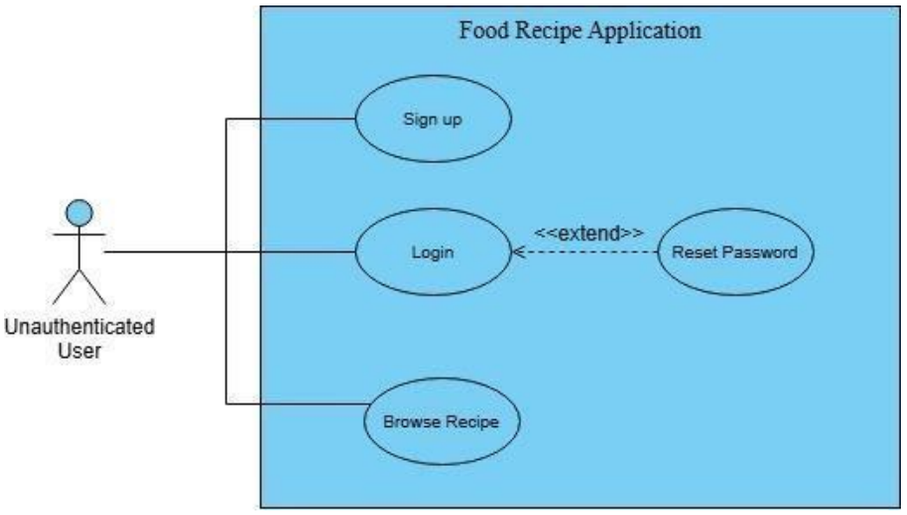


Figure 4.2 Use Case Diagram (Unauthenticated User)

4.2 Use Case Diagram

The use case diagram illustrates the system's actors and their interactions with various functionalities. The system encompasses two distinct actor types: unauthenticated users, and authenticated users. Unauthenticated users, who have not yet registered or logged in via email, password, or Google sign-in, have limited access to the system's features. Their interactions are restricted to browsing recipes, signing up for an account, and logging in. In contrast, authenticated users must verify their identity to access additional system functionalities such as Save Recipe & Ingredients, Search Recipe by Name, Add Ingredients to Shopping List and more.

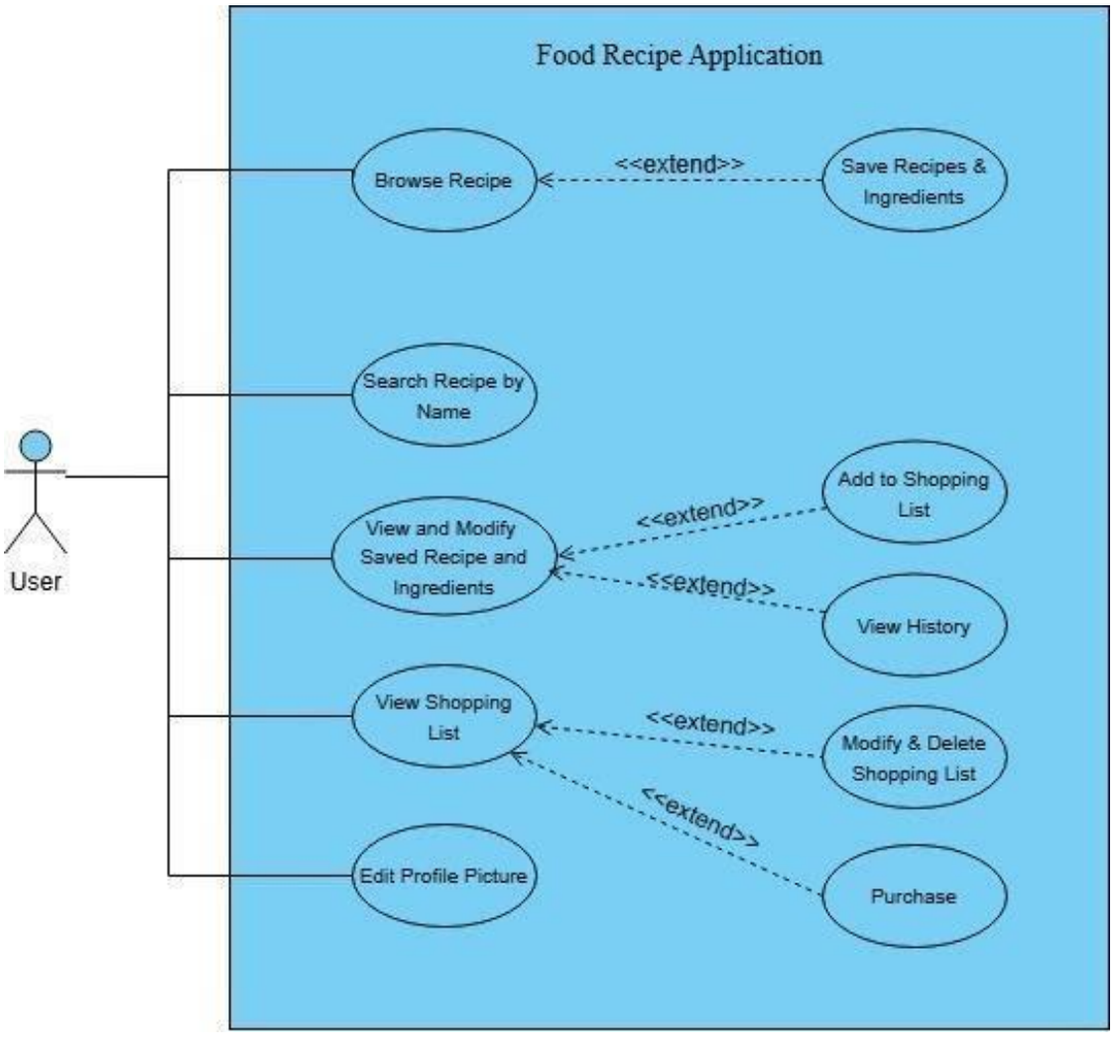


Figure 4.3 Use Case Diagram (Authenticated User)

4.3 [F001] Sign up

4.3.1 [F001] Sign up Activity Diagram

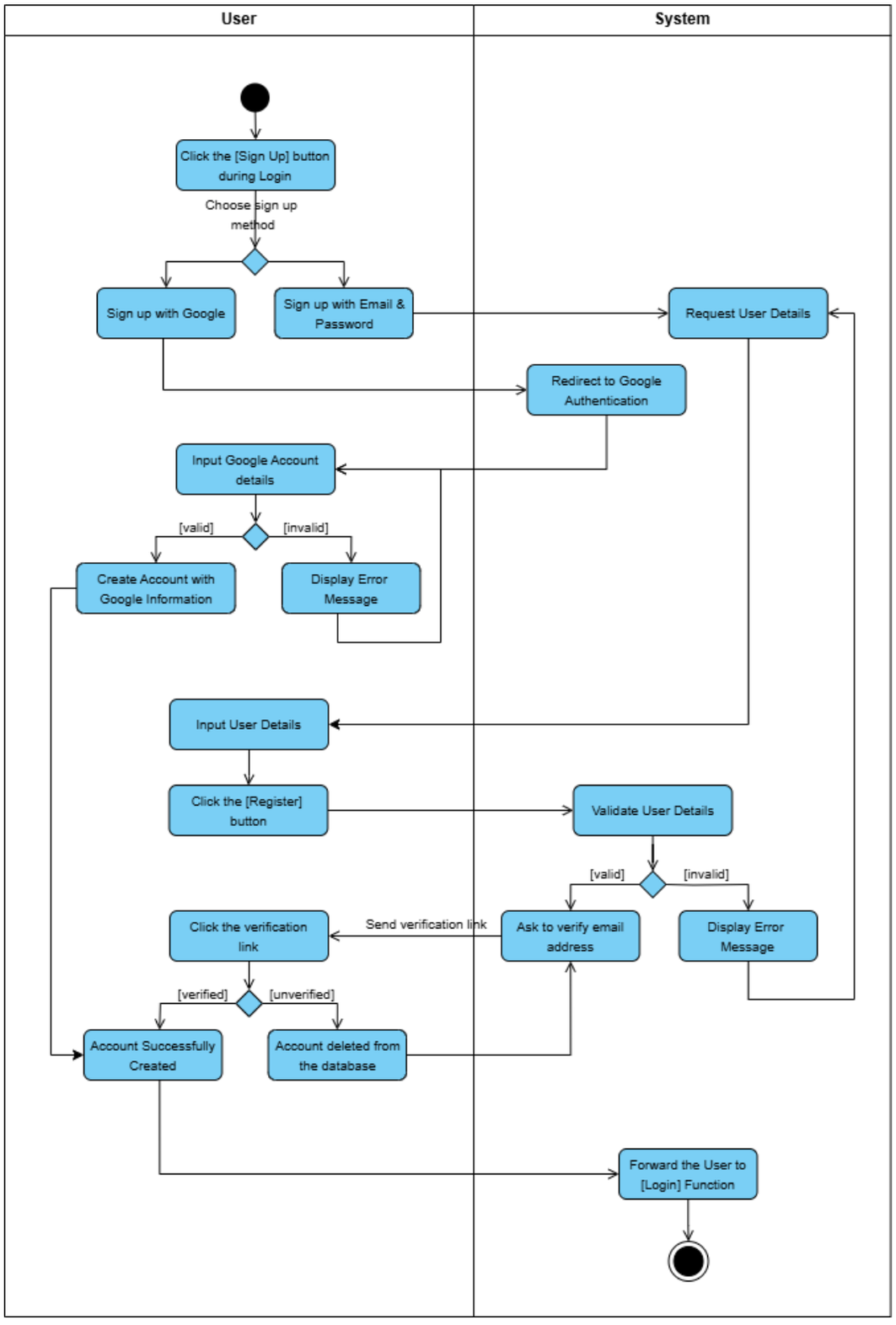


Figure 4.3.1 Sign Up Activity Diagram

4.3.2 [F001] Sign Up Module

This module is for unauthenticated users to register a new account before accessing the system. Users can choose to sign up using their Google account or through email and password registration. For Google sign-up, users are redirected to Google Authentication where they input their Google Account details. If valid, an account is created using the Google information. For email and password registration, users need to fill in personal details: username, email, and password. When the user clicks the "Register" button, it first validates all the data entered by the user. After that, the system stores the data in the database and asks the user to verify their email account. If the account is registered successfully, the user will receive a verification message in their mailbox for the users to verify the email account owner before logging in to the system. If the user doesn't verify their email within a certain timeframe, the account is deleted from the database. If a user signs up with both methods using the same email address (for example, example@gmail.com), the email address will be combined into a single account, allowing the user to sign in with both email & password and Google sign-in methods. Upon successful account creation and verification, the user is forwarded to the **[F002] Login** function where they can access the system using their new account.

4.4 [F002] Login

4.4.1 [F002] Login Activity Diagram

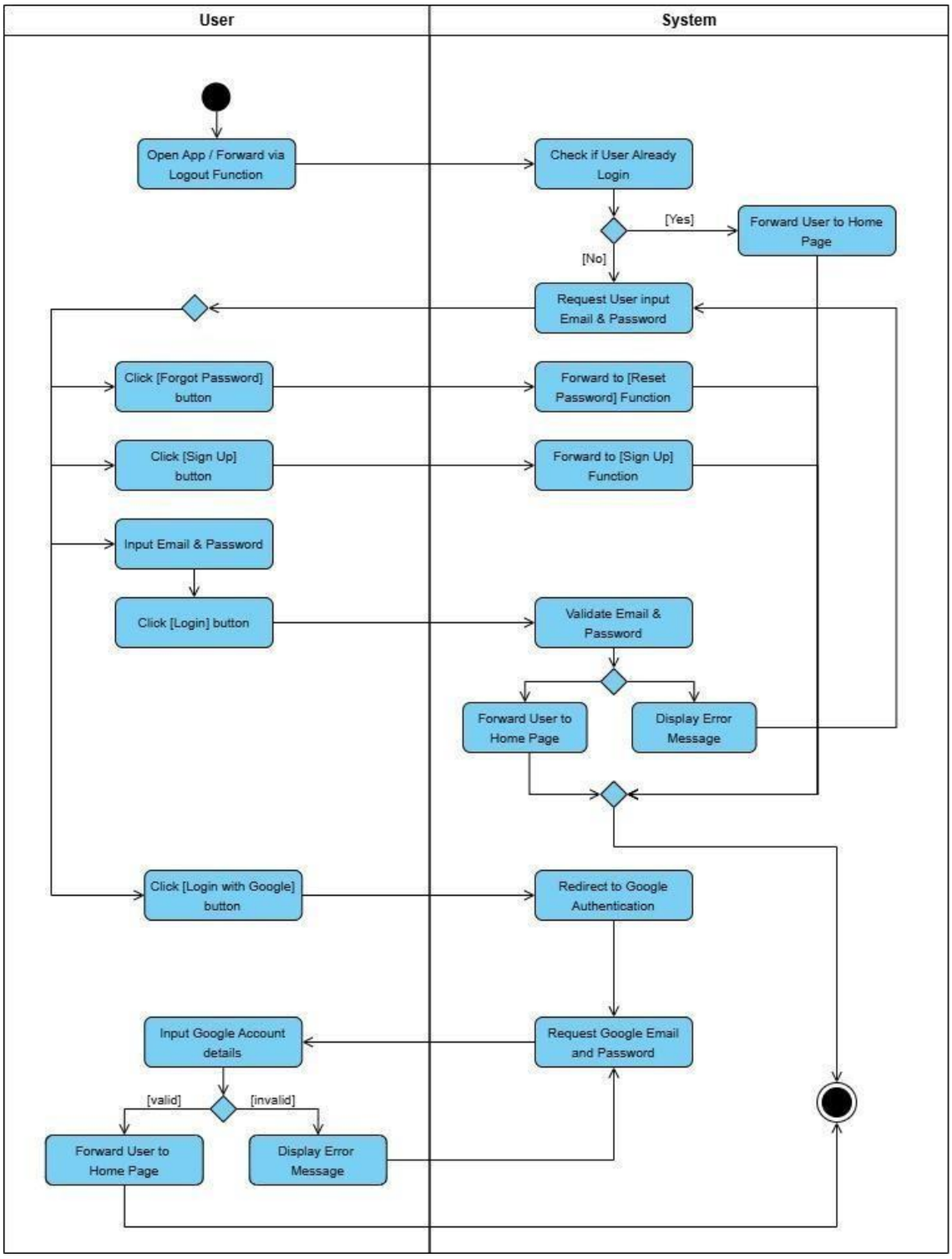


Figure 4.4.1 Login Activity Diagram

4.4.2 [F002] Login Module

This module is for users to access the system through various authentication methods. When users open the application or navigate via the logout function, the system first checks if they are already logged in. If so, they are immediately directed to the Home Page. If not, users are presented with login options. They can input their email and password, click the login button, and the system validates these credentials. If valid, users are forwarded to the Home Page; if invalid, an error message is displayed. Alternatively, users can click the "Login with Google" button, which redirects the users to Google Authentication. Users input their Google Account details, and if valid, they are forwarded to the Home Page; if invalid, an error message is shown. The module also provides options for users who have forgotten their password or need to create a new account. Clicking the "Forgot Password" button forwards users to the **[F003] Reset Password** function, while the "Sign Up" button directs the users to the **[F001] Sign Up** function.

4.5 [F003] Reset Password

4.5.1 [F003] Reset Password Activity Diagram

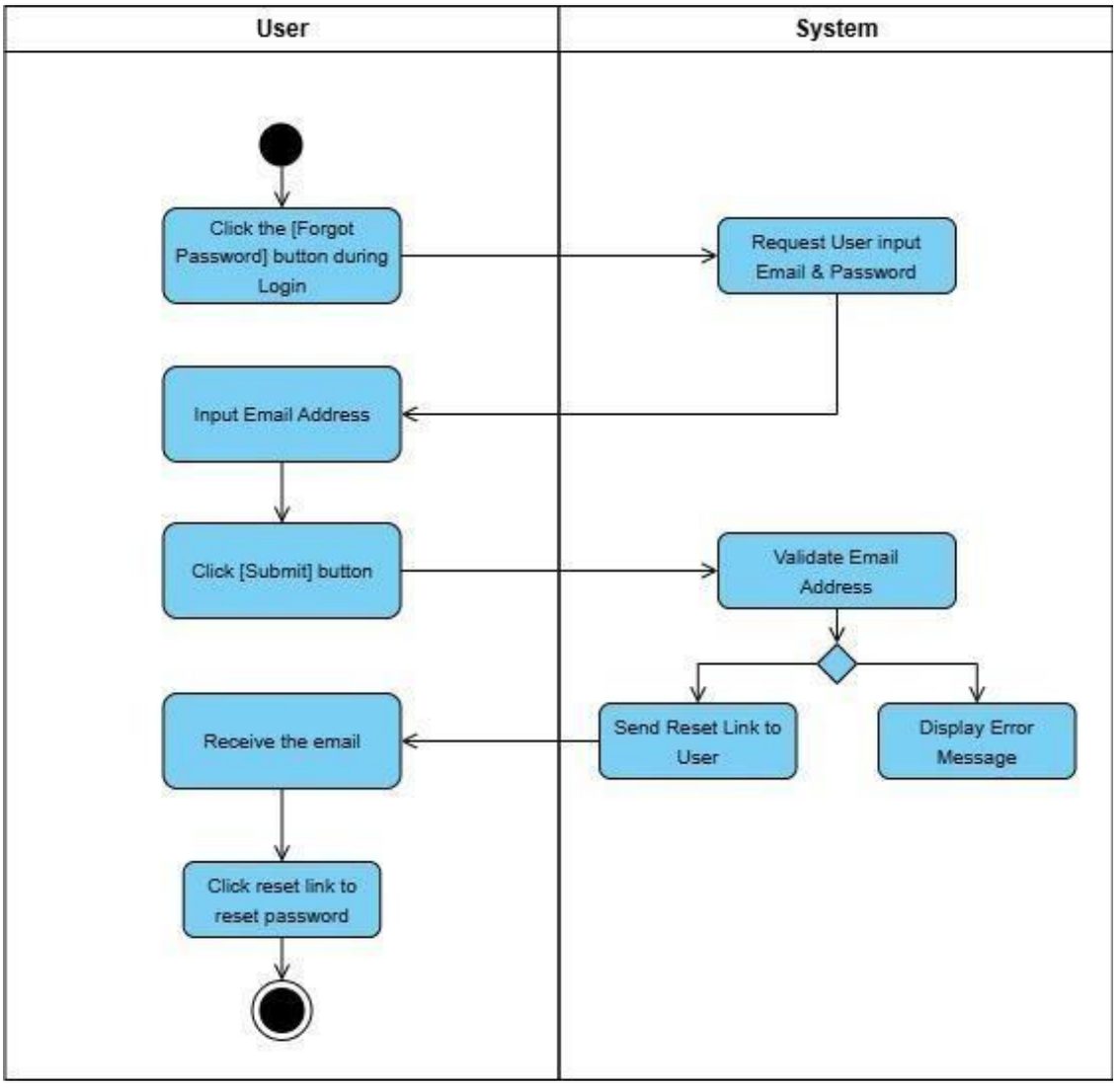


Figure 4.5.1 Reset Password Activity Diagram

4.5.2 [F003] Reset Password Module

This module is for users who have forgotten their password and need to reset it before accessing the system. When users click the “Forgot Password” button during the login process, the system requests their email address. Users input their email address and click the “Submit” button. The system then validates the email address. If the email is valid and associated with an account, the system sends a reset link to the user's email address. If the email is invalid or not found in the system, an error message is displayed. Upon receiving the email, users can click the reset link to proceed with resetting their password.

4.6 [F004] Logout

4.6.1 [F004] Logout Activity Diagram

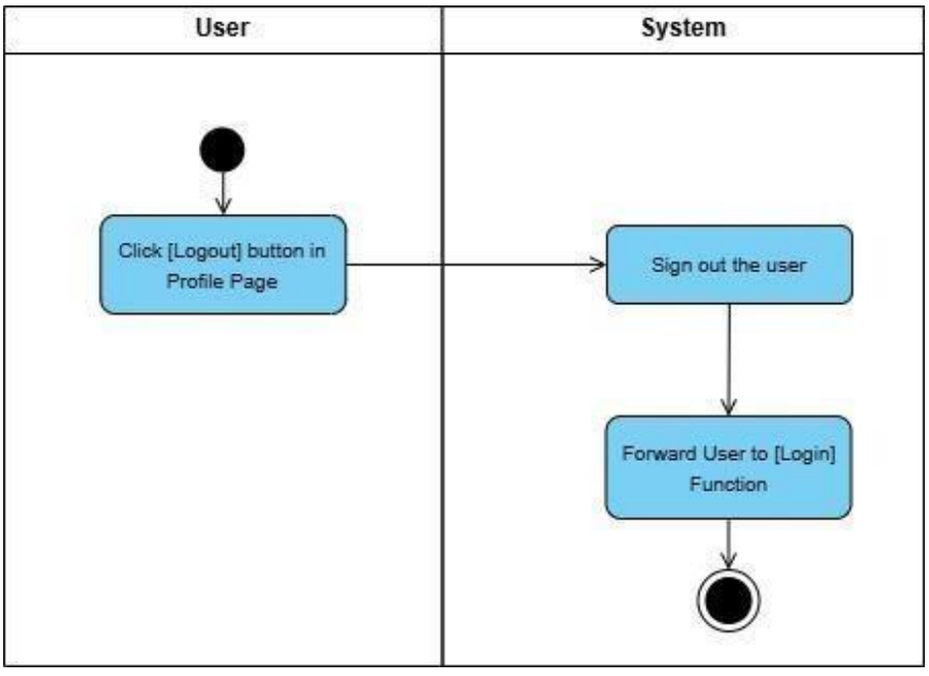


Figure 4.6.1 Logout Activity Diagram

4.6.2 [F004] Logout Module

This module is for authenticated users to securely exit the system. When users click the “Logout” button in the Profile Page, the system initiates the sign-out process. The system signs out the user, clearing their session data and any temporary credentials stored for the current session. After successfully signing out the user, the system

forwards the users to the **[F002] Login** Function, where they can choose to log in again or exit the application.

4.7 [F005] Browse Recipes

4.7.1 [F005] Browse Recipes Activity Diagram

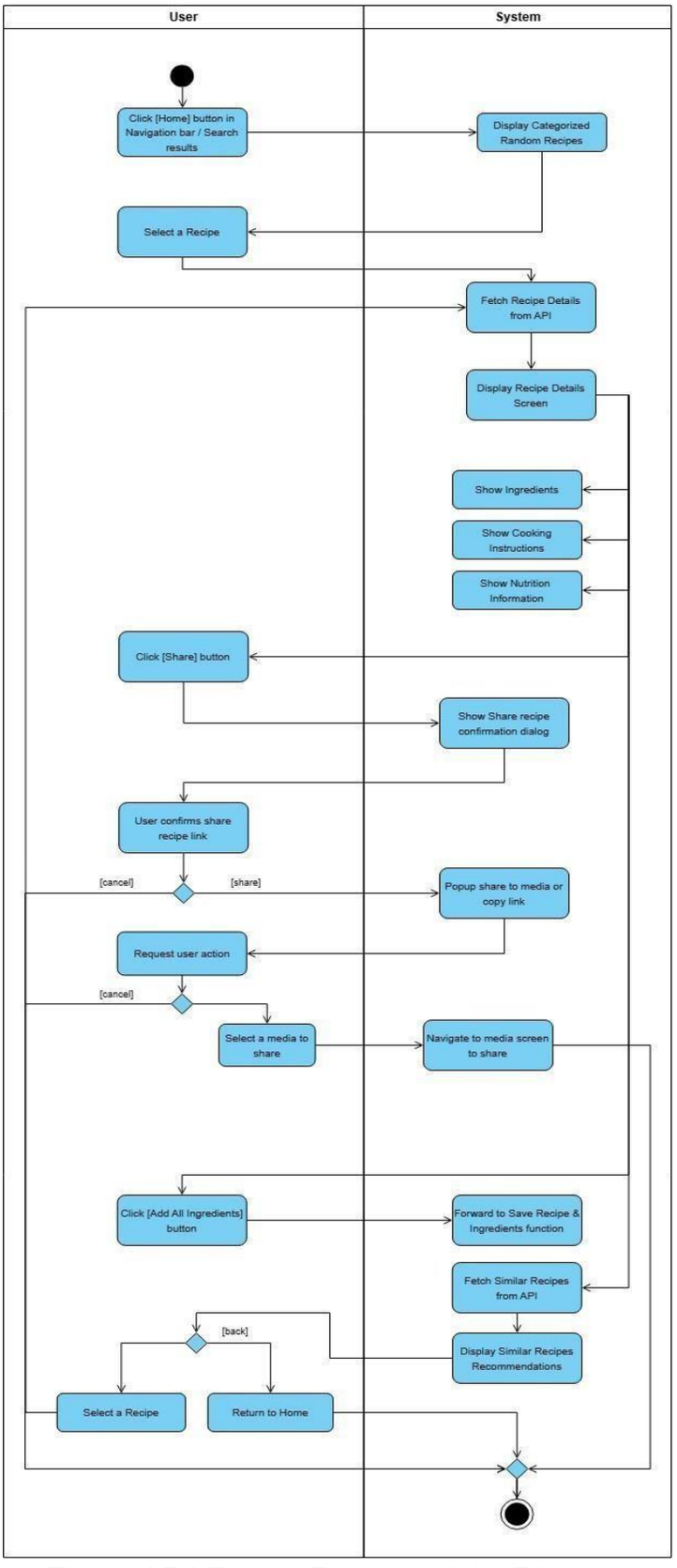


Figure 4.7.1 Browse Recipe Activity Diagram

4.7.2 [F005] Browse Recipe Module

The Browse Recipe module allows users to explore recipes through categorized displays on the home screen or in search results. Upon selecting a recipe, the system fetches and displays detailed information, including ingredients, cooking instructions, and nutrition facts. Users can share recipes via a confirmation dialog, leading to options for copying a link or sharing on social media. The "Add All Ingredients" button forwards recipe data to the **[F006] Save Recipe and Ingredients** function. The system also fetches and displays similar recipe recommendations. Users can easily navigate between viewing recipe details, exploring similar recipes, or returning to the home screen.

4.8 [F006] Save Recipe and Ingredients

4.8.1 [F006] Save Recipe and Ingredients Activity Diagram

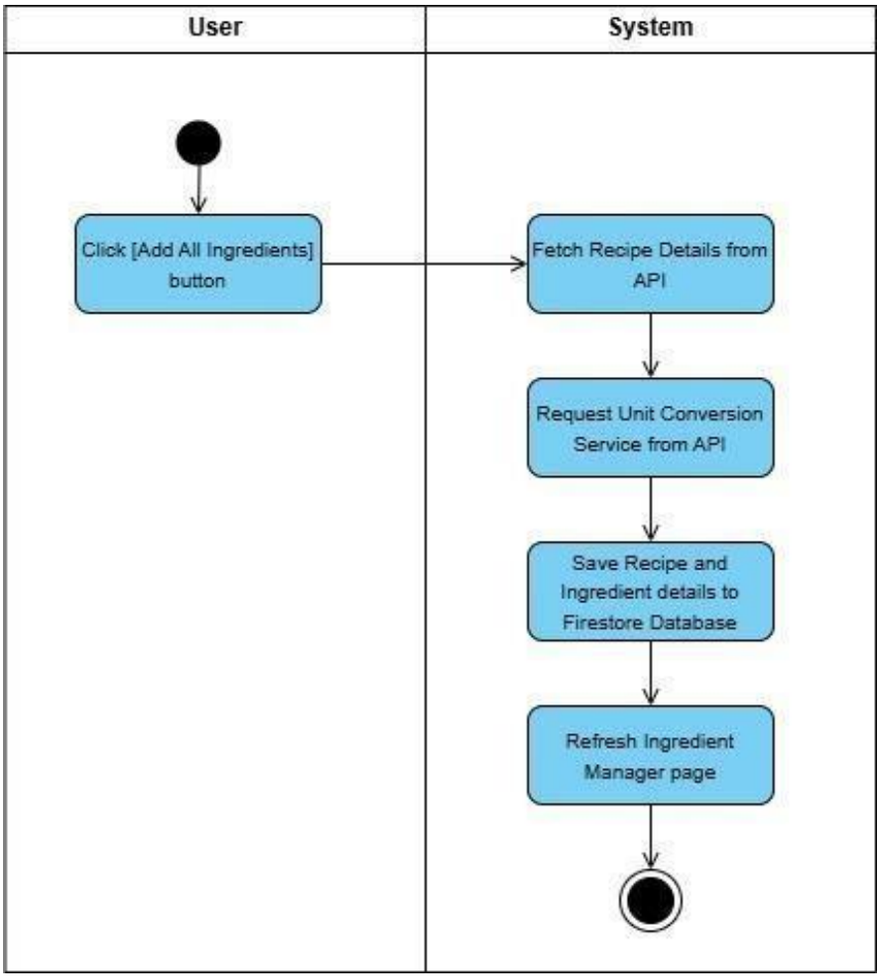


Figure 4.8.1 Save Recipe and Ingredients Activity Diagram

4.8.2 [F006] Save Recipe and Ingredients Module

The “Save Recipe and Ingredients” module simplifies storing recipes and their ingredients. When a user clicks the “Add All Ingredients” button, the system interacts with Spoonacular APIs to retrieve recipe details and convert units for consistent measurements. The recipe and ingredient information are then securely stored in the Firestore Database. The Ingredient Manager page automatically refreshes, providing immediate feedback by displaying the new or updated data. This streamlined process ensures accuracy and ease in saving culinary information for future use.

4.9 [F007] Search Recipe by Name

4.9.1 [F007] Search Recipe by Name Activity Diagram

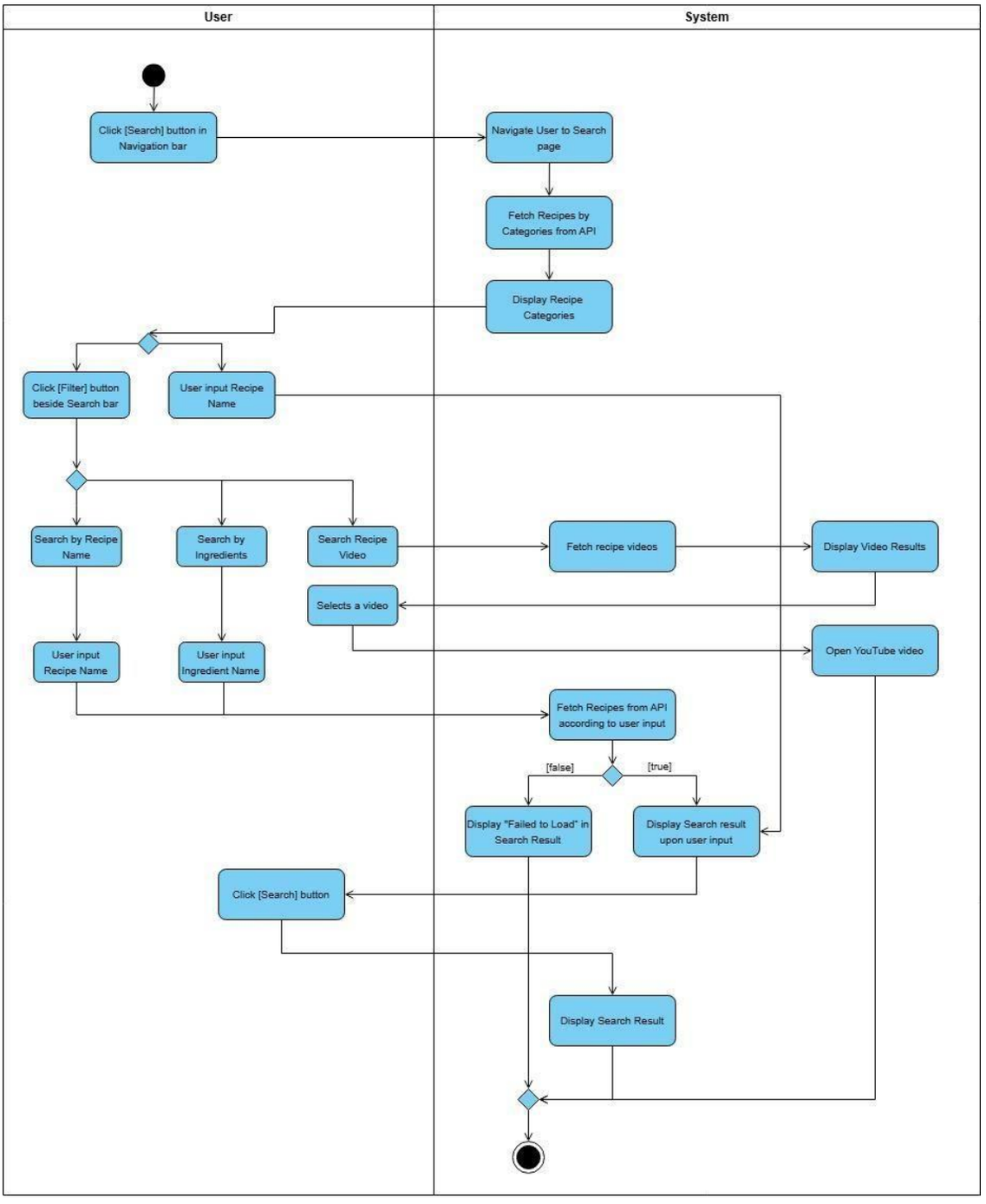


Figure 4.9.1 Search by Recipe Activity Diagram

4.9.2 [F007] Search Recipe by Name Module

The "Search Recipe" module offers versatile recipe discovery through three modes: Recipe Name, Ingredients, and Videos. Users access the search page via the navigation bar, where categorized recipes are displayed. The "Filter" button allows switching between search modes. In Recipe Name mode, users type a name for real-time results. Ingredients mode finds recipes matching entered ingredients, sorting by relevance. Video mode searches for recipe videos, which can be opened on YouTube when selected. The system queries the API as users type, displaying results accordingly. For recipes, users can view detailed information by clicking on a result. If the API fails, a "Failed to load" message appears. This streamlined module caters to various search preferences, enhancing the user's ability to find desired recipes efficiently.

4.10 [F008] View and Modify Saved Recipe & Ingredients

4.10.1 [F008] View and Modify Saved Recipe & Ingredients Activity Diagram

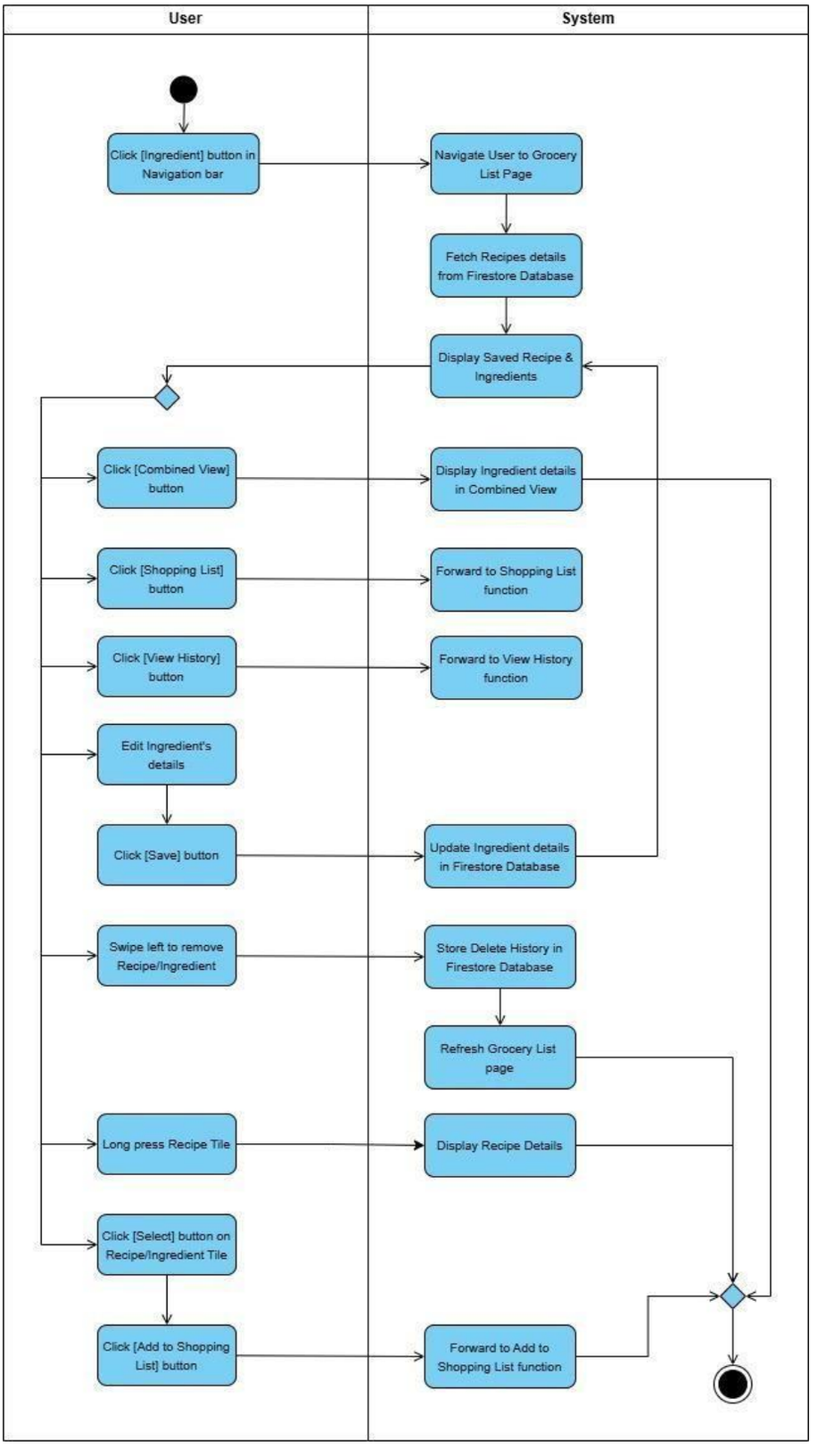


Figure 4.10.1 View and Modify Saved Recipe & Ingredients Activity Diagram

4.10.2 [F008] View and Modify Saved Recipe & Ingredients Module

The "View and Modify Recipe & Ingredients" module offers users a comprehensive and interactive platform for managing their recipes and grocery lists. The process begins when a user clicks the “Ingredient” button in the navigation bar, prompting the system to navigate to the Grocery List Page and fetch recipe details from the Firestore Database. The system then displays saved recipes and ingredients. From this main view, users can access various functions: they can view ingredient details in a combined view, access their shopping list through the **[F009] View Shopping List** function, or [**F010] View History** function to check history of removed recipes or ingredient. The module allows for detailed management, enabling users to edit ingredient details and save changes, which are then updated in the Firestore Database. Users can remove recipes or ingredients by swiping left, with the system storing deletion history and refreshing the grocery list page accordingly. Other than that, users can long-press any recipe tile in the Grocery List Page to display detailed recipe information via the sub-function of **[F005] Browse Recipe** function. Additionally, users can select specific recipes or ingredients and add to the user’s shopping list, triggering the system to forward the selection to the **[F011] Add to Shopping List** function.

4.11 [F009] View Shopping List

4.11.1 [F009] View Shopping List Activity Diagram

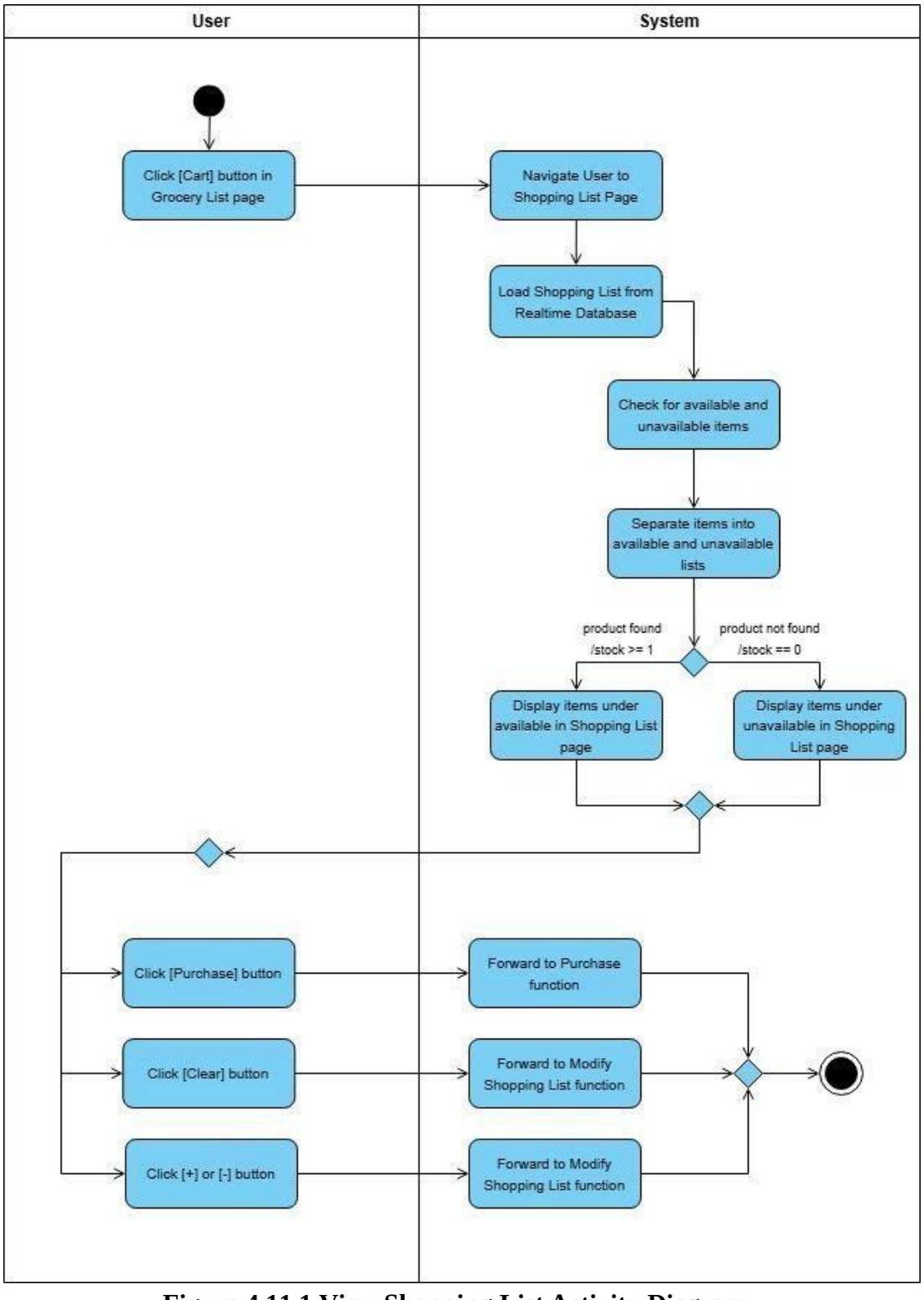


Figure 4.11.1 View Shopping List Activity Diagram

4.11.2 [F009] View Shopping List Module

The "View Shopping List" module starts when the user clicks the "Cart" button on the Grocery List page, leading to the Shopping List Page. The system loads the shopping list from the Realtime Database and separates the items into two categories: available items (in stock) and unavailable items (out of stock or not found). On the Shopping List page, the items are displayed under these two categories. The module offers three main interactive functions: the "Purchase" button forwards the user to the **[F012] Purchase** function, the "Clear" button and the "+" or "-" buttons forwards the user to the **[F013] Modify & Delete Shopping List** function.

4.12 [F010] View History

4.12.1 [F010] View History Activity Diagram

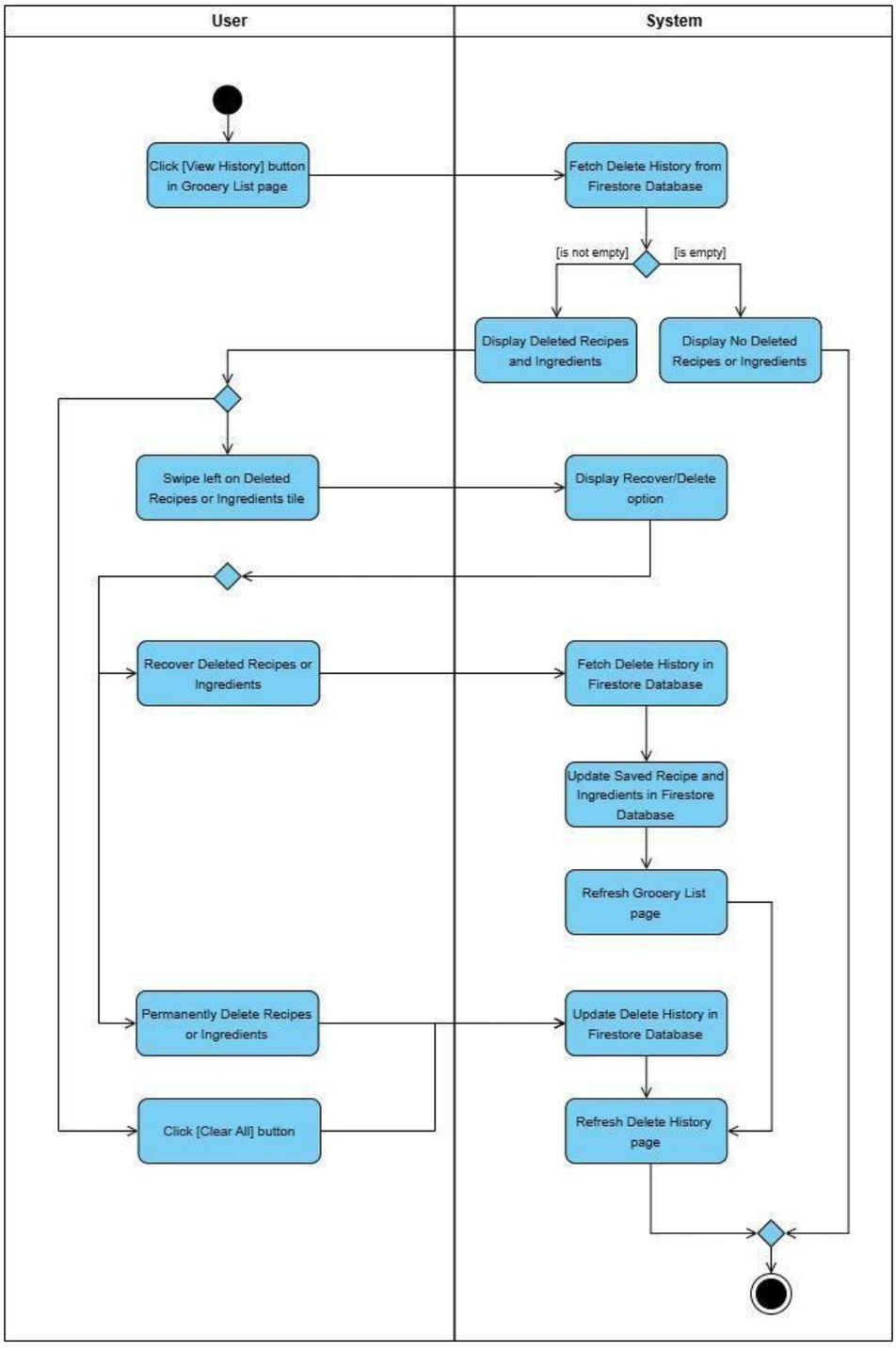


Figure 4.12.1 View History Activity Diagram

4.12.2 [F010] View History Module

The "View History" module provides users with a comprehensive system to manage their deleted recipes and ingredients. When a user clicks the “View History” button on the Grocery List page, the system fetches the delete history from the Firestore Database. The system then checks if there are any deleted items. If the history is not empty, it displays the deleted recipes and ingredients; otherwise, it shows a message indicating no deleted items. Users can interact with the deleted items by swiping left on a deleted recipe or ingredient tile. This action presents two options: recover or permanently delete. If the user chooses to recover deleted recipes or ingredients, the system fetches the delete history, updates the saved recipes and ingredients in the Firestore Database, and refreshes the Grocery List page to reflect these changes. Alternatively, if the user opts to permanently delete recipes or ingredients, the system updates the delete history in the Firestore Database. The module also includes a “Clear All” button, allowing users to remove all items from the delete history at once. Both permanent deletion and clearing all items result in refreshing the Delete History page.

4.13 [F011] Add to Shopping List

4.13.1 [F011] Add to Shopping List Activity Diagram

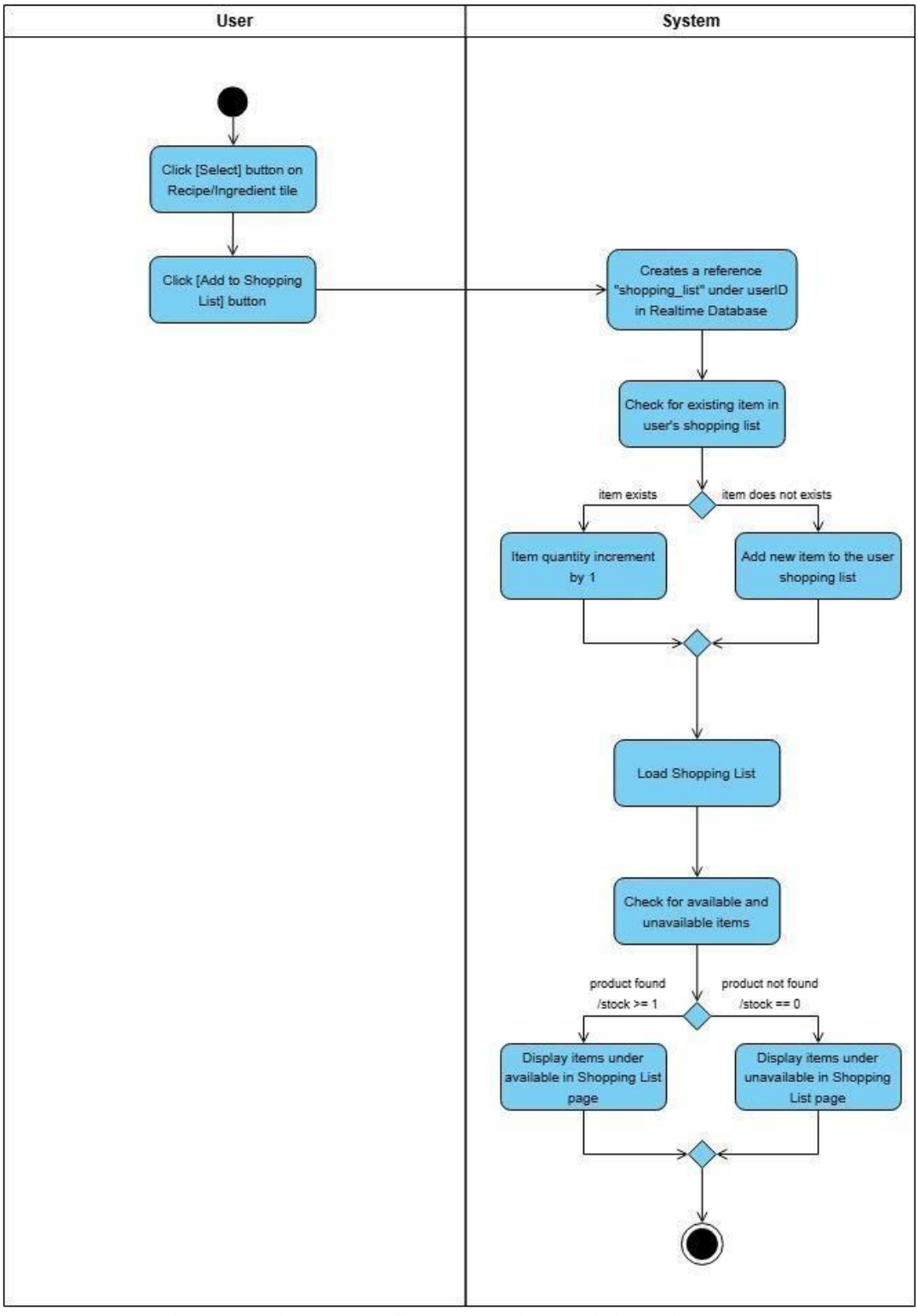


Figure 4.13.1 View Shopping List Activity Diagram

4.13.2 [F011] Add to Shopping List Module

The "View shopping list" module begins when the user clicks the "Select" button on a Recipe/Ingredient tile. Following this, the user clicks the "Add to Shopping List" button, which triggers the system to create a reference for the "shopping_list" under the user's ID in the Realtime Database. The system then checks if the selected item already exists in the user's shopping list. If the item exists, its quantity is incremented by 1. If it doesn't exist, a new item is added to the user's shopping list. After updating the shopping list, the system loads the entire list and checks for available and unavailable items. This check is based on whether the product is found in the inventory and if its stock is greater than zero. The system then displays the items on the Shopping List page, categorizing the items as either available or unavailable. Available items (products found with stock ≥ 1) are shown under the "available in Shopping List" section, while unavailable items (products not found or with stock = 0) are displayed under the "unavailable in Shopping List" section.

4.14 [F012] Purchase

4.14.1 [F012] Purchase Activity Diagram

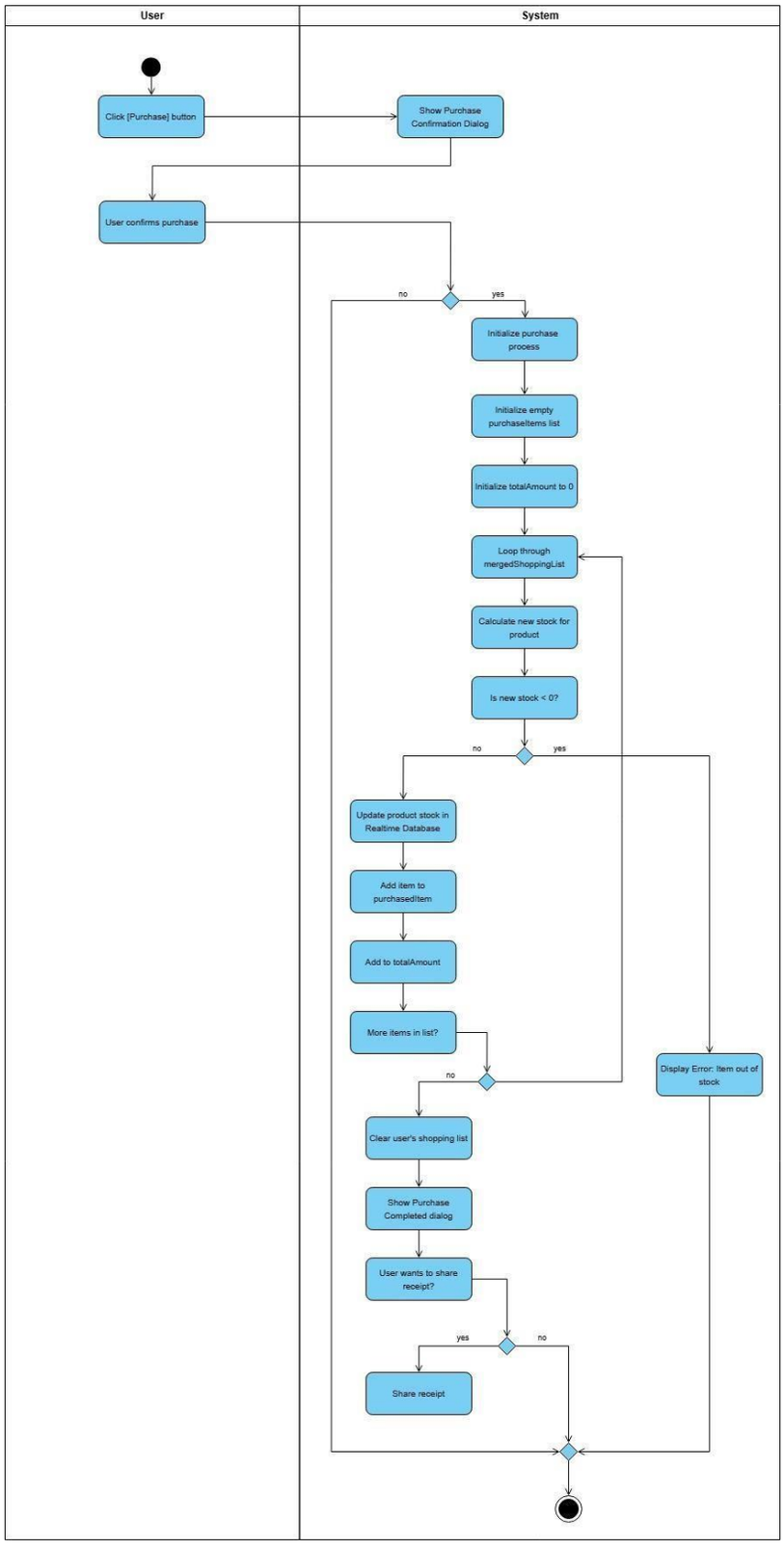


Figure 4.14.1 Purchase Activity Diagram

4.14.2 [F012] Purchase Module

The "Purchase" module begins when the user clicks the "Purchase" button. The system responds by displaying a Purchase Confirmation Dialog. If the user confirms the purchase, the system initiates the purchase process. It starts by initializing an empty purchaseItems list and setting the totalAmount to 0. The system then loops through the mergedShoppingList, calculating the new stock for each product. If the new stock would be less than 0, the system displays an error message indicating the item is out of stock. Otherwise, it updates the product stock in the Realtime Database, adds the item to the purchaseItems list, and adds its cost to the totalAmount. This process continues until all items in the list have been processed. Once complete, the system clears the user's shopping list and shows a Purchase Completed dialog. The user is then asked if they want to share the receipt. If the user chooses to share, the system facilitates sharing the receipt. Whether the user shares the receipt or not, the purchase process concludes, returning the user to the main interface.

4.15 [F013] Modify & Delete Shopping List

4.15.1 [F013] Modify & Delete Shopping List Activity Diagram

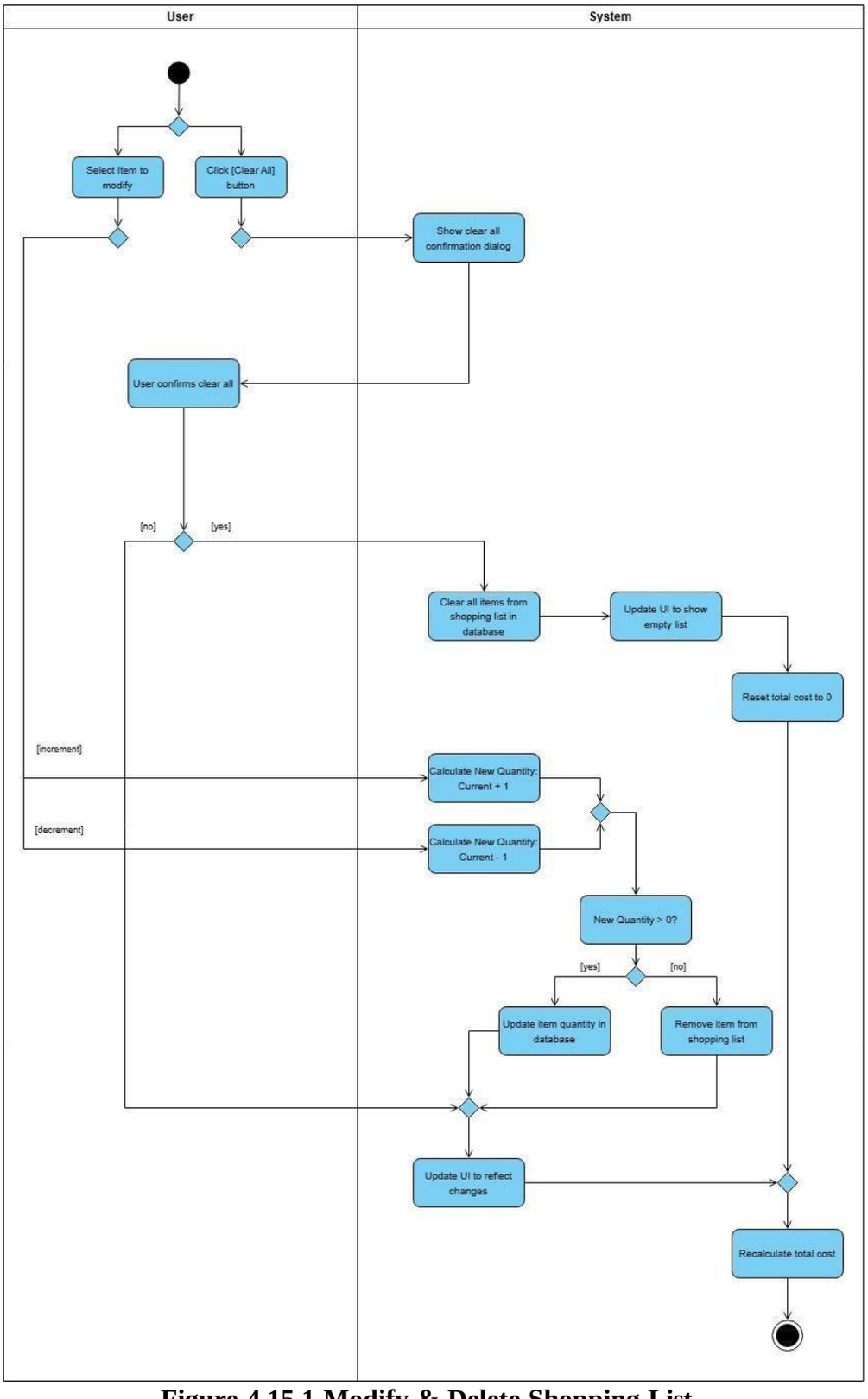


Figure 4.15.1 Modify & Delete Shopping List

4.15.2 [F013] Modify & Delete Shopping List Module

The "Modify & Delete Shopping List" module presents a streamlined process for managing a shopping list. Users have two main options: modifying individual items or clearing the entire list. When modifying an item, users can increase or decrease its quantity. The system calculates the new quantity and checks if it has reached zero. If so, the item is removed from the shopping list; otherwise, the quantity is updated in the database. The "Clear All" function begins with a confirmation dialog to prevent accidental deletions. If confirmed, the system removes all items from the database, updates the UI to display an empty list, and resets the total cost to zero. After any changes, whether modifying an item or clearing the list, the UI is updated to reflect the current state of the shopping list. Finally, the system recalculates the total cost based on the updated list.

4.16 [F014] Edit Profile Picture

4.16.1 [F014] Edit Profile Picture Activity Diagram

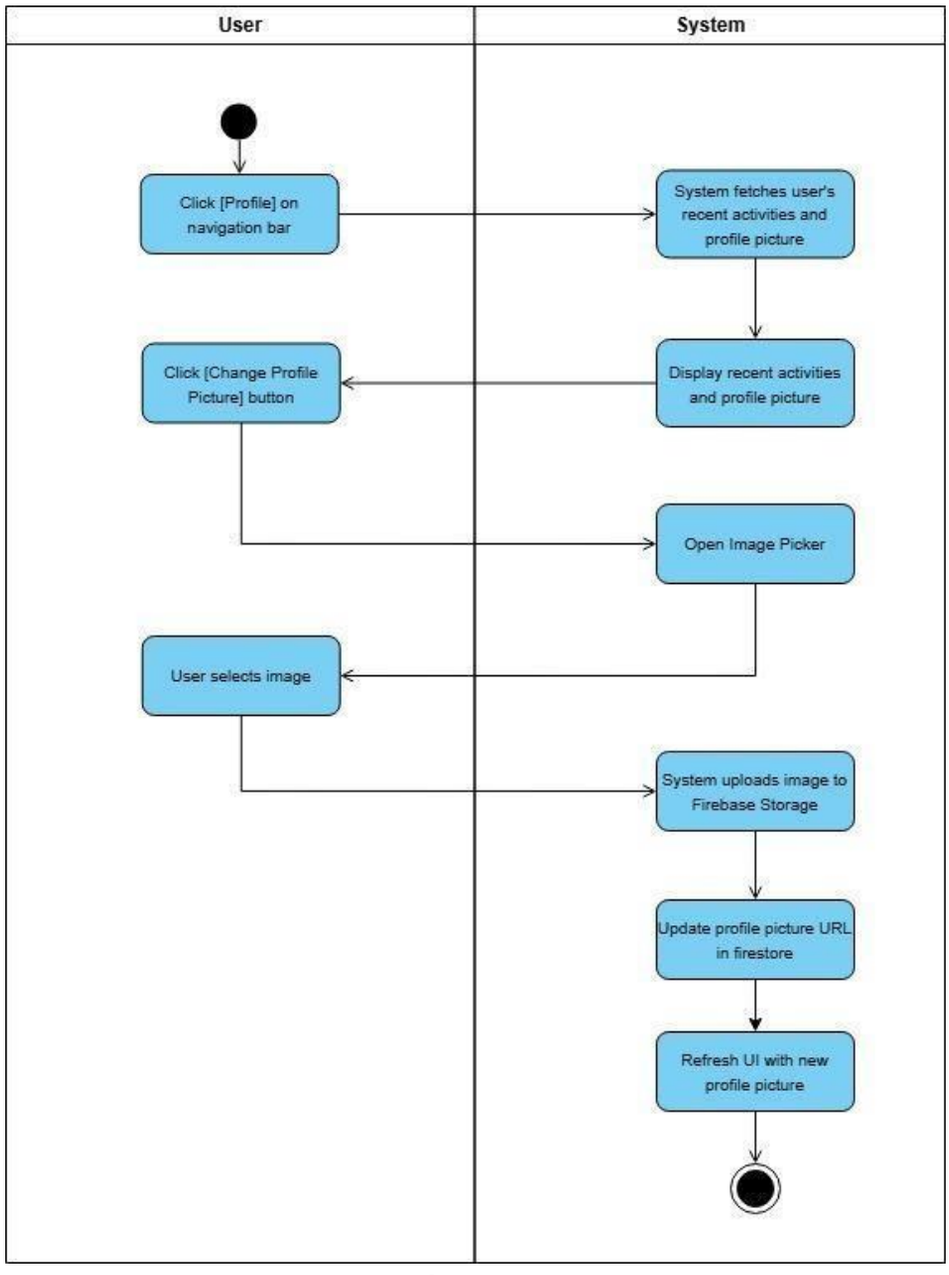


Figure 4.16.1 Edit Profile Picture Activity Diagram

4.16.2 [F014] Edit Profile Picture Module

The Edit Profile Picture module streamlines the process of updating user profile images. When users access their profile, the system fetches and displays their current picture and recent activities. To change their image, users click the "Change Profile Picture" button, opening an Image Picker. Once a new image is selected, the system automatically uploads it to Firebase Storage, updates the profile picture URL in

Firestore, and refreshes the UI to show the new image. This efficient flow leverages Firebase services for seamless backend integration, ensuring a smooth and responsive user experience while maintaining data consistency across the application. The module enhances user engagement by providing a quick and easy way for users to personalize their profiles.

4.17 Database Design

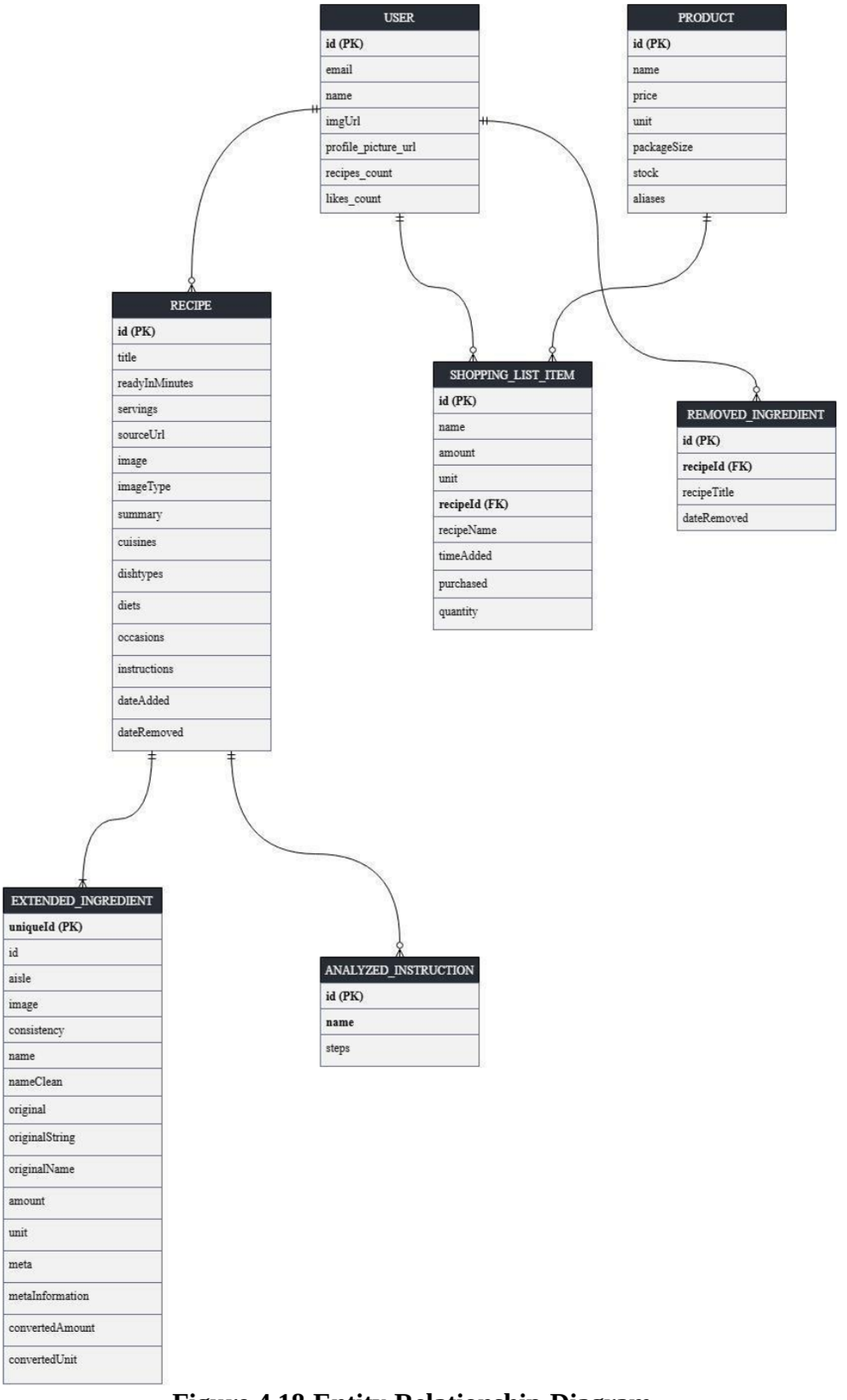


Figure 4.18 Entity Relationship Diagram

Entity	Description
User	This entity is used to store all the user personal details.
Recipe	This entity is used to store all the recipe details.
Extended Ingredient	This entity is used to store detailed information about ingredients in recipes.
Analyzed Instruction	This entity is used to store step-by-step instructions for recipes.
Shopping List Item	This entity is used to store the items in users' shopping lists.
Product	This entity is used to store all the product details available in the store inventory.
Removed Ingredient	This entity is used to store ingredients that have been removed from recipes.

Table 4.1 Entity Description

Field Name	Data Type	Null	PK/FK	Description
id	String	No	PK	User identification.
email	String	No		User's email address.
name	String	No		User's display name.
imgUrl	String	Yes		URL to user's profile image.
profile_picture_url	String	Yes		URL to user's profile picture.
recipes_count	Integer	No		Number of recipes saved by the user.
likes_count	Integer	No		Number of likes received by the user.

Table 4.2 Data Dictionary of User

Field Name	Data Type	Null	PK/FK	Description
id	Integer	No	PK	Recipe identification.
title	String	No		Recipe title.
readyInMinutes	Integer	Yes		Time needed to prepare the recipe (in minutes).
servings	Integer	Yes		Number of servings the recipe yields.
sourceUrl	String	Yes		Original source URL of the recipe.
image	String	Yes		URL to an image of the prepared recipe.
imageType	String	Yes		File type of the recipe image.
summary	String	Yes		Brief summary or description of the recipe.
instructions	String	Yes		Textual instructions for preparing the recipe.
dateAdded	Timestamp	No		Date and time when the recipe was added.
dateRemoved	Timestamp	Yes		Date and time when the recipe was removed (if applicable).

Table 4.3 Data Dictionary of Recipe

Field Name	Data Type	Null	PK/FK	Description
uniqueId	String	No	PK	Unique identifier for the ingredient instance.
id	Integer	Yes		Identifier for the ingredient type.
name	String	No		Name of the ingredient.
amount	Float	No		Quantity of the ingredient needed.
unit	String	No		Unit of measurement for the ingredient.

aisle	String	Yes		Supermarket aisle where the ingredient can be found.
image	String	Yes		URL to an image of the ingredient.
consistency	String	Yes		Texture or consistency of the ingredient.
originalString	String	Yes		Original string representation of the ingredient.
convertedAmount	Float	Yes		Converted quantity of the ingredient (if applicable).
convertedUnit	String	Yes		Converted unit of measurement (if applicable).

Table 4.4 Data Dictionary of Extended Ingredient

Field Name	Data Type	Null	PK/FK	Description
id	String	No	PK	Shopping list item identification.
name	String	No		Name of the item.
amount	Float	No		Quantity of the item needed.
unit	String	No		Unit of measurement for the item.
recipeId	String	Yes	FK	ID of the recipe this item is for (if applicable).
recipeName	String	Yes		Name of the recipe this item is for (if applicable).
added	Timestamp	No		Date and time when the item was added to the list.
purchased	Boolean	No		Indicates whether the item has been purchased.
quantity	Integer	No		Number of units of this item needed.

Table 4.5 Data Dictionary of Shopping List Item

4.18 Concluding Remark

This chapter defines the system's process, including its architecture, use cases, activity flows, functional modules, and database design. The architectural diagram provides a visual overview of all the system's components and how they interconnect. Use case diagrams illustrate how users with different roles interact with the system, highlighting the various functions available to each user type. Activity diagrams map out the system's control flow from start to finish for each function, showcasing the different decision paths a user can take. The functional module section details the input, processing, and output for each of the system's functions, offering a deeper understanding of how each component operates. The database design section outlines how the system's data is structured and organized. The entity relationship diagram (ERD) presents a graphical representation of the database, clearly depicting the relationships between different data entities. Data dictionaries complement the ERD by providing detailed descriptions of each attribute within the database.

CHAPTER 5: System Implementation

5.1 Software Setup and Configuration

5.1.1 Android Studio

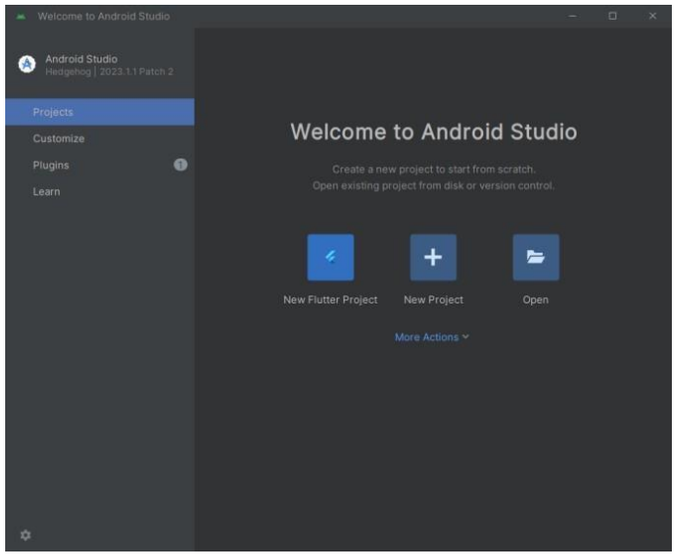


Figure 5.1.1.1– Android Studio

Android Studio can be downloaded from its official website (Android Studio download archives | Android Developers). The version used for this project is the Hedgehog version. After installation, the Android Studio window will appear as illustrated in Figure 5.1.1.1 shows the Android Studio. Android Studio is used to support the main development code editor, Visual Studio Code, by providing an Android emulator. Next, let's proceed with setting up the emulator in Android Studio.

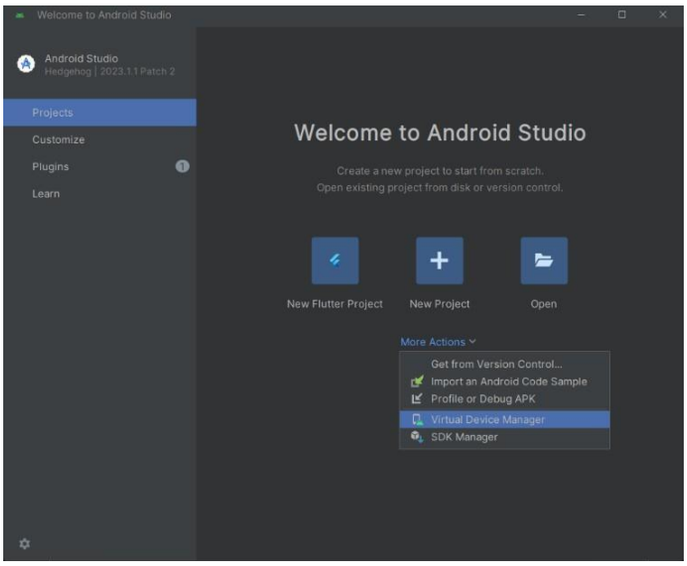


Figure 5.1.1.2 – Android Emulator Setup (i)

To setup the Android emulator in Android Studio, click on the "More Actions" that displayed in the window in Figure 5.1.1.2. A list of selection will be shown, click on the "Virtual Device Manager".

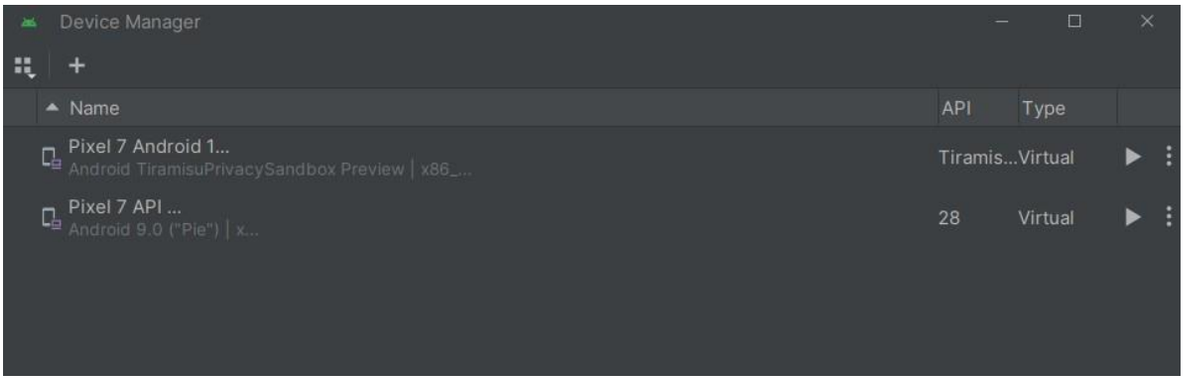


Figure 5.1.1.3 – Android Emulator Setup (ii)

After clicking the "Virtual Device Manager" button, the device manager window will be displayed. Now, click on the "+" button, the second selection in the top panel window. (Figure 5.1.1.3 shows the Android Emulator Setup ii)

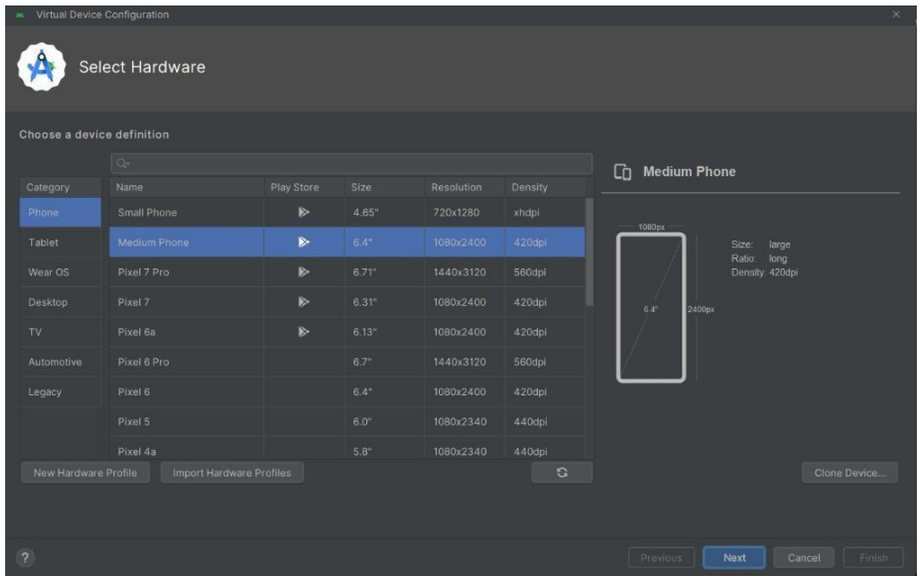


Figure 5.1.1.4 – Android Emulator Setup (iii)

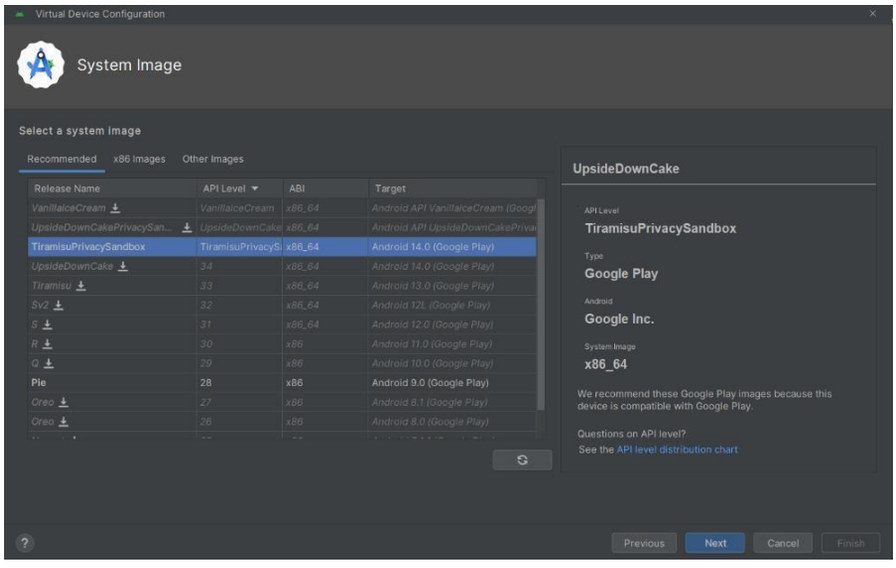


Figure 5.1.1.5 – Android Emulator Setup (iv)

The virtual device configuration window will be displayed, as shown in Figure 5.1.1.4, illustrating the Android Emulator Setup (iii). For this project, the device configuration is set to Pixel 7, with the API level set to TiramisuPrivacySandbox, as shown in Figure 5.1.1.5 (Android Emulator Setup iv). The rest of the configurations are left as default. Once the setup is complete, the Android Emulator can be launched in Android Studio via the AVD Manager.

5.1.2 Visual Studio Code

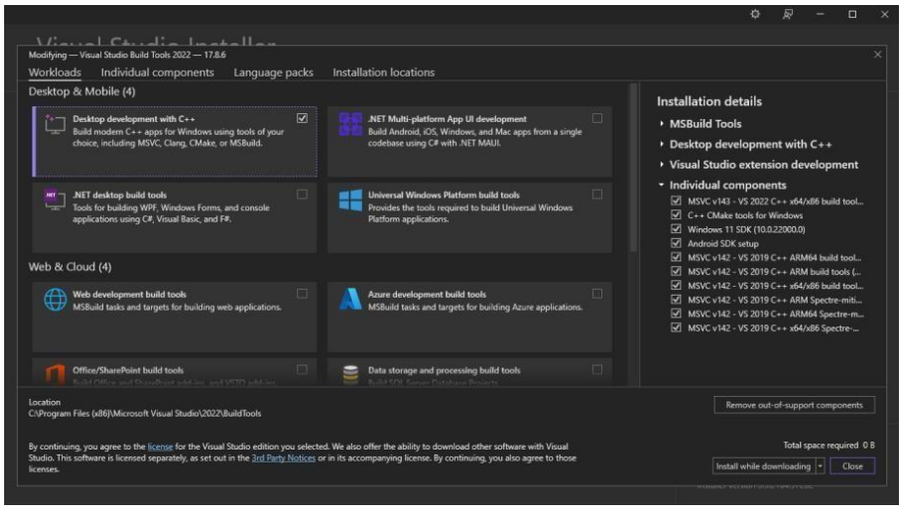


Figure 5.1.2.1 – Visual Studio Setup (i)

Visual Studio Code is the core code editor that being used to develop this project. To download Visual Studio Code, simply navigate to their official website (Download Visual Studio Code - Mac, Linux, Windows). After finished the installation, the Visual Studio Code Build tools window will appear. (Figure 5.1.2.1 shows the Visual Studio Setup i) The components that has been installed include “Desktop development with C++”, “Visual Studio Extension Development”, Android SDK setup, Windows 11 SDK, and MSVC v142 2019 components. (Figure 5.1.2.1 shows the Visual Studio Setup i)

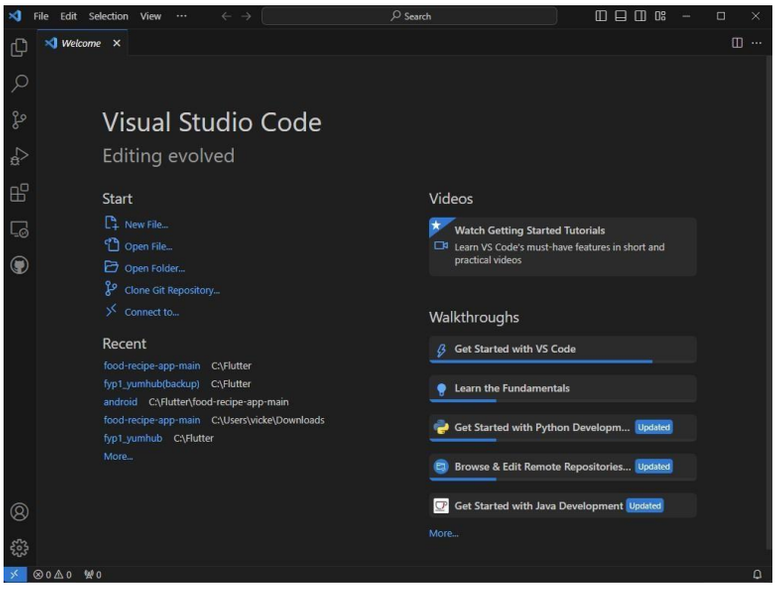


Figure 5.1.2.2 – Flutter Setup

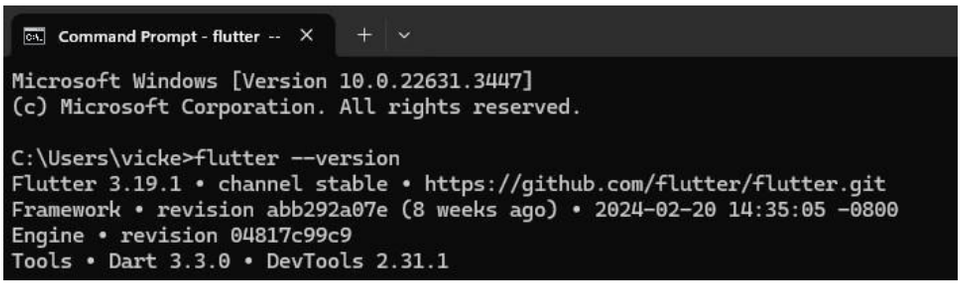


Figure 5.1.2.3 – Flutter version

To setup Flutter on Visual Studio Code, Flutter and Dart SDK must be installed on device system. Both installers can be downloaded from their official website. The flutter version being implement in this project is 3.19.1, and 3.3.0 for the version of Dart. (Figure 5.1.2.3 shows the Flutter version)

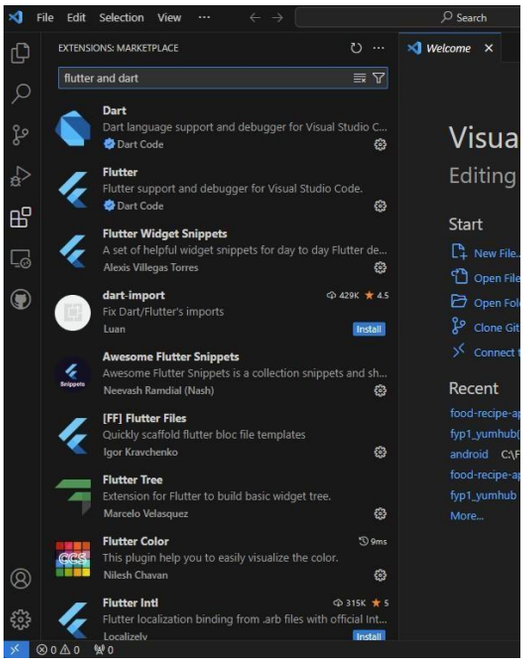


Figure 5.1.2.4 – Install Flutter and Dart Extension

Navigating to the Visual Studio Code window as shown in Figure 5.1.2.2 (Flutter Setup), click on the fifth left side-bar icon, it's the extension marketplace icon. Then, type in "flutter and dart" in the search box to get relevant extensions. The core extensions to install flutter on Visual Studio Code is the "Dart" and "Flutter" extension as shown in Figure 5.1.2.4 (Install Flutter and Dart Extension). After finished the installation, flutter project can now be created in Visual Studio Code.

5.1.3 Firebase

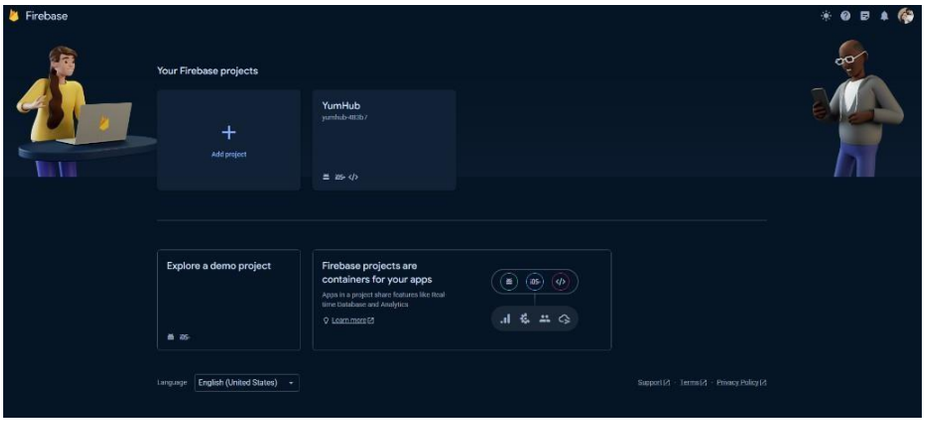


Figure 5.1.3.1 – Google Firebase

To setup Google Firebase, simply navigate to their official firebase console website (Firebase console (google.com)). Log in to access Firebase features. After successfully logging in, click on the "Add project" button, as shown in Figure 5.1.3.1 (Google Firebase).

Figure 5.1.3.4 – Project Setup (i)

The page will redirect to the project creation screen. Enter the project name and press "Continue" (Figure 5.1.3.4 shows the Project Setup i).

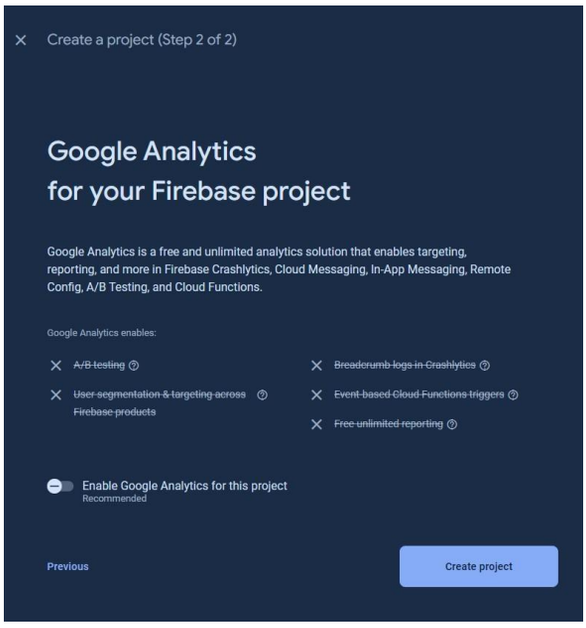


Figure 5.1.3.5 – Project Setup (ii)

Next, disable Google Analytics and create the project (Figure 5.1.3.5 shows Project Setup ii). After the project is successfully created, the page shown in Figure 5.1.3.6 (Project Setup iii) will be displayed.

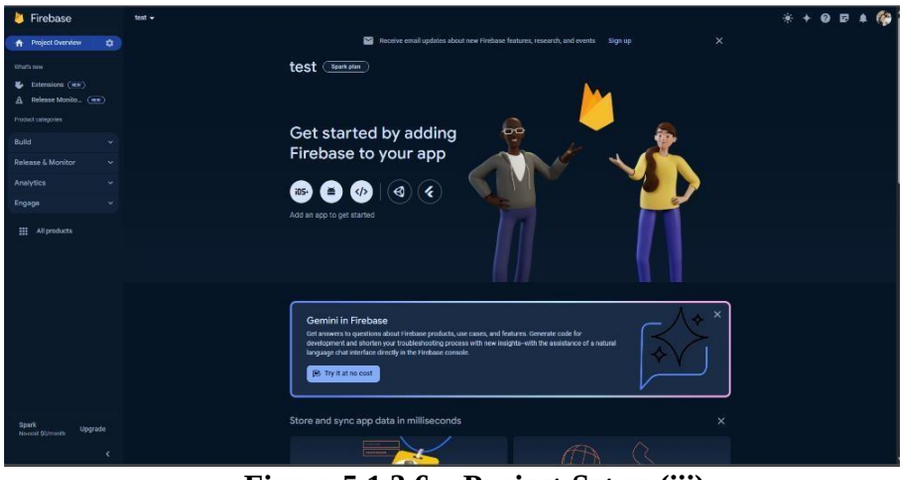


Figure 5.1.3.6 – Project Setup (iii)

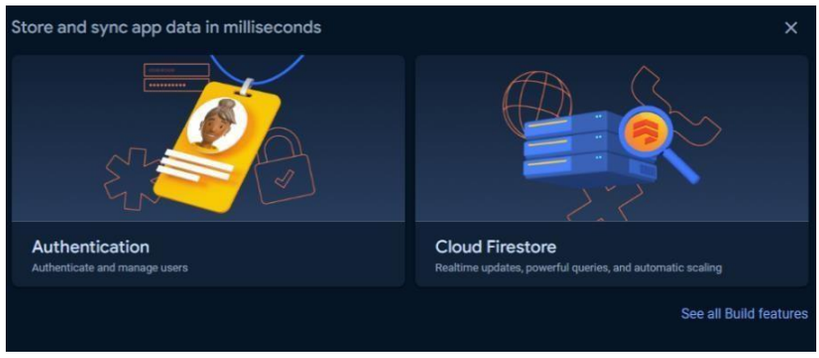


Figure 5.1.3.7 – Setup Authentication

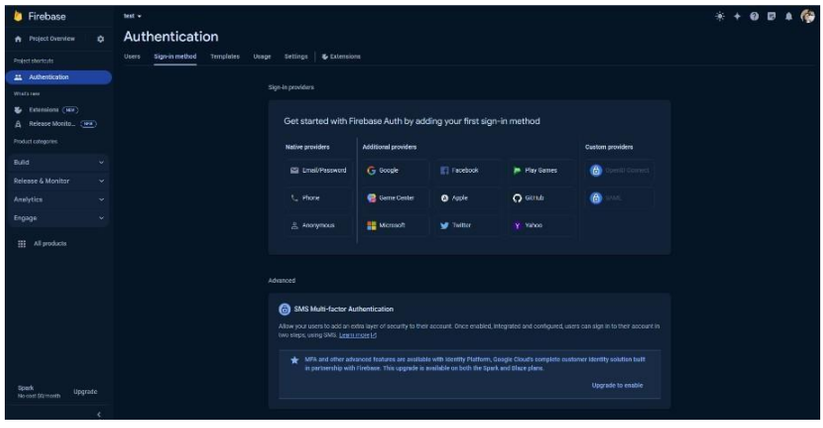


Figure 5.1.3.8 – Authentication page

Scroll down and hover over the “Authentication” option and click it, the website will redirect user to the authentication page. (Figure 5.1.3.8 shows the Authentication page) The authentication that has been implemented in this project are “Email Password” and “Google”.

5.2 System Operation

5.2.1 Welcome, Login, Sign-up

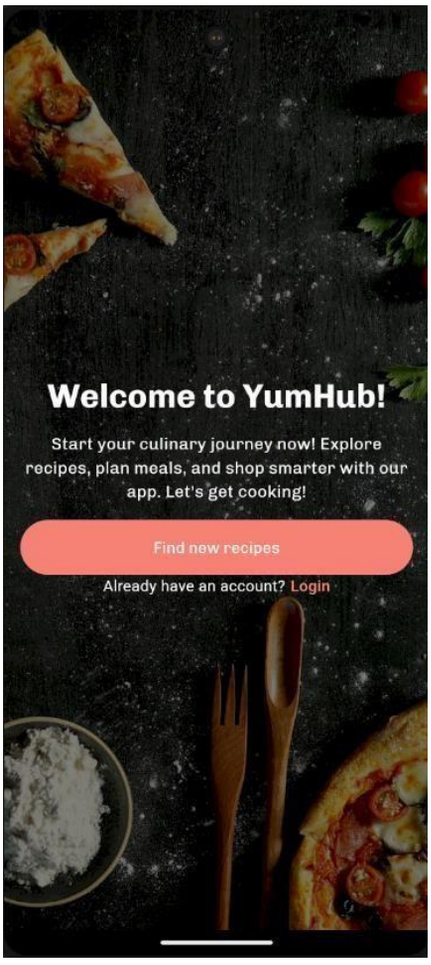


Figure 5.2.1.1 – Welcome page

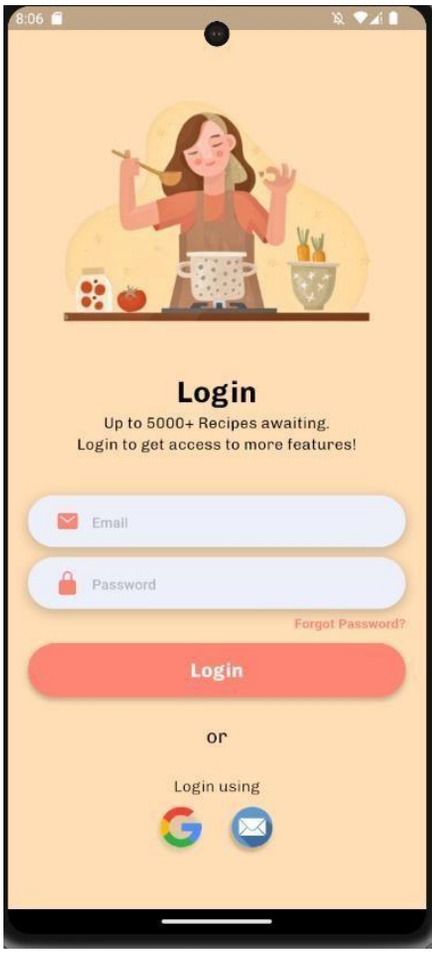


Figure 5.2.1.2 – Login page

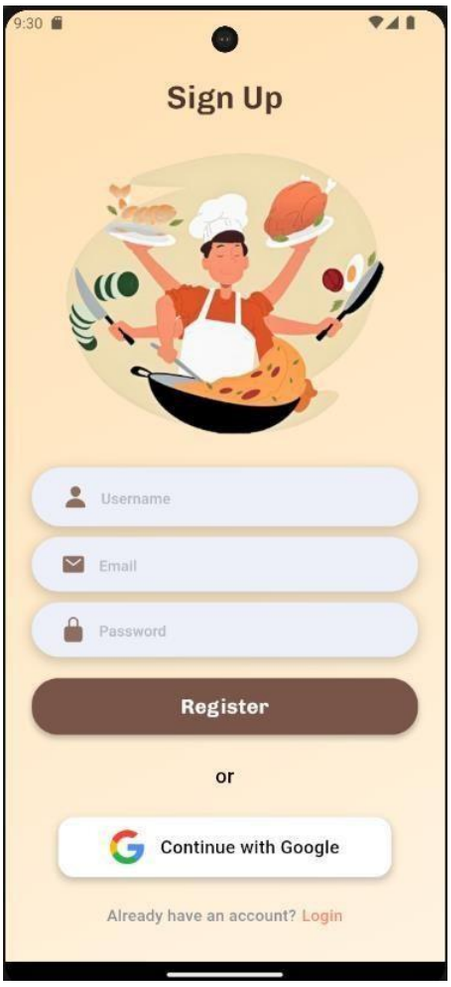


Figure 5.2.1.3 – Sign-up page

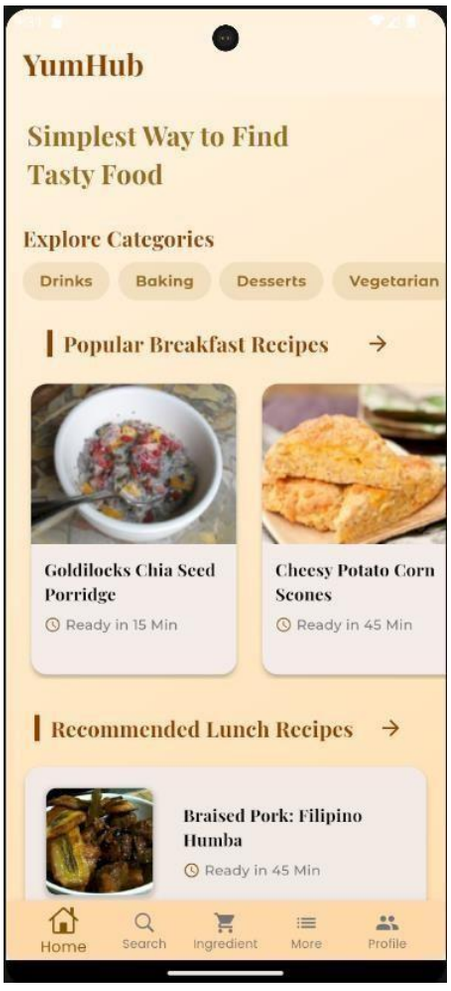


Figure 5.2.1.4 – Home page

The welcome page will be displayed to the first-time user when they run the application (Figure 5.2.1.1 shows the Welcome page). The user can choose to get access to the application home page without creating any account by clicking the “Find new recipes” button. However, the user is required to sign-in to get access to full features of the application including save recipe as favorite, meal planning, grocery management and profile page. Moving to the login page (Figure 5.2.1.2 shows the Login page), there are two authentication options for the user to login the application. Authentication options include email & password authentication and Google sign-in. Unverified user will be navigated to the login page when they run the application. User are required to enter registered email address and password in-order to get full access to the features of the application. For new users, they are required to register a new account by filling in username, email address, and password. (Figure 5.2.1.3 shows the Sign-up page)

Other than that, user can sign-up with Google account as well. The home page will be displayed to both verified and un-verified users as illustrated in Figure 5.2.1.4 (Home page). At the bottom of the Home page, there's a navigation bar that shows five different categories of selections. Which includes homepage, search recipe page, recipe saved as favorite page, more page, and profile page. The content of the home page includes various type of floating widgets that were implemented with functions and features.

```
Future<UserCredential> createUserWithEmailAndPassword({
  required String email,
  required String password,
}) async {
  return UserCredential._(
    this,
    await _delegate.createUserWithEmailAndPassword(email, password),
  );
}
```

Figure 5.2.1.5 – Firebase API (Create User)

As shown in Figure 5.2.1.5, which illustrates the Firebase API (Create User), there is a function called `createUserWithEmailAndPassword()`. This function is part of the `firebase_auth` Dart package that has been integrated into the application.

```
void registration() async {
  if (passwordController.text != "" &&
      nameController.text != "" &&
      emailController.text != "") {
    try {
      UserCredential userCredential = await FirebaseAuth.instance
          .createUserWithEmailAndPassword(
              email: email.trim(), password: password.trim());
      uid = userCredential.user!.uid;
      await FirebaseAuth.instance.currentUser!.updateDisplayName(name.trim());
      await FirestoreServices.saveUser(name.trim(), email.trim(), uid);
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
          content: Text(
            "Account created successfully.",
            style: TextStyle(fontSize: 15.0),
          ), // Text
        ), // SnackBar
      );
```

Figure 5.2.1.6 – Registration Function

Figure 5.2.1.7 – User Account

When the user pressed the "Register" button in the email and password sign-up page (Figure 5.2.1.3 shows the Sign-up page), the registration() function will be called. (Figure 5.2.1.6 shows the Registration Function) Inside the registration() function, the system first will validate the text-input box in Figure 5.2.1.3 Sign-up page, system proceed to waiting for FirebaseAuth to initialize, then create user with email and password and stores in Firebase Authentication. After the account is successfully created, a unique userID will be generated to authenticate the user, userID can be found in Figure 5.2.1.7 shows the User Account last column.

```dart
import 'package:cloud_firestore/cloud_firestore.dart';

class FirestoreServices {
  static saveUser(String name, email, uid) async {
    await FirebaseFirestore.instance
        .collection('email-auth-users')
        .doc(uid)
        .set({'email': email, 'username': name});
  }

  static deleteUser(String name, email, uid) async {
    await FirebaseFirestore.instance
        .collection('email-auth-users')
        .doc(uid)
        .delete();
  }
}
```

Figure 5.2.1.8 – Store & Delete user details in Firestore Database

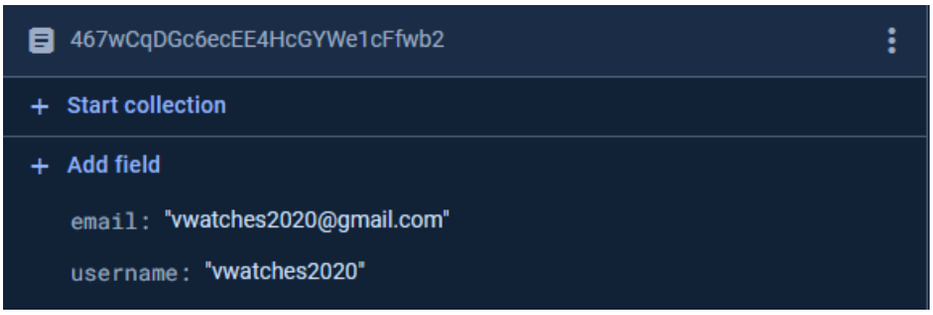


Figure 5.2.1.9 – User details in Firestore Database (Email & password)

At the same time, in Figure 5.2.1.8 shows the Store & Delete user details in Firestore Database, the Firestore “saveUser” function were called to store user details such as email address and username into Firestore database, information has been stored inside Firestore database as illustrated in Figure 5.2.1.9 shows the User details in Firestore Database (Email & password).

```dart
class AuthMethods {
  final FirebaseAuth auth = FirebaseAuth.instance;

  getCurrentUser() async {
    return auth.currentUser;
  }

  signInWithGoogle(BuildContext context) async {
    final FirebaseAuth firebaseAuth = FirebaseAuth.instance;
    final GoogleSignIn googleSignIn = GoogleSignIn();

    final GoogleSignInAccount? googleSignInAccount =
        await googleSignIn.signIn();

    final GoogleSignInAuthentication? googleSignInAuthentication =
        await googleSignInAccount?.authentication;

    final AuthCredential credential = GoogleAuthProvider.credential(
        idToken: googleSignInAuthentication?.idToken,
        accessToken: googleSignInAuthentication?.accessToken);

    UserCredential result = await firebaseAuth.signInWithCredential(credential);

    User? userDetails = result.user;

    if (result != null) {
      Map<String, dynamic> userInfoMap = {
        "email": userDetails!.email,
        "name": userDetails.displayName,
        "imgUrl": userDetails.photoURL,
        "id": userDetails.uid
      };
      await DatabaseMethods()
          .addUser(userDetails.uid, userInfoMap)
          .then((value) {
        Navigator.push(
            context, MaterialPageRoute(builder: (context) => BottomNavView()));
      });
    }
  }
}
```

Figure 5.2.1.10 – Google Sign-In Function

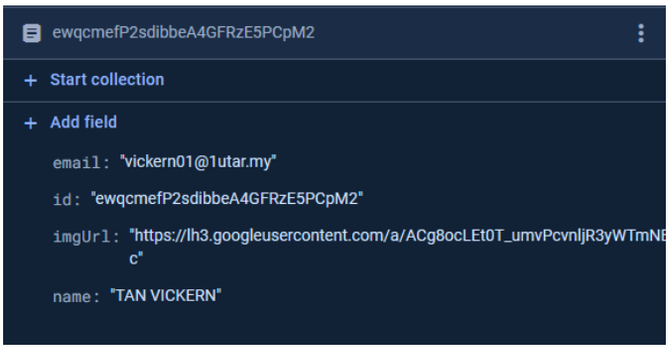


Figure 5.2.1.11 – User details in Firestore Database (Google)

For users who create their account with Google, the system calls the Google Sign-In function included in the imported `google_sign_in` Dart package (Figure 5.2.1.10 shows the Google Sign-In Function). This function creates an account in Firebase Authentication and simultaneously stores user details, such as email address, user ID, image URL, and name, in the Firestore database, as shown in Figure 5.2.1.11 (User details in Firestore Database - Google). These details are directly extracted from the associated Google Gmail account used during registration.

Figure 5.2.1.12 – Invalid Email (i)

Before the system creates an account for the user, input validations are performed. For example, if a user enters an email that is already registered, an error message stating "This email has already been registered." will appear at the bottom of the screen. This indicates that the user is attempting to register a new account with an email that is already stored in the database (Figure 5.2.1.12 shows the Invalid Email i).

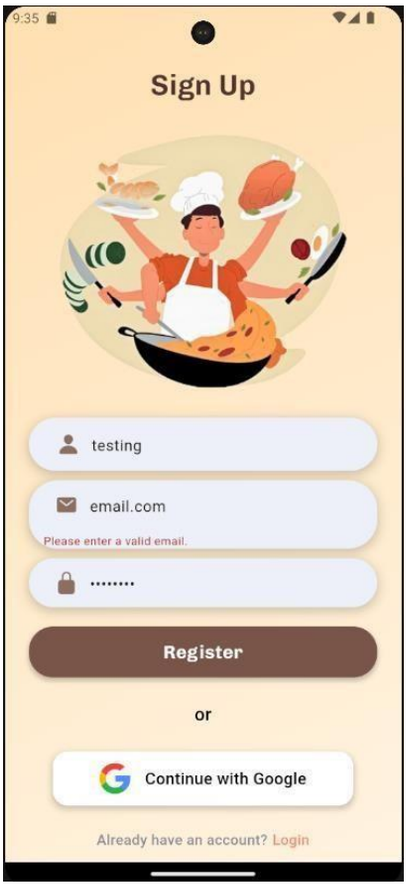


Figure 5.2.1.13 – Invalid Email (ii)

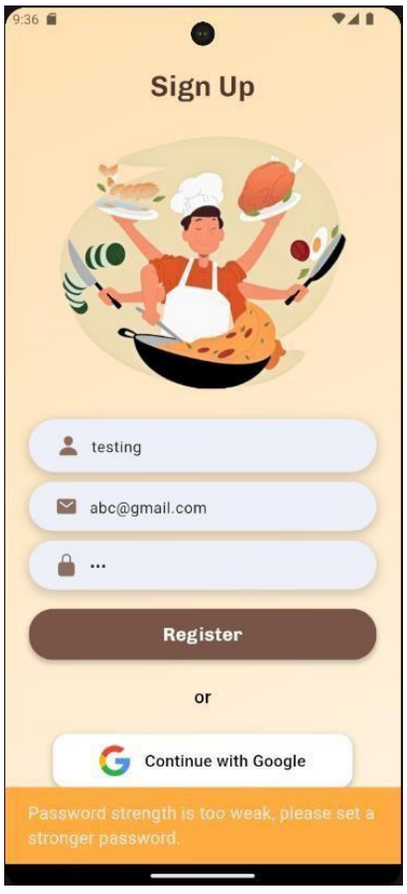


Figure 5.2.1.14 – Weak Password Strength

If the user provides an incorrectly formatted email address, an error message saying "Please enter a valid email" in red font will be displayed at the bottom of the input textbox (Figure 5.2.1.13 shows Invalid Email ii). If the user enters a password that is less than 6 characters, the system will display a message stating, "Password strength is too weak, please set a stronger password" (Figure 5.2.1.14 shows Weak Password Strength).

```
} on FirebaseAuthException catch (e) {
  if (e.code == 'weak-password') {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(
        backgroundColor: Colors.orangeAccent,
        content: Text(
          "Password strength is too weak, please set a stronger password.",
          style: TextStyle(fontSize: 18.0),
        ), // Text
      ), // SnackBar
    );
  } else if (e.code == 'email-already-in-use') {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(
        backgroundColor: Colors.orangeAccent,
        content: Text(
          "The email has already registered.",
          style: TextStyle(fontSize: 18.0),
        ), // Text
      ), // SnackBar
    );
```

Figure 5.2.1.15 – Email Validation Code

In Figure 5.2.1.15 Email Validation Code is the source code of validating both condition in Figure 5.2.1.12 (Invalid Email i), and Figure 5.2.1.14 (Weak Password Strength).

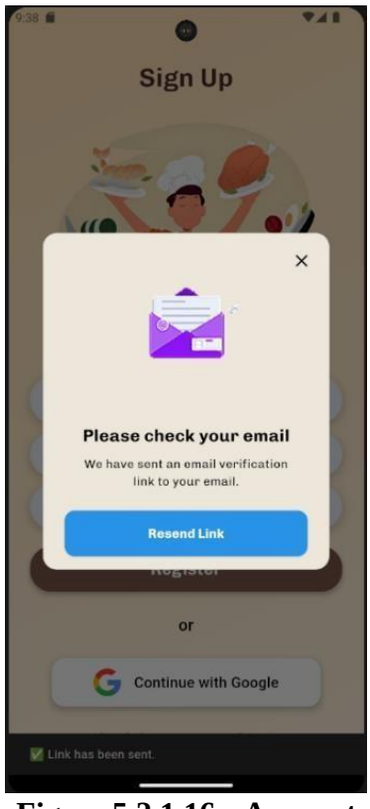


Figure 5.2.1.16 – Account Registered Successfully Prompt

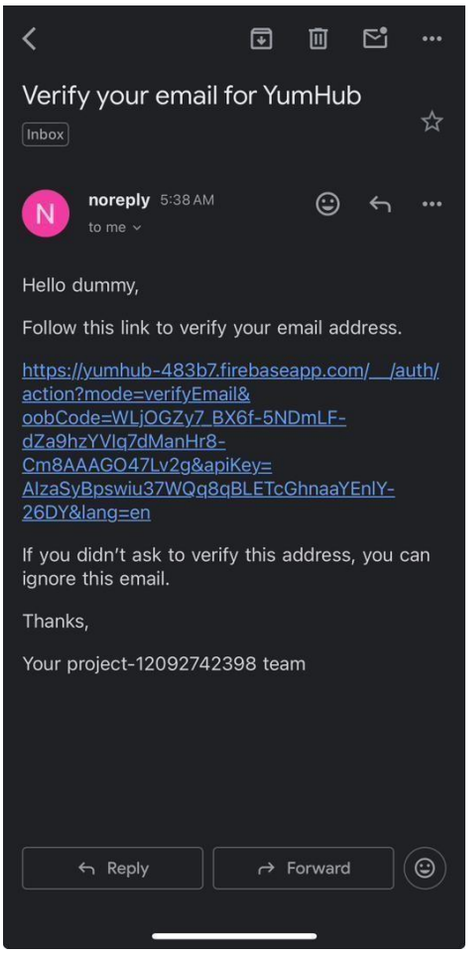


Figure 5.2.1.17 – Email Verification Link

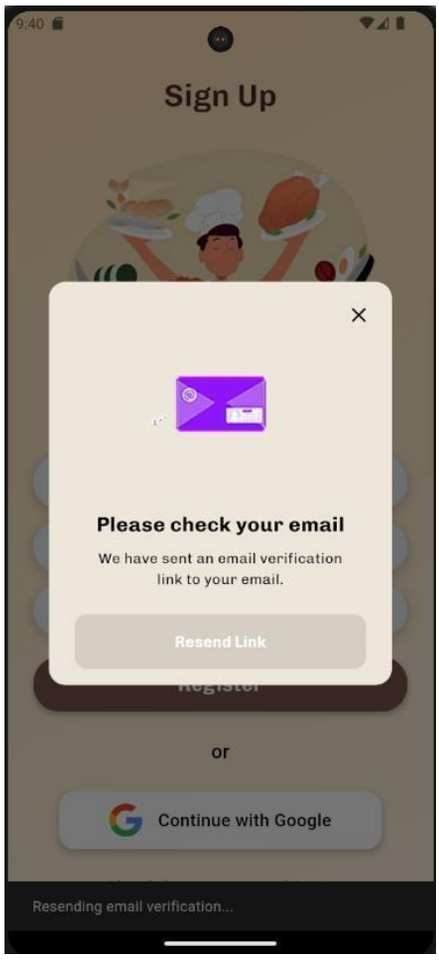


Figure 5.2.1.18 – Resend Verification Code

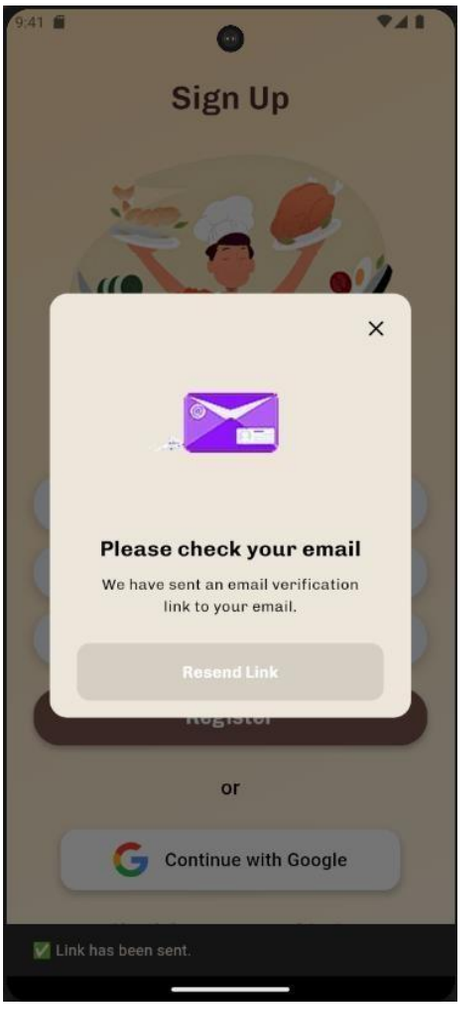


Figure 5.2.1.19 – Resend Link (Success)

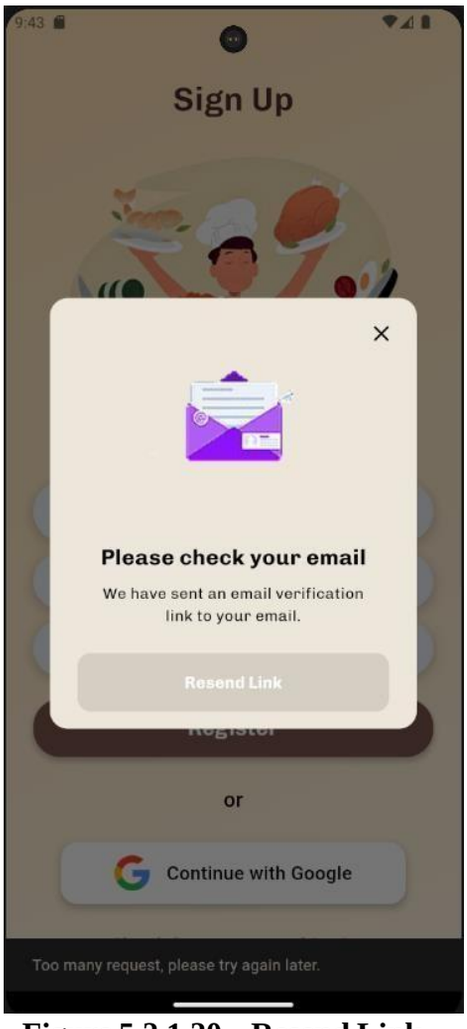


Figure 5.2.1.20 – Resend Link (Failed)

After the user enters a valid email address, the system will display a dialog box with the message "Please check your email" and "We have sent an email verification link to your email," as illustrated in Figure 5.2.1.16 (Account Registered Successfully Prompt). The user will receive an email verification link, as shown in Figure 5.2.1.17 (Email Verification Link). The user can choose to resend the verification link by clicking the “Resend Link” button, also shown in Figure 5.2.1.16 (Account Registered Successfully Prompt). After clicking this button, the system will display a message saying "Link has been sent" (Figure 5.2.1.19 shows Resend Link Success). However, if the user has requested the verification link too many times, the system will display the message "Too many requests, please try again later" (Figure 5.2.1.20 shows Resend Link – Failed).

Figure 5.2.1.21 – Invalid Credentials

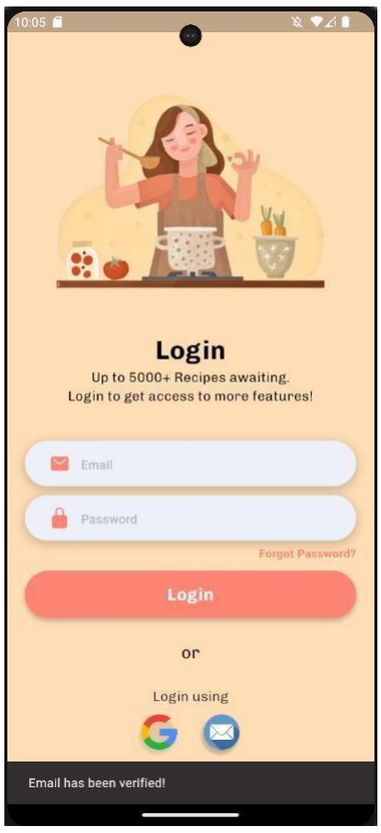


Figure 5.2.1.22 – Successfully Verified

If the user attempts to log in without verifying their registered email address, the system will display the message "Invalid credentials" (Figure 5.2.1.21 shows Invalid Credentials). This occurs because the system checks the verification status of the email address. If the email is verified, the user will receive the message "Email has been verified" and will be redirected to the login page (Figure 5.2.1.22 shows Successfully Verified).

The "Invalid credentials" message can appear in several situations, such as if the entered email address is not valid or not registered, if the password is incorrect, or if the email is registered but not verified. This implementation is designed to prevent Email Enumeration Attacks, where attackers might use leaked email addresses to determine if an email is valid or registered in the system. If the error message were "Incorrect password" instead of "Invalid email," it would reveal that the email is already registered, allowing attackers to focus on password brute-forcing to gain access to the user's account.

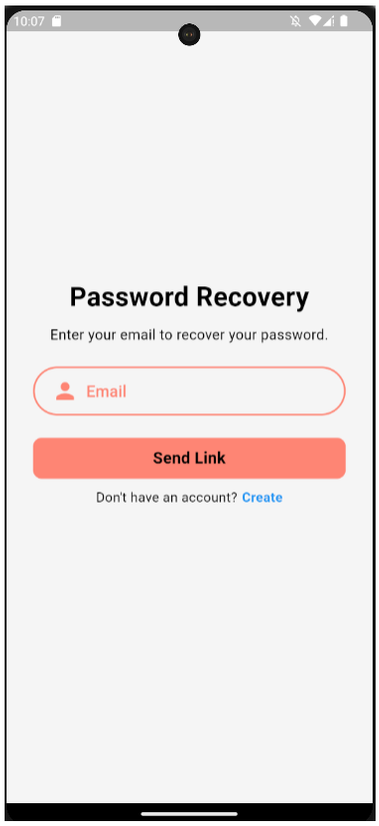


Figure 5.2.1.23 – Password Recovery page

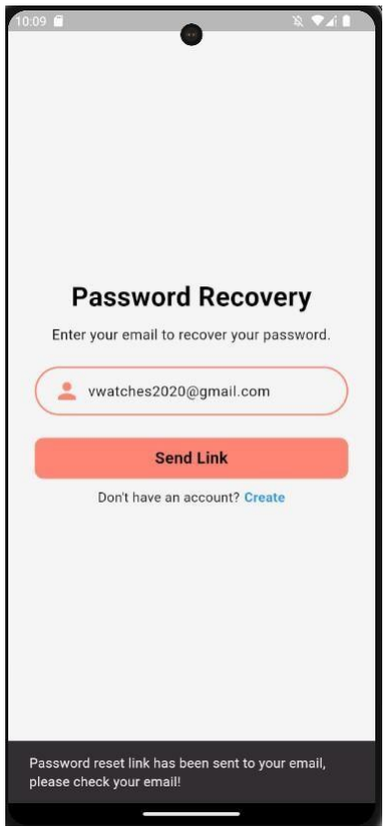


Figure 5.2.1.24 – Password Reset Link Sent

If the user forgets their account password, they can choose to reset it by clicking the "Forgot password" link on the login page. They will then be redirected to the password recovery page, as illustrated in Figure 5.2.1.23 (Password Recovery Page). After entering their email address and pressing the "Send Link" button, the user will receive a prompt with the message "Password reset link has been sent to your email, please check your email!" (Figure 5.2.1.24 shows Password Reset Link Sent). The user can then click the password reset link in the email to reset their password (Figure 5.2.1.25 shows Password Reset Link, Figure 5.2.1.26 shows Enter New Password, and Figure 5.2.1.27 shows Password Changed Message).

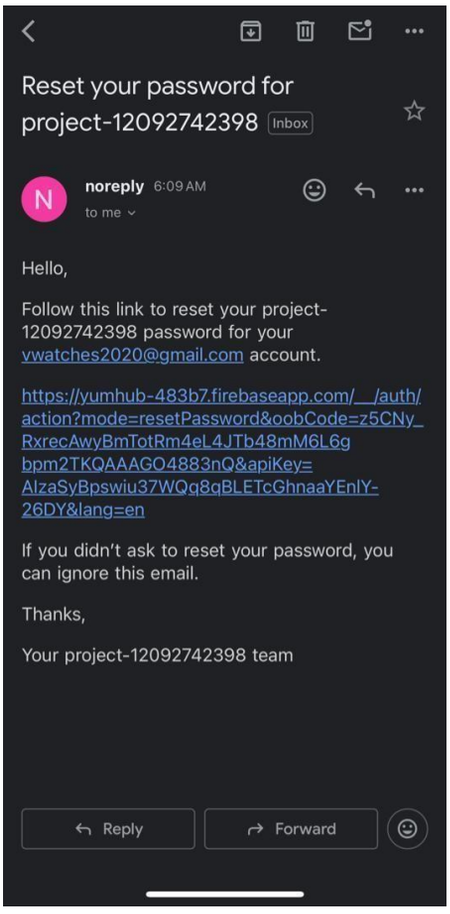


Figure 5.2.1.25 – Password Reset Link

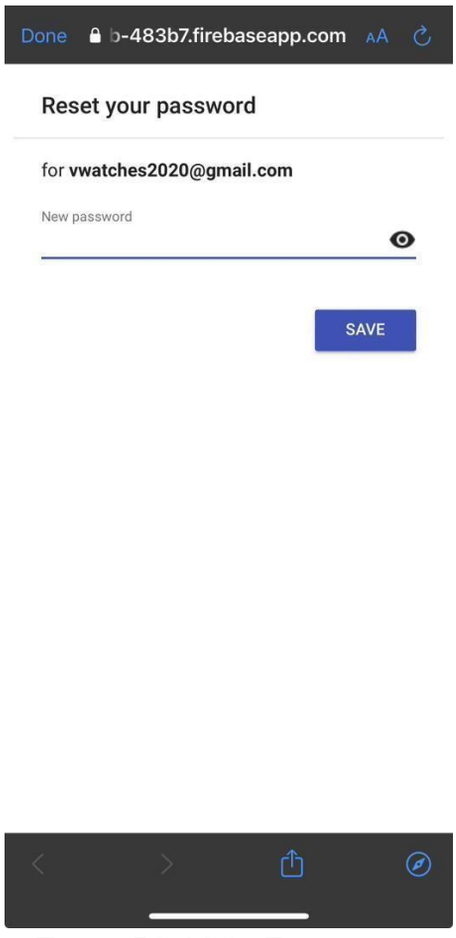


Figure 5.2.1.26 – Enter new password

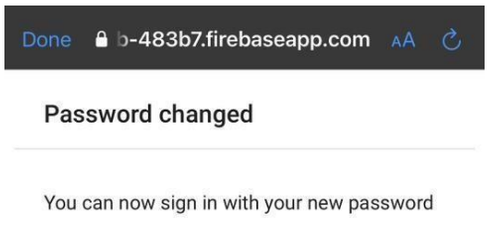


Figure 5.2.1.27 – Password changed message

5.2.2 Recipe Information

Figure 5.2.2.1 – Recipe Information

Figure 5.2.2.2 – Ingredient's details

When the user clicks on any recipe on the home page, they will be navigated to the recipe page. This page displays useful information, including the Spoonacular recipe rating, recipe title, preparation time (Ready in), serving size, and price per serving (Figure 5.2.2.1 shows Recipe Information).

In the ingredients list section, each ingredient is clickable. When the user toggles an ingredient, a container with detailed information about the ingredient will be displayed (Figure 5.2.2.2 shows Ingredient's Details). This information includes the type of ingredient, its consistency, and the amount required.

Figure 5.2.2.3 – Recipe Information (Share Recipe – Confirmation)

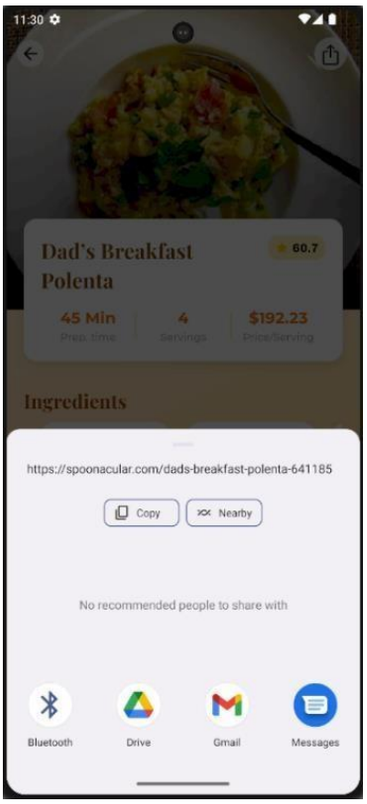


Figure 5.2.2.4 – Recipe Information (Share – Media)

On the recipe information page, a "Share" button is located in the top right corner (Figure 5.2.2.1 shows Recipe Information). When the user taps this button, a confirmation dialog will appear (Figure 5.2.2.3 shows Recipe Information - Share Recipe Confirmation). If the user confirms the action, they can share the recipe link through any online messaging application, or choose to copy the link and send it manually (Figure 5.2.2.4 shows Recipe Information - Share - Media).

Figure 5.2.2.5 – Recipe Instructions

Figure 5.2.2.6 – Utensils

As the user scrolls down the recipe page, the recipe instructions and related equipment will be displayed. The recipe instructions will include all the steps required to prepare the dish (Figure 5.2.2.5 shows Recipe Instructions), while the utensils section will list all the kitchen accessories needed for preparing the meal (Figure 5.2.2.6 shows Utensils).

Figure 5.2.2.7 – Nutrition Information & Similar Recipes

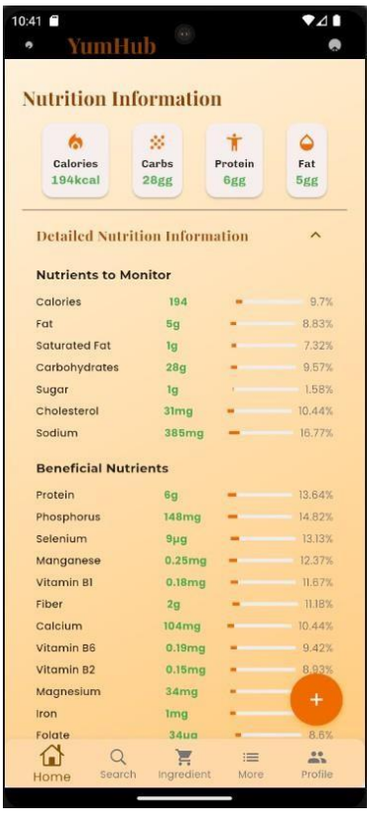


Figure 5.2.2.8 – Detailed Nutrition Information

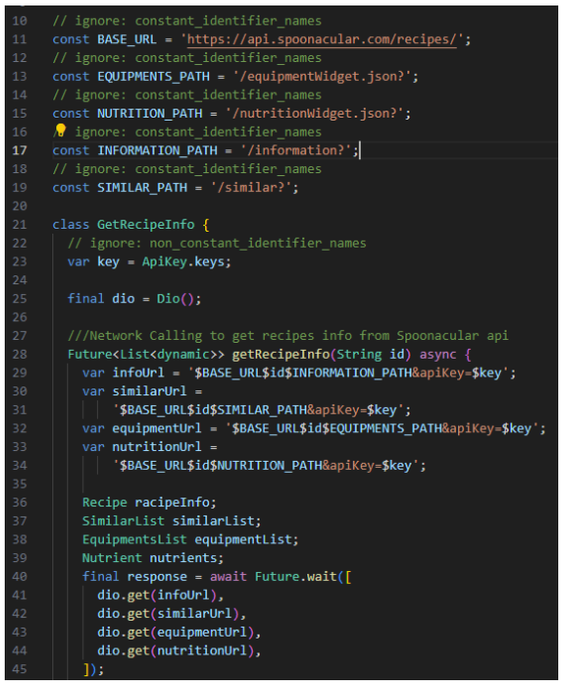
As the user continues scrolling to the end of the page, they will encounter the "Nutrition Information" section, which includes nutrition facts for the specific recipe, and the "Similar Recipes" section, which displays recipes similar to the one shown on the information page. The user can tap on any recipe widget in the "Similar Recipes" section to view it. The nutrition facts include details such as calories, carbohydrates, protein, and fat (Figure 5.2.2.7 shows Nutrition Information & Similar Recipes).

Below this section is the "Detailed Nutrition Information," which provides two categories of summarized useful data: "Nutrients to Monitor" and "Beneficial Nutrients." This data is displayed with various metrics and can be accessed by toggling the "Detailed Nutrition Information" tab (Figure 2.2.8 shows Detailed Nutrition Information).

Figure 5.2.2.9 – Save Recipe & Ingredient button (i)

Figure 5.2.2.10 – Save Recipe & Ingredient button (ii)

Additionally, a “Save Recipe & Ingredient” button is pinned to the bottom right corner of the recipe information page (Figure 5.2.2.9 shows Save Recipe & Ingredient Button i). This button allows the user to save all the recipe's ingredients to the grocery list for easy reference and access. When the user clicks the “+” sign, the button will expand to show additional options. Clicking the “x” sign will revert the button to its original form (Figure 5.2.2.10 shows Save Recipe & Ingredient Button ii).



```dart
// ignore: constant_identifier_names
const BASE_URL = 'https://api.spoonacular.com/recipes/';
// ignore: constant_identifier_names
const EQUIPMENTS_PATH = '/equipmentWidget.json?';
// ignore: constant_identifier_names
const NUTRITION_PATH = '/nutritionWidget.json?';
// ignore: constant_identifier_names
const INFORMATION_PATH = '/information?';
// ignore: constant_identifier_names
const SIMILAR_PATH = '/similar?';

class GetRecipeInfo {
  // ignore: non_constant_identifier_names
  var key = ApiKey.keys;

  final dio = Dio();

  ///Network Calling to get recipes info from Spoonacular api
  Future<List<dynamic>> getRecipeInfo(String id) async {
    var infoUrl = '$BASE_URL$id$INFORMATION_PATH&apiKey=$key';
    var similarUrl =
        '$BASE_URL$id$SIMILAR_PATH&apiKey=$key';
    var equipmentUrl = '$BASE_URL$id$EQUIPMENTS_PATH&apiKey=$key';
    var nutritionUrl =
        '$BASE_URL$id$NUTRITION_PATH&apiKey=$key';

    Recipe racipeInfo;
    SimilarList similarList;
    EquipmentsList equipmentList;
    Nutrient nutrients;
    final response = await Future.wait([
      dio.get(infoUrl),
      dio.get(similarUrl),
      dio.get(equipmentUrl),
      dio.get(nutritionUrl),
    ]);
```

Figure 5.2.2.11 – Spoonacular API Implementation code

Each page displaying information related to the food recipe, integrated from the Spoonacular API, requires code implementation as illustrated in Figure 5.2.2.11 (Spoonacular API Implementation Code). The Dio library, an external Dart package imported for making HTTP requests, is used to retrieve data from the API. The information obtained from the API includes recipe details, similar recipes, cooking equipment, and nutritional information (Figure 5.2.2.11 shows Spoonacular API Implementation Code). The Dio library is utilized to fetch this data from the Spoonacular API using the specified URLs.

5.2.3 Search Page

Figure 5.2.3.1 – Search page

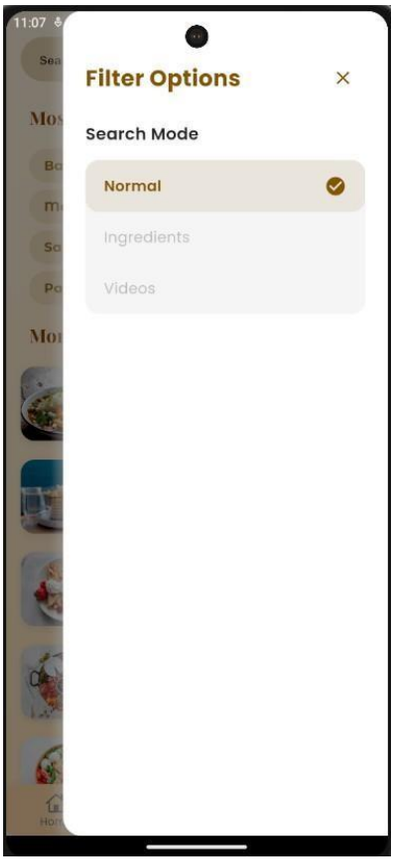


Figure 5.2.3.2 – Filter Options (Search Mode)

On the search page, users can view the most searched recipes, followed by a sorted list of food recipe categories (Figure 5.2.3.1 shows the Search Page). Users can choose from three different search modes: Normal, Ingredients, and Videos (Figure 5.2.3.2 shows Filter Options - Search Mode). By clicking on any category, users will be navigated to a page that displays the relevant food recipes.

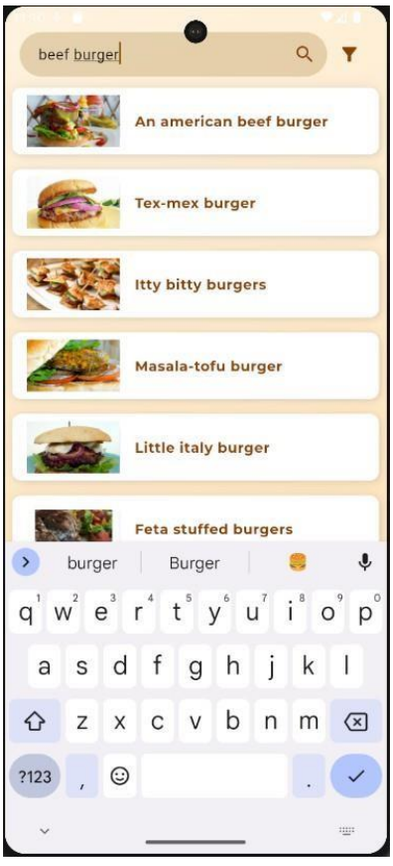


Figure 5.2.3.3 – Search Mode (Normal)

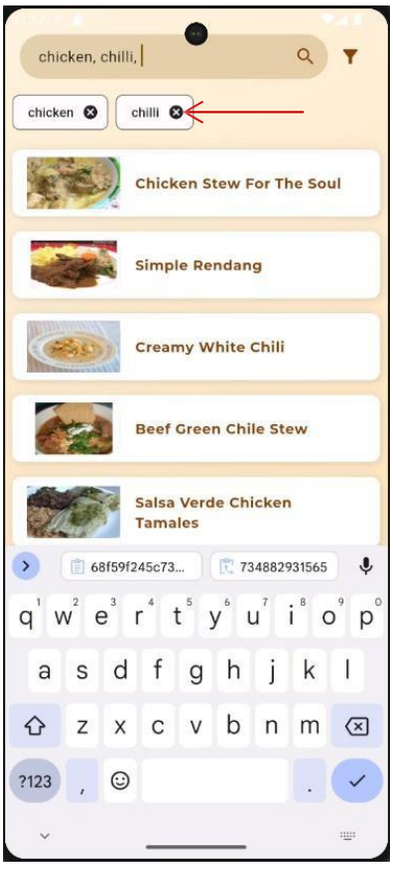


Figure 5.2.3.4 – Search Mode (Ingredients)

The default search mode is set to "Normal Search Mode," allowing users to find recipes by entering the recipe name into the search box (Figure 5.2.3.3 shows Search Mode - Normal). The "Ingredient Search Mode" enables users to search for recipes based on ingredient names (Figure 5.2.3.4 shows Search Mode - Ingredients). To search with multiple ingredients, users can separate each ingredient with a comma. Additionally, users can remove ingredients by tapping the "x" icon next to each ingredient name in the list (Figure 5.2.3.4 shows Search Mode - Ingredients).

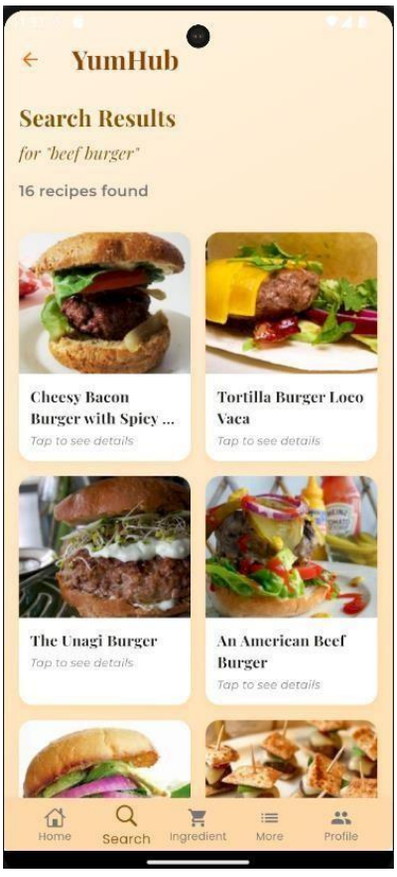


Figure 5.2.3.5 – Search Results (Normal)

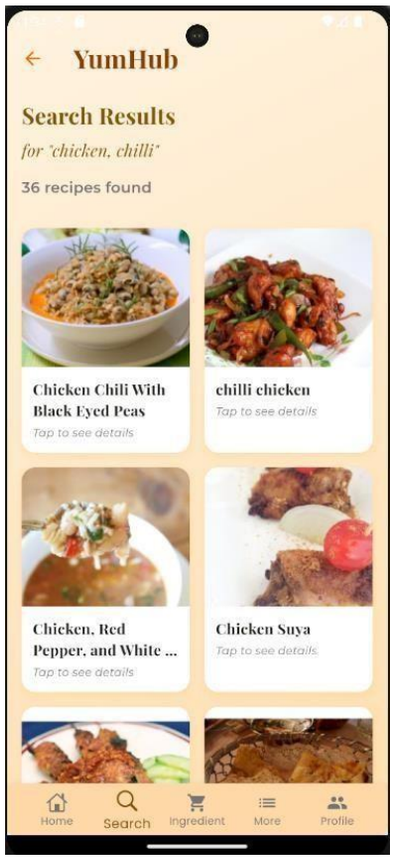


Figure 5.2.3.6 – Search Results (Ingredients)

When users tap the "Enter" button on the phone's on-screen keyboard, the "Search Results" page will be displayed. This page shows the number of recipes found based on the user's input and presents a list of recipe widgets (Figure 5.2.3.5 shows Search Results - Normal; Figure 5.2.3.6 shows Search Results - Ingredients). By selecting any of the recipe widgets on the "Search Results" page, users will be directed to the detailed recipe information page.

Figure 5.2.3.7 – Invalid Input (Normal)

Figure 5.2.3.8 – Invalid Input (Ingredients)

If a user enters an invalid search term or tries to find a recipe name that does not exist in the API's database, a message will be displayed saying, "No recipes found for keyword. Try a different search term." This message will appear in the "Search Results" (Figure 5.2.3.7 shows Invalid Input - Normal; Figure 5.2.3.8 shows Invalid Input - Ingredients).

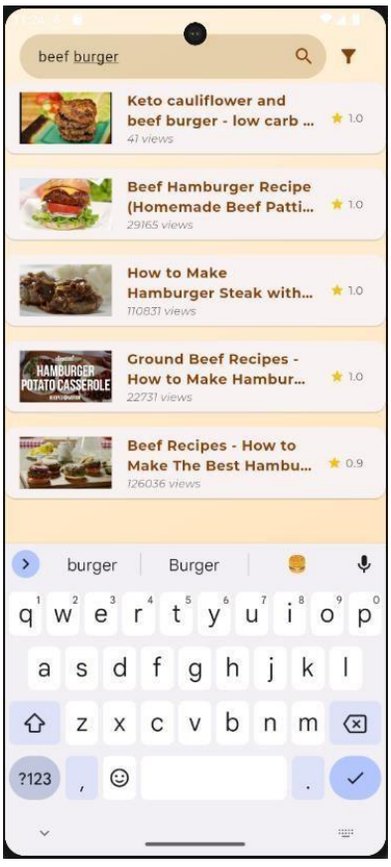


Figure 5.2.3.9 – Search Mode (Videos)

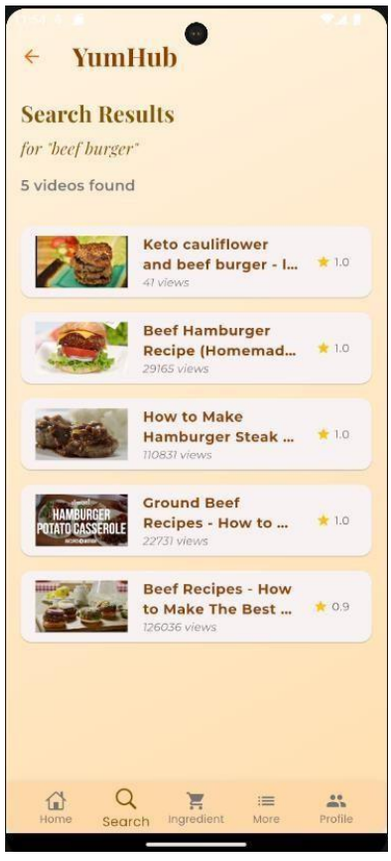


Figure 5.2.3.10 – Search Results (Videos)

For the final search mode in the "Filter Options," after selecting "Videos" in Filter Options – Search Mode (Figure 5.2.3.2), users can search for recipe videos by entering the recipe name into the search box (Figure 5.2.3.9 shows Search Mode - Videos). When users tap the "Enter" button on the phone's on-screen keyboard, the "Search Results" page will be displayed. This page shows the number of videos found based on the user's input and presents a list of recipe video widgets (Figure 5.2.3.10 shows Search Results - Videos).

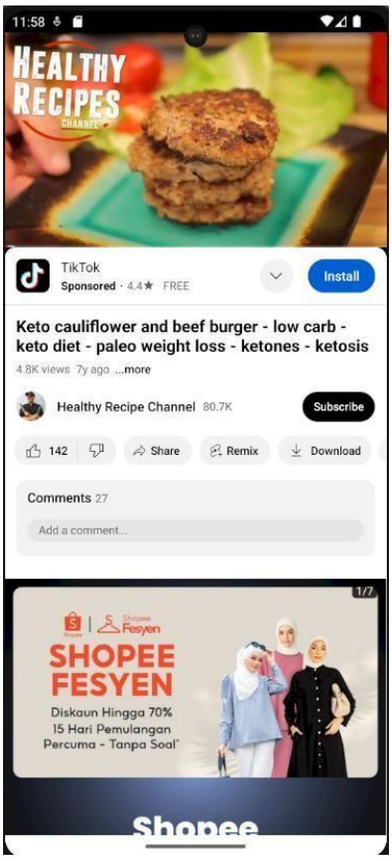


Figure 5.2.3.11 – YouTube Redirect (Videos)

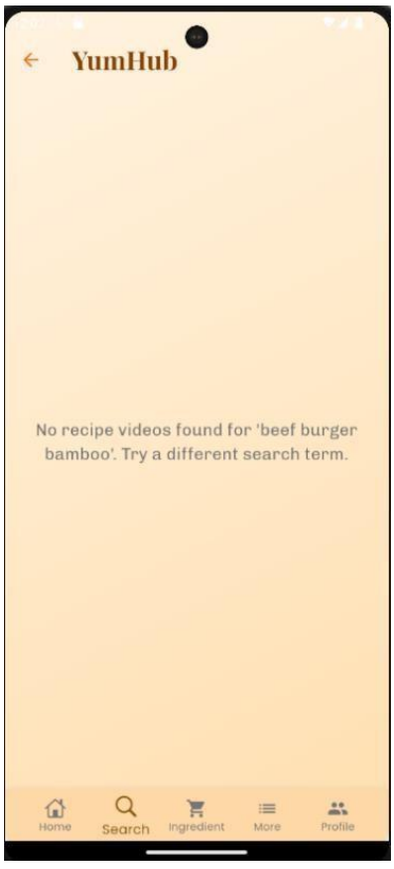


Figure 5.2.3.12 – Invalid Input (Videos)

By selecting any of the widgets in the "Search Results," users will be redirected to the YouTube page of the selected recipe videos (Figure 5.2.3.11 shows YouTube Redirect - Videos). If a user enters an invalid search term or tries to find a recipe name that does not exist in the API's database, a message will be displayed saying, "No recipes found for keyword. Try a different search term." This message will appear in the "Search Results" (Figure 5.2.3.12 shows Invalid Input - Videos).

5.2.4 Grocery List

Figure 5.2.4.1 – Grocery List page (Empty)

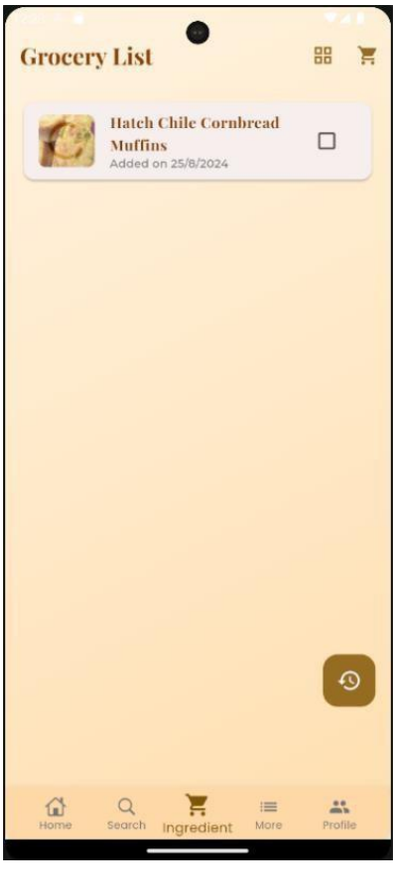


Figure 5.2.4.2 – Grocery List (Default view)

On the grocery list page, if no recipes or ingredients have been added, the page will be empty and display the message “No ingredients added yet” (Figure 5.2.4.1 shows Grocery List - Empty). If ingredients have been added from the recipe information page, the grocery list will show the image, name, and date of the added recipe (Figure 5.2.4.2 shows Grocery List - Default View).

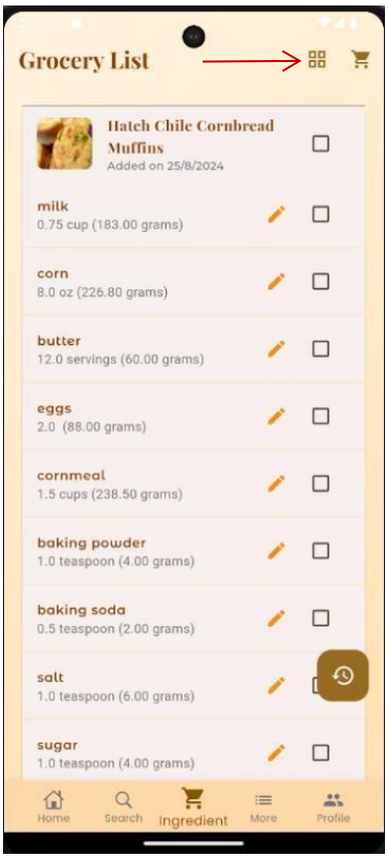


Figure 5.2.4.3 – Grocery List (Default - Expanded)

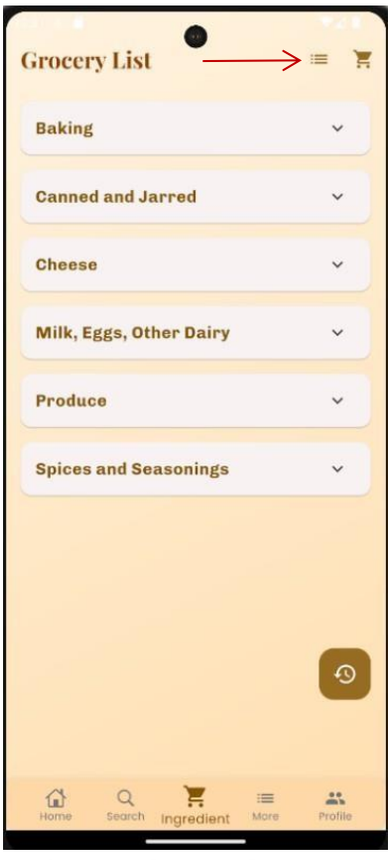


Figure 5.2.4.4 – Grocery List (Combined view)

Users can expand each recipe tile to view the ingredient details, as shown in Figure 5.2.4.3 (Grocery List - Default - Expanded). Ingredient amounts are listed according to the recipe, with two different measurement units displayed. The converted amounts are provided in grams (g) for more accurate measurement during purchasing or cooking.

On the grocery list page, users can switch to the “Combined View” by pressing the combined button, as shown in Figure 5.2.4.3 (Grocery List - Default - Expanded). In the combined view, ingredients are categorized into different groups such as baking, cheese, produce, milk, eggs, and other dairy products, as illustrated in Figure 5.2.4.4 (Grocery List - Combined View). Categorizing ingredients helps users by making grocery shopping more efficient and organized, allowing for quick access to items, reducing time spent in different store sections, and simplifying inventory management and meal planning. Users can switch back to the default view by pressing the button shown in Figure 5.2.4.4 (Grocery List - Combined View).

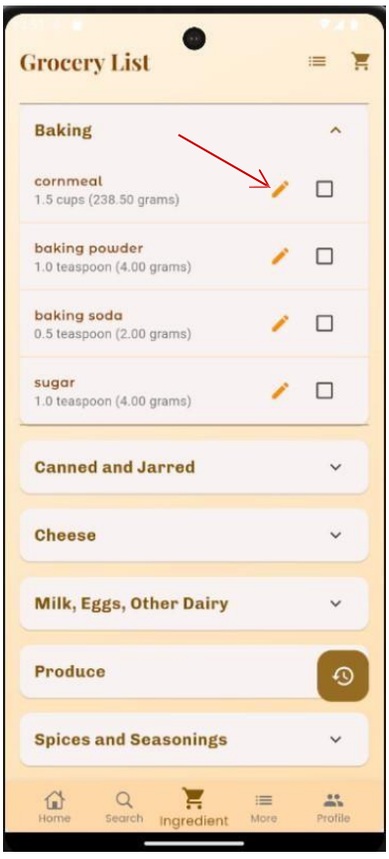


Figure 5.2.4.5 – Grocery List (Combined – Expanded)

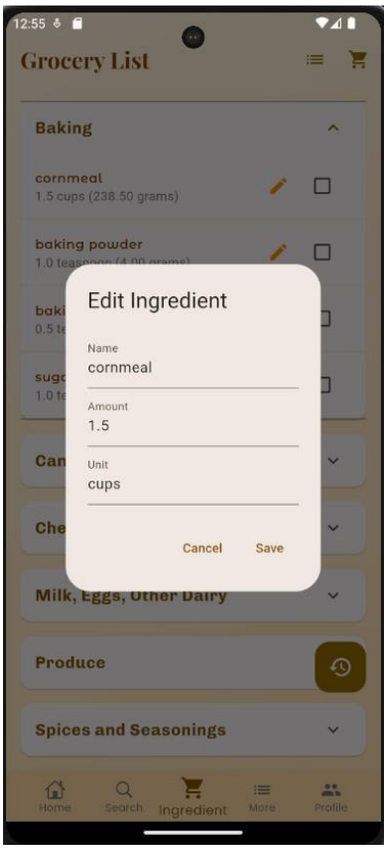


Figure 5.2.4.6 – Grocery List (Edit Ingredient)

Users can expand each recipe tile to view detailed ingredient information, as illustrated in Figure 5.2.4.5 (Grocery List – Combined – Expanded). Ingredient amounts are provided according to the recipe, with measurements displayed in two different units, including a conversion to grams (g) for more accurate measurement during purchasing or cooking. Users can edit these ingredient details in both the default and combined views by tapping the pencil icon, as shown in Figure 5.2.4.6 (shows the Grocery List – Edit Ingredient). Editable details include the ingredient name, amount, and unit.

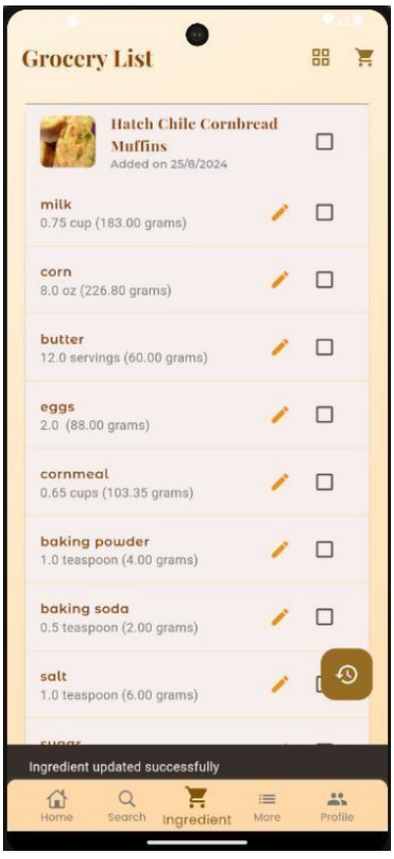


Figure 5.2.4.7 – Grocery List (Edit Ingredient – Success)

Figure 5.2.4.8 – Grocery List (Delete)

After editing the details, users must click the “Save” button, as illustrated in Figure 5.2.4.6 (shows the Grocery List – Edit Ingredient), to save the updated ingredient information. A message will appear stating “Ingredient updated successfully” after pressing the “Save” button, as shown in Figure 5.2.4.7 (Grocery List – Edit Ingredient – Success). To delete an ingredient or recipe, users can swipe left on the tile, as demonstrated in Figure 5.2.4.8 (shows the Grocery List – Delete).

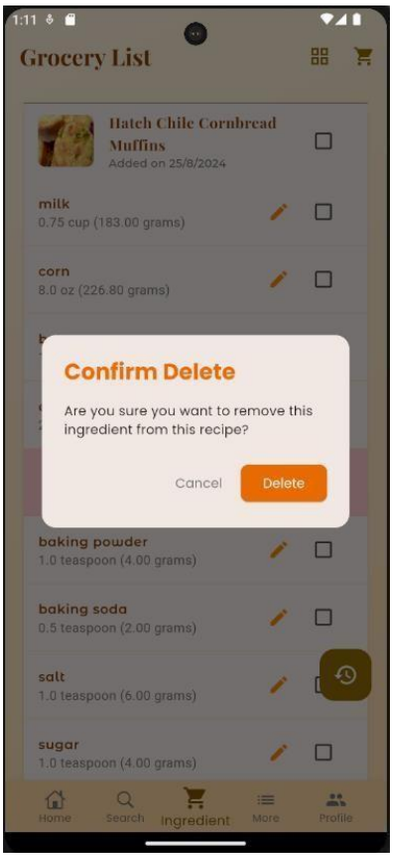


Figure 5.2.4.9 – Grocery List (Delete Confirmation)

Figure 5.2.4.10 – Grocery List (Delete Confirmed)

After swiping left to delete an ingredient or recipe, a confirmation dialog will appear, prompting the user to confirm the deletion, as demonstrated in Figure 5.2.4.9 (Grocery List – Delete Confirmation). If the user presses "Cancel," the deletion will not occur. If the user presses the "Delete" button, the ingredient or recipe will be removed from the list, and a message saying “Ingredient removed successfully” will be displayed, as shown in Figure 5.2.4.10 (Grocery List – Delete Confirmed). This deletion process works the same way in both the default and combined views on the grocery list page.

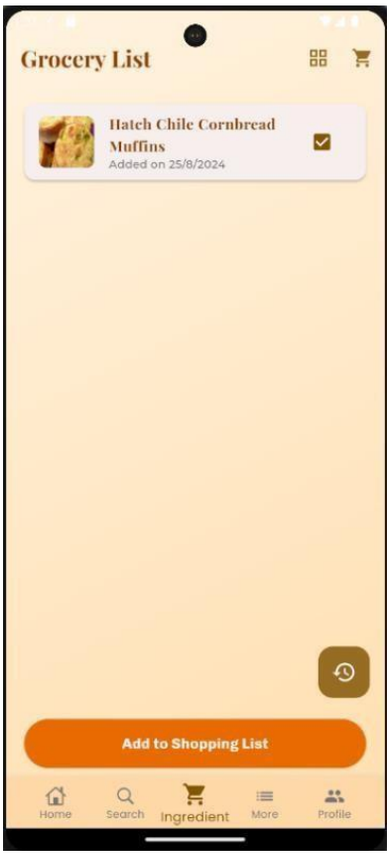


Figure 5.2.4.11 – Grocery List (Default - Add to Shopping List)

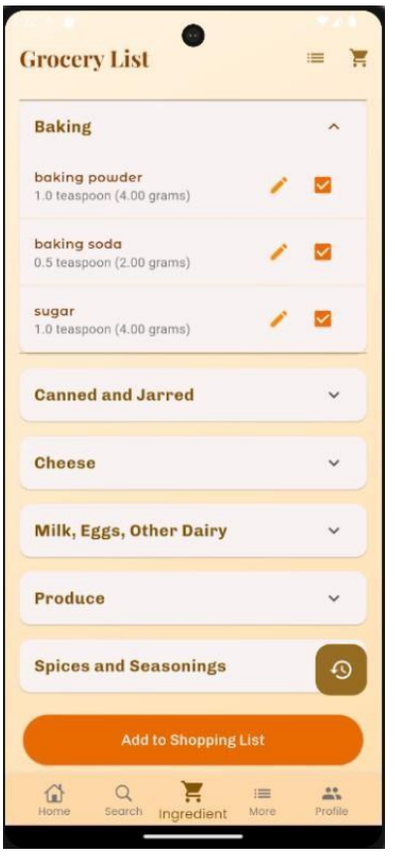


Figure 5.2.4.12 – Grocery List (Combined - Add to Shopping List)

On the grocery list page, users can select recipes or ingredients to add them to the shopping list. In the default view, selecting a recipe will automatically select all its ingredients, as shown in Figure 5.2.4.11 (Grocery List – Default – Add to Shopping List). Users can also manually select individual ingredients under the saved recipe. In the combined view, users need to expand each category tile to select ingredients, as illustrated in Figure 5.2.4.12 (Grocery List – Combined – Add to Shopping List).

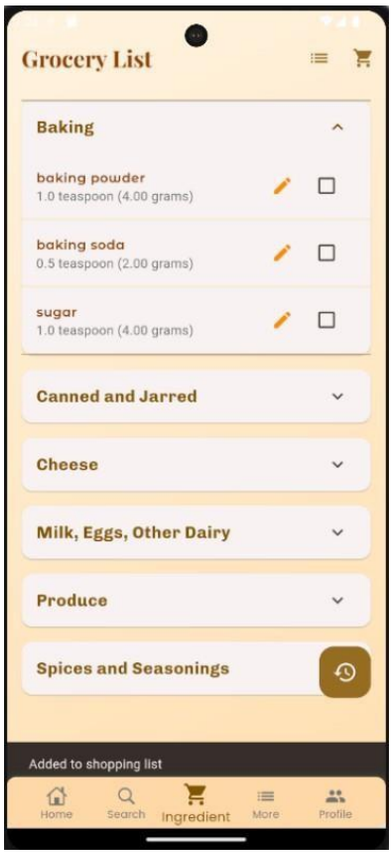


Figure 5.2.4.13 – Grocery List (Add to Shopping List – Message)

After the user clicks the "Add to Shopping List" button, the selected recipe or ingredients will be added to the shopping list, and a message stating "Added to shopping list" will be displayed, as shown in Figure 5.2.4.13 (Grocery List – Add to Shopping List – Message).

5.2.5 Delete History

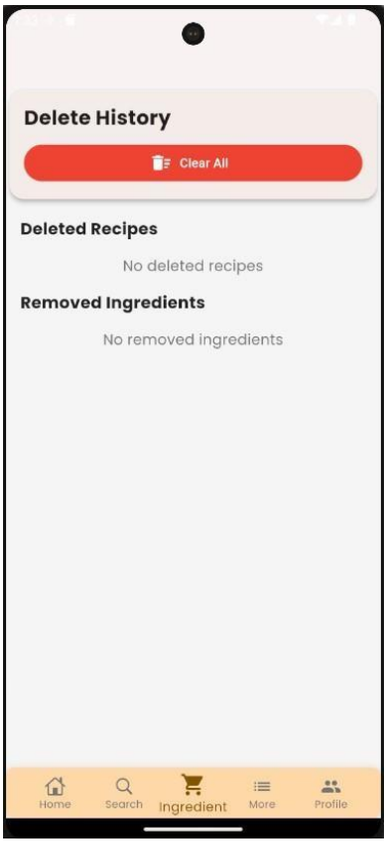


Figure 5.2.5.1 Delete History page (Empty)

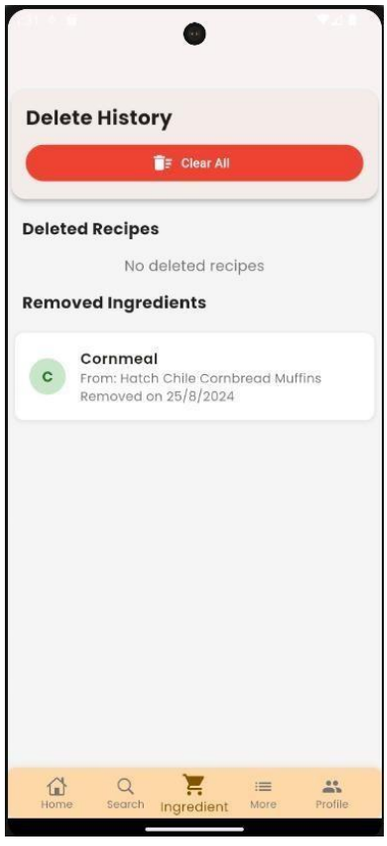


Figure 5.2.5.2 Delete History page

Moreover, the delete history page will show the user's removed recipes and ingredients history. If no recipes or ingredients have been removed, the page will display “No deleted recipes” and “No removed ingredients,” as illustrated in Figure 5.2.5.1 (Delete History Page – Empty). If the user has removed any items, they will be listed on the delete history page, as shown in Figure 5.2.5.2 (Delete History Page).

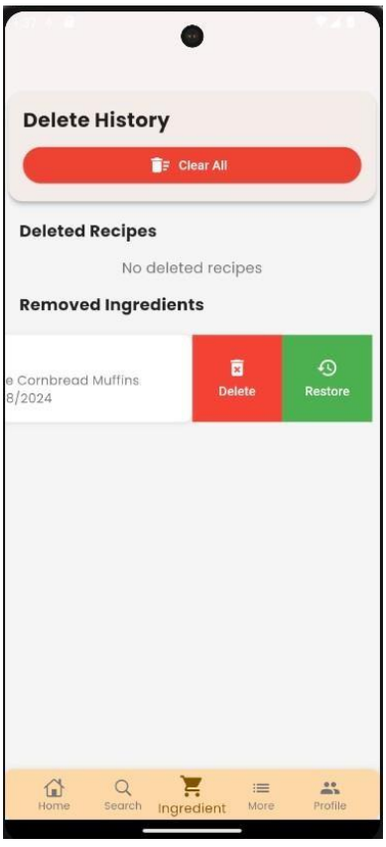


Figure 5.2.5.3 – Delete History (Delete/Restore)

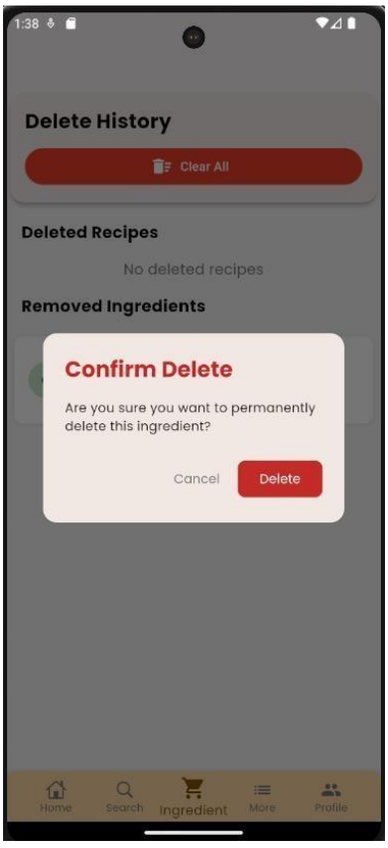


Figure 5.2.5.4 – Delete History (Delete Confirmation)

On the delete history page, users can swipe left on an ingredient or recipe to choose between permanently deleting it or restoring it, as illustrated in Figure 5.2.5.3 (Delete History – Delete/Restore). Selecting the "Delete" button will trigger a confirmation dialog, as shown in Figure 5.2.5.4 (Delete History – Delete Confirmation). Pressing "Cancel" will cancel the deletion process, while pressing "Delete" will remove the item permanently from the system. The "Restore" option, also illustrated in Figure 5.2.5.3 (Delete History – Delete/Restore), will restore the deleted recipe or ingredient to the grocery list when toggled.

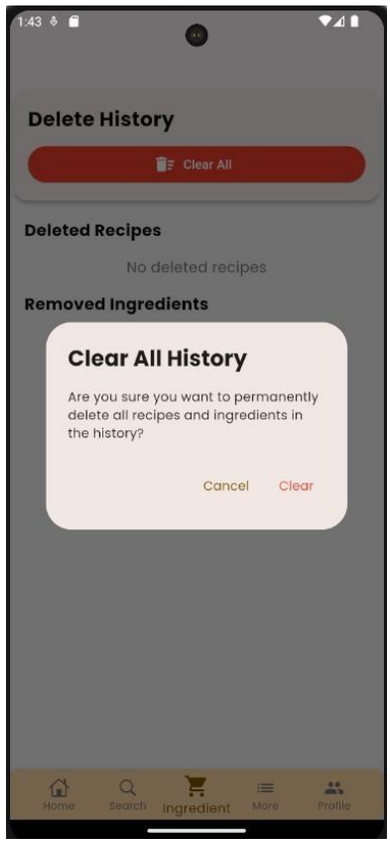


Figure 5.2.5.5 – Delete History (Clear All History)

There is also a “Clear All” button on the delete history page, which allows users to permanently delete all recipes and ingredients listed in the delete history. This functionality is illustrated in Figure 5.2.5.5 (Delete History – Clear All History).

5.2.6 Shopping List

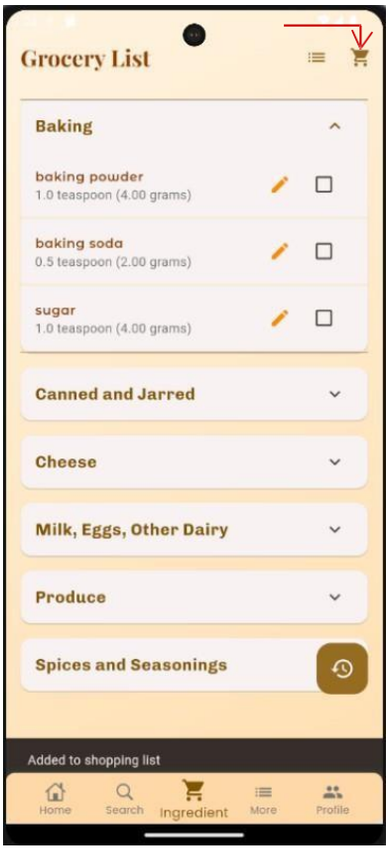


Figure 5.2.6.1 – Grocery List (Shopping List button)

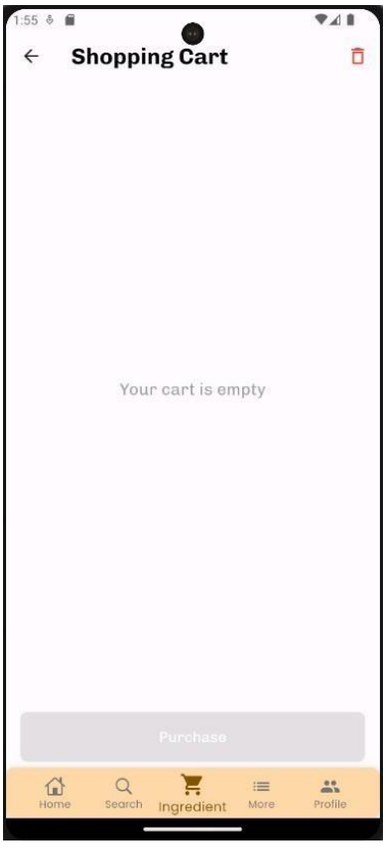


Figure 5.2.6.2 – Shopping List page

The shopping list can be accessed by toggling the “Cart” button, as labeled in Figure 5.2.6.1 (Grocery List – Shopping List Button). This action navigates the user to the shopping list page, as shown in Figure 5.2.6.2 (Shopping List Page). If no ingredients have been added to the shopping list, the page will display the message “Your cart is empty” (Figure 5.2.6.2 shows the Shopping List page).

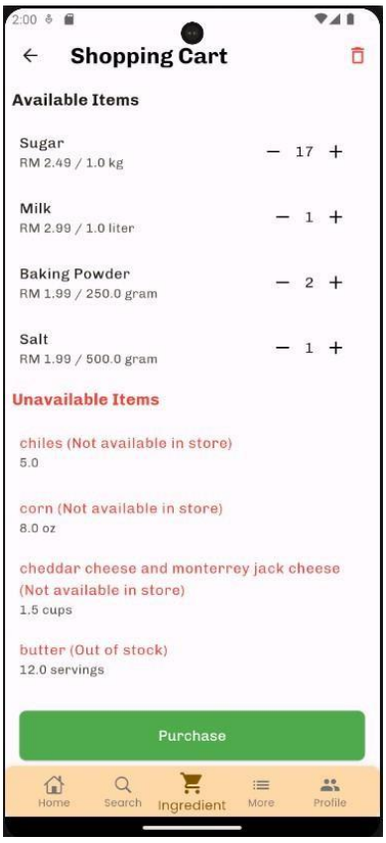


Figure 5.2.6.3 – Shopping List page (Added Items)

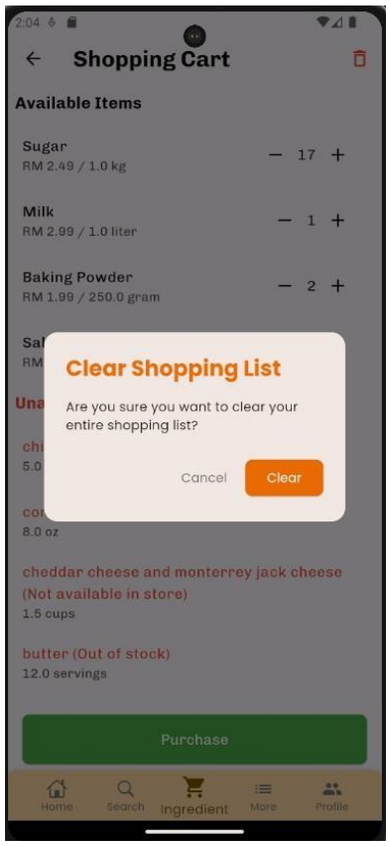


Figure 5.2.6.4 – Shopping List page (Clear Shopping List)

On the shopping list page, users will see two sections: “Available Items” and “Unavailable Items”, Figure 5.2.6.3 shows the (Shopping List Page – Added Items). The “Available Items” section displays items that are in-stock and available in-store, while the “Unavailable Items” section lists items that are out-of-stock or not available in-store. Users can adjust the quantity of items by tapping the “+” and “-” buttons. If the item quantity drops below zero, it will be automatically removed from the shopping list. The quantity can only be increased up to the current stock level. To clear the shopping list, users can click the red “Delete” button in the top right corner, Figure 5.2.6.3 shows the (Shopping List Page – Added Items). This action will prompt a confirmation dialog, Figure 5.2.6.4 shows the (Shopping List Page – Clear Shopping List). Selecting “Clear” will empty the shopping list, while “Cancel” will abort the action.

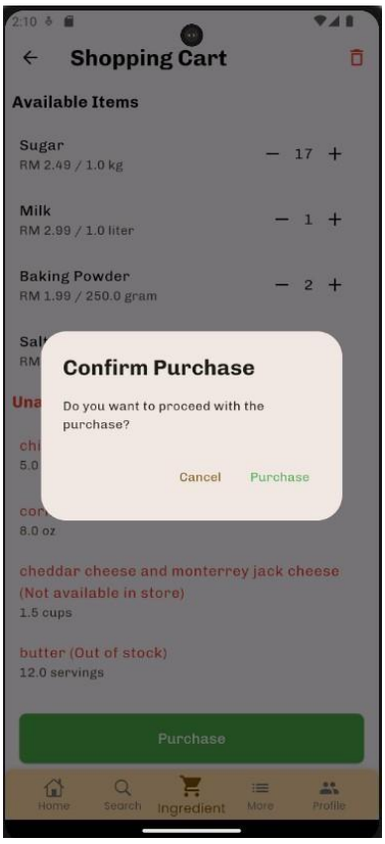


Figure 5.2.6.5 – Shopping List page (Purchase Confirmation)

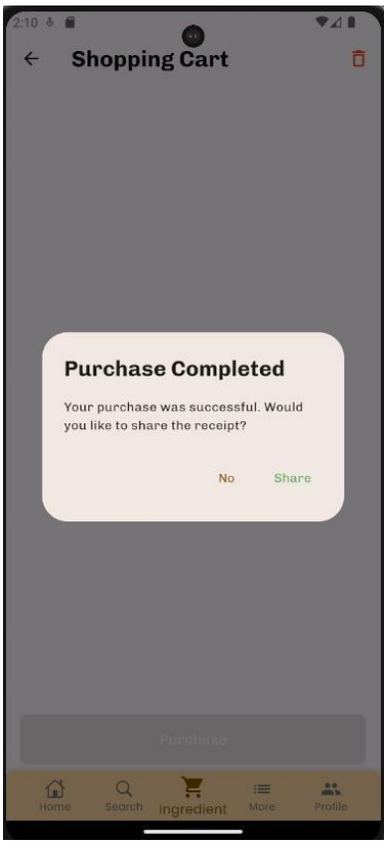


Figure 5.2.6.6 – Shopping List page (Share Receipt)

After deciding to purchase the items, the user can click the green “Purchase” button, Figure 5.2.6.3 shows the (Shopping List Page – Added Items). This action will prompt a purchase confirmation dialog, Figure 5.2.6.5 shows the (Shopping List Page – Purchase Confirmation). If the user presses “Purchase,” a share receipt dialog will appear, Figure 5.2.6.6 shows the (Shopping List Page – Share Receipt). Pressing “Cancel” will abort the purchase action.

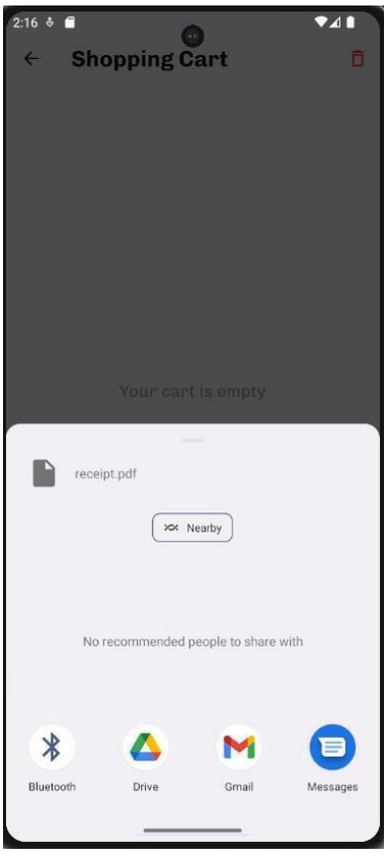


Figure 5.2.6.7 – Share to Media

YumHub

Purchase Receipt

Date: 2024-08-25 14:10:54

Item	Qty	Price	Total
Sugar	17	RM 2.49	RM 42.33
Milk	1	RM 2.99	RM 2.99
Baking Powder	2	RM 1.99	RM 3.98
Salt	1	RM 1.99	RM 1.99

Total Amount: **RM 51.29**

Thank you for your purchase!

Please come again

Figure 5.2.6.8 – Purchase Receipt (PDF)

After confirming the purchase, if the user presses the “Share” button, the application will display media sharing options, Figure 5.2.6.7 shows the (Share to Media). The receipt will be provided in “.pdf” format, Figure 5.2.6.8 shows the (Purchase Receipt – PDF).

5.2.7 More

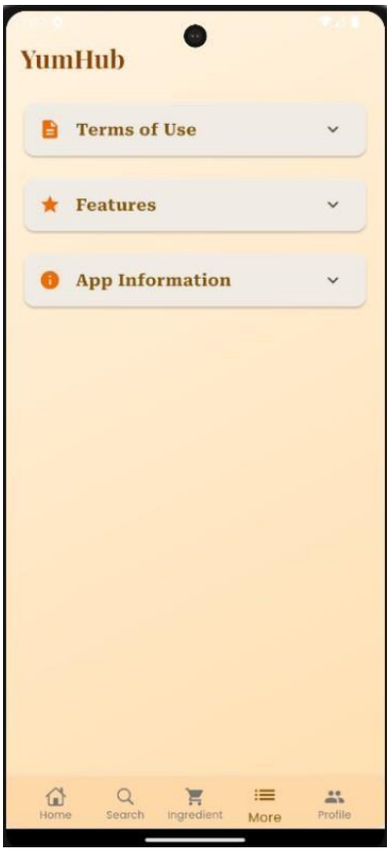


Figure 5.2.7.1 – More page

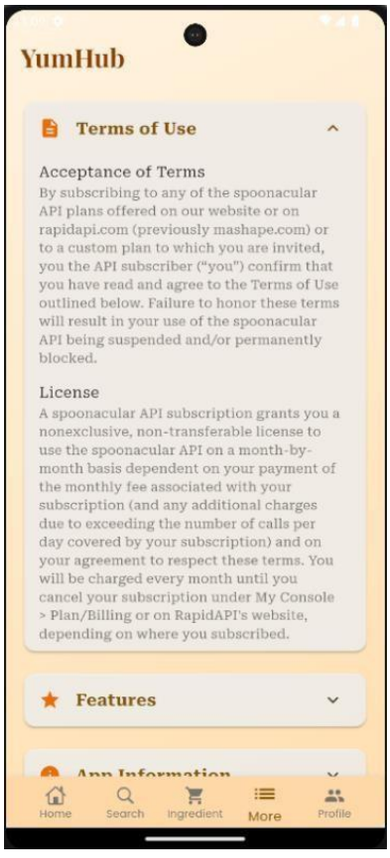


Figure 5.2.7.2 – More page (Terms of Use)

In Figure 5.2.7.1 (More-page), information related to the application is displayed, including sections like "Terms of Use," "Features," and "App Information." The "Terms of Use" section outlines the acceptance of terms and licenses provided by the Spoonacular API, Figure 5.2.7.2 shows the (More page – Terms of Use).

Figure 5.2.7.3 – More page (Features)

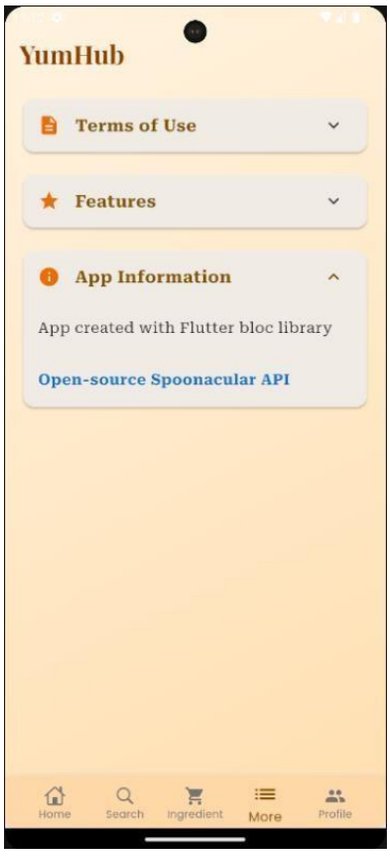


Figure 5.2.7.4 – More page (App Information)

The “Features” section includes a list of features provided by the application, such as Browse Recipes, Advanced Recipe Searching, and more, Figure 5.2.7.3 shows the (More page – Features). The final tile, “App Information,” displays details about the app’s development using the Flutter BLoC library and includes a hyperlink to the Spoonacular API website, Figure 5.2.7.4 shows the (More page – App Information).

5.2.8 Profile

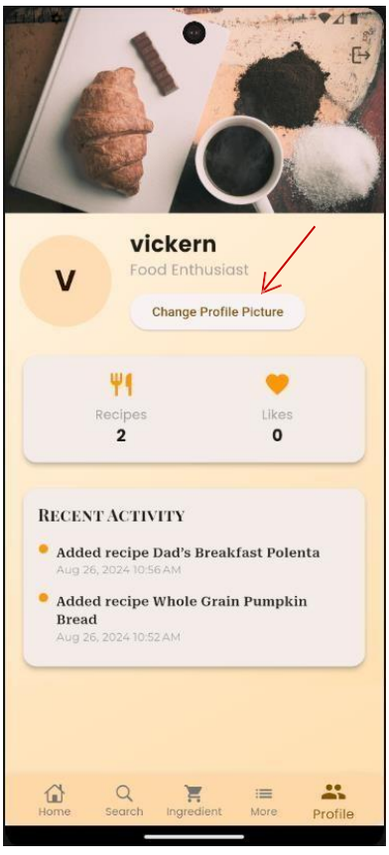


Figure 5.2.8.1 – Profile page

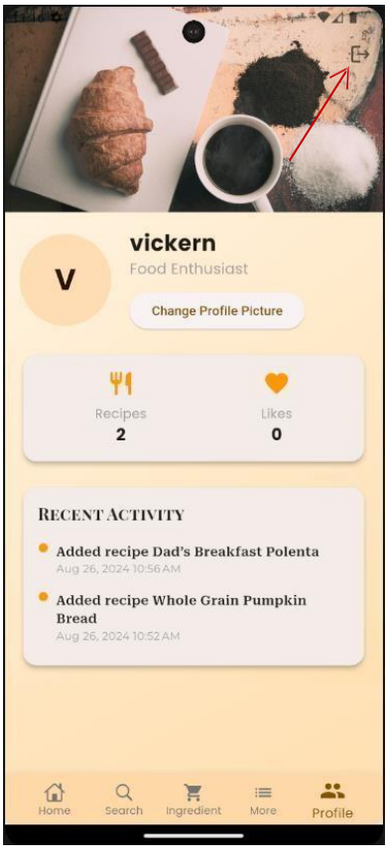


Figure 5.2.8.2 – Profile page (Logout)

On the profile page, users can view their username, the number of saved recipes, and change their profile picture. To change the profile picture, users can tap the "Change Profile Picture" button, as labeled in Figure 5.2.8.1 (Profile page). This action will redirect the users to their photo gallery to select a new image. Below this, there is a "Recent Activity" section where users can view their recent grocery list activities, including recently added or removed recipes, along with the date and time of these actions, Figure 5.2.8.1 shows the (Profile page). Users can log out by pressing the "Logout" button labeled in Figure 5.2.8.2 (Profile page – Logout), which will log the users out of their account and redirect the users to the "Login page."

5.3 Concluding Remark

System implementation refers to the set of procedures followed to develop the system based on the system design specifications. In section 5.1, the techniques, solutions, settings, and configurations implemented in the system are thoroughly discussed. This section provides a comprehensive overview of the technical aspects of the system's development. In addition to the technical details, this chapter showcases the implemented user interfaces through a series of screenshots. These visual representations offer a clear understanding of the system's look and feel, as well as its overall design aesthetic. Accompanying the screenshots are explanations of the system's operation, providing insights into how users interact with the various features and functionalities.

CHAPTER 6 System Evaluation and Discussion

6.1 System Performance Evaluation

The performance evaluation of the mobile food recipe application aims to assess its responsiveness, resource utilization, and user experience under different usage scenarios. This evaluation ensures that the application meets expectations, performs efficiently on various devices, and can handle a growing user base and expanding recipe database. Conducting a thorough performance evaluation helps identify potential bottlenecks, optimize resource allocation, and ensure a smooth and enjoyable experience as users explore recipes, manage grocery lists, and engage with the application's features.

6.1.1 Network Performance

This section focuses on quantifying and understanding the response times of GET requests in the mobile food recipe application. The analysis covers various functions that interact with external services, primarily the Spoonacular API and Firebase. Each of the response time was recorded for 10 times to get the average response time value. Flutter DevTools was employed to capture and analyze network traffic during typical application usage scenarios. This tool provided detailed metrics on request durations, status codes, and data types, enabling a comprehensive evaluation of the application's network performance.

6.1.1.1 Request Recipe Data (Spoonacular API)

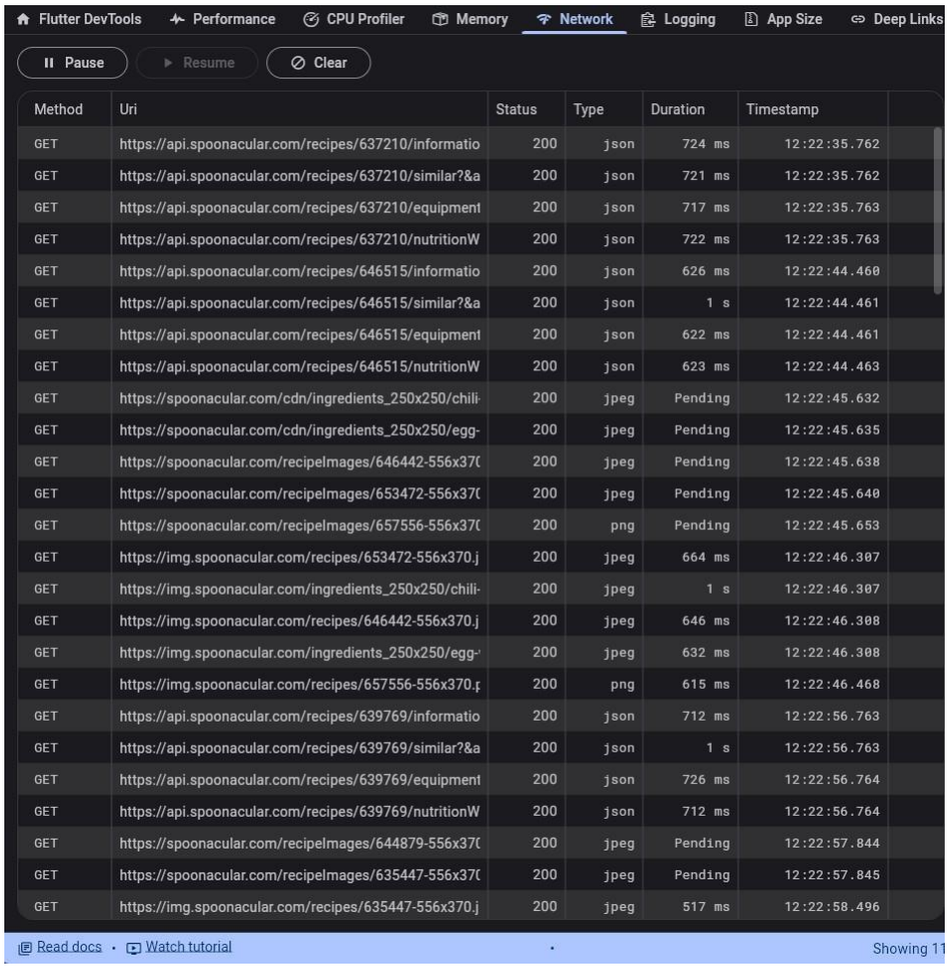


Figure 6.1.1.1 Request Recipe Data (Spoonacular API)

Based on the data captured in Figure 6.1.2.1, which shows the Request Recipe Data (Spoonacular API) for the [F005] Browse Recipe function, [F007] Search Recipe by Name function, and [F008] View and Modify Saved Recipe & Ingredients, the following observations can be made:

Response Time:

- The average response time for GET requests is 1.9 seconds, indicating moderate API performance considering the diverse functions being served.
- This response time contributes to a generally acceptable user experience across browsing, searching, and managing saved recipes, though there may be room for optimization.

Consistency:

- All visible requests in the figure returned a 200-status code.
- This consistency suggests reliable API availability and correct request formatting across all three functions (Browse Recipe, Search Recipe by Name, View and Modify Saved Recipe & Ingredients).

Data Types:

- The majority of requests fetch JSON data, which is lightweight and quick to parse.
- Image requests (JPEG, PNG) are also present and naturally take longer due to larger file sizes.
- This mix of data types is typical for a recipe browsing feature, balancing textual information with visual elements.

Variations in Request Duration:

- Some requests show significantly longer durations (e.g., 724ms, 721ms), which was caused by larger image file sizes.
- A few requests display very short durations (approximately 1ms), potentially indicating cached responses or minimal data transfer.
- These variations highlight the diverse nature of data being fetched during recipe browsing.

Overall Performance:

- The consistent 200 status codes across all functions ([F005] Browse Recipe, [F007] Search Recipe by Name, and [F008] View and Modify Saved Recipe & Ingredients) indicate robust error handling and reliable API interactions within the application.
- While the average response time of 1.9 seconds is acceptable, it may be noticeable to users on mobile devices where faster interactions are often expected. Further optimization could enhance the user experience, especially for operations involving larger data transfers like image loading.
- The presence of very quick responses suggests effective use of local data management strategies, which is crucial for maintaining responsiveness in a mobile application.

6.1.1.2 Retrieve User Profile Image (Firebase)

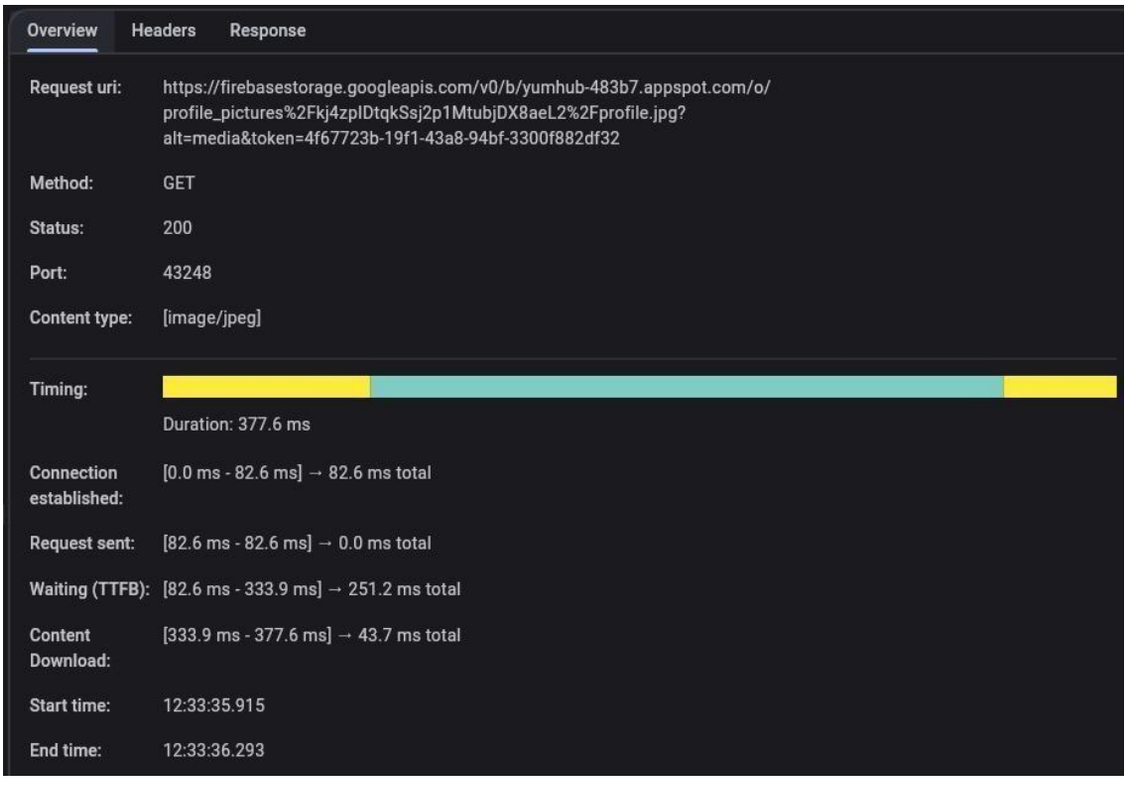


Figure 6.1.1.2 Retrieve User Profile Image (Firebase)

Based on Figure 6.1.1.2, which shows the Retrieve User Profile Image (Firebase) for the [F014] Edit Profile Picture function, the following observations can be made:

Response Time:

- The total duration for the GET request is 377.6 ms, indicating fast API performance for retrieving profile picture data from Firebase Storage.
- This quick response time contributes to a smooth user experience when editing profile pictures, allowing for near-instantaneous image preview and updates.

Consistency:

- The request returned a 200 status code, suggesting reliable Firebase Storage availability and correct request formatting for the Edit Profile Picture function.

Data Type:

- The content type is [image/jpeg], which is appropriate for profile pictures and typically offers a good balance between image quality and file size for mobile applications.

Request Breakdown:

- Connection established: 82.6 ms
- Request sent: 0.0 ms (negligible time)
- Waiting (Time to First Byte): 251.2 ms
- Content Download: 43.7 ms

Overall Performance:

- The consistent 200 status code indicates robust error handling and reliable interaction with Firebase Storage for profile picture management.
- The breakdown of the request timing shows that the majority of the time (251.2 ms) is spent waiting for the initial response from the server, while the actual content download is relatively quick (43.7 ms).
- The total response time of 377.6 ms is well within the range of what users would perceive as instantaneous, contributing to a responsive feel when interacting with profile picture functionalities.
- The use of Firebase Storage for profile pictures appears to be an efficient choice, offering fast retrieval times suitable for a mobile application context.

6.2 System Testing Setup and Result

6.2.1 [F001] Sign Up Testing

Sign Up Test Case				
Use Case: Sign Up				
Function Id: F001				
Date Created: 26/8/2024				
Role: User				
No.	**Input Values**	**Expected Results**	**Actual Results**	**Pass/Fail**
1	Click the "Sign Up" button on the Login page.	The system requests the user details.	The system requests the user details.	Passed.
2	Click the "Register" button after entering the user details.	The system displays a message informing the user that the account is registered.	The system displays a message informing the user that the account is registered.	Passed.
3	Click the "Google" icon button on the Sign Up page.	The system will forward the user to Google Sign in page.	The system will forward the user to Google Sign in page.	Passed.
4	Enter an invalid email address.	The system displays an error message and asks the user to modify it.	The system displays an error message and asks the user to modify it.	Passed.
5	Enter password with a length less than 6	The system displays an error message and ask the user to modify it.	The system displays an error message and ask the user to modify it.	Passed.
6	Username not entered.	The system displays an error message and ask the user to modify it.	The system displays an error message and ask the user to modify it.	Passed.
7	Email not entered.	The system displays an error message and ask the user to modify it.	The system displays an error message and ask the user to modify it.	Passed.

8	Password not entered.	The system displays an error message and ask the user to modify it.	The system displays an error message and ask the user to modify it.	Passed.

Table 6.1 Sign Up Testing

6.2.2 [F002] Login Testing

Login Test Case				
Use Case: Login				
Function Id: F002				
Date Created: 26/8/2024				
Role: User				
No.	**Input Values**	**Expected Results**	**Actual Results**	**Pass/Fail**
1	Open the application.	The system will navigate the user to login page if the user is not logged in.	The system will navigate the user to login page if the user is not logged in.	Passed.
2	Click the “Forget Password” button on the Login page.	The system forwards the user to reset the password.	The system forwards the user to reset the password.	Passed.
3	Click the “Email” icon button.	The system forwards the user to the Sign Up page.	The system forwards the user to the Sign Up page.	Passed.
4	Click the “Google” icon button.	The system will forward the user to Google Sign in page.	The system will forward the user to Google Sign in page.	Passed.
5	Click the “Login” button after entering the email and password.	The system will navigate the user to home page after login successful.	The system will navigate the user to home page after login successful.	Passed.
6	Enter an invalid email address.	The system displays an error message and asks the user to modify it.	The system displays an error message and asks the user to modify it.	Passed.
7	Enter an invalid password.	The system displays an error message	The system displays an error message and asks	Passed.

		and asks the user to modify it.	the user to modify it.	
8	The email and password are not entered.	The system displays an error message and asks the user to modify it.	The system displays an error message and asks the user to modify it.	Passed.

Table 6.2 Login Testing

6.2.3 [F003] Reset Password Testing

Reset Password Test Case				
Use Case: Reset Password				
Function Id: F003				
Date Created: 26/8/2024				
Role: User				
No.	**Input Values**	**Expected Results**	**Actual Results**	**Pass/Fail**
1	Click the "Forgot Password" button on the Login page.	The system requests the user to enter an email address.	The system requests the user to enter an email address.	Passed.
2	Click the "Submit" button after entering the email address.	The system sends an email to the user, and the user receives the email to reset the password.	The system sends an email to the user, and the user receives the email to reset the password.	Passed.
3	Enter an invalid email.	The system displays an error message and asks the user to modify it.	The system displays an error message and asks the user to modify it.	Passed.

Table 6.3 Reset Password Testing

6.2.4 [F004] Logout Testing

Logout Test Case				
Use Case: Logout				
Function Id: F004				
Date Created: 26/8/2024				
Role: User				
No.	**Input Values**	**Expected Results**	**Actual Results**	**Pass/Fail**
1	Click the "Logout" button on the Profile page.	The system signs out the user and displays the Login page.	The system signs out the user and displays the Login page.	Passed.

Table 6.4 Logout Testing

6.2.5 [F005] Browse Recipes Testing

Browse Recipes Test Case				
Use Case: Browse Recipes				
Function Id: F005				
Date Created: 26/8/2024				
Role: User				
No.	**Input Values**	**Expected Results**	**Actual Results**	**Pass/Fail**
1	Click any recipe widget in home page, similar recipes, or categorized recipes.	The system display recipe information page with recipe image, title, ratings, preparation time, servings, price/servings, instructions, utensils, and nutrition information.	The system display recipe information page with recipe image, title, ratings, preparation time, servings, price/servings, instructions, utensils, and nutrition information.	Passed.
2	Click "Share Recipe" button on top right corner of the recipe information page.	The system will display a confirmation dialog. After the user confirmed the request, the media	The system will display a confirmation dialog. After the user confirmed the request, the media	Passed.

		sharing dialog will be prompted.	sharing dialog will be prompted.	
3	Click the ingredient.	The system will display the ingredient's details such as type, consistency and amount.	The system will display the ingredient's details such as type, consistency and amount.	Passed.

Table 6.5 Browse Recipes Testing

6.2.6 [F006] Save Recipe and Ingredients Testing

Save Recipes and Ingredients Test Case				
Use Case: Save Recipes and Ingredients				
Function Id: F006				
Date Created: 26/8/2024				
Role: User				
No.	**Input Values**	**Expected Results**	**Actual Results**	**Pass/Fail**
1	Click "+" button in the recipe information page. After the button expanded, click the "Add All" button.	The system will display ingredients added to grocery list message and call the unit conversion service before adding into the grocery list.	The system will display ingredients added to grocery list message and call the unit conversion service before adding into the grocery list.	Passed.

Table 6.6 Save Recipes and Ingredients Testing

6.2.7 [F007] Search Recipe by Name Testing

Search Recipe by Name Test Case				
Use Case: Search Recipe by Name				
Function Id: F007				
Date Created: 26/8/2024				
Role: User				
No.	**Input Values**	**Expected Results**	**Actual Results**	**Pass/Fail**

1	Click “Search” icon on navigation bar.	The system requests the search input.	The system requests the search input.	Passed.
2	Click “enter” button on the on-screen keyboard after entering invalid input.	The system displays an no recipes found and asks the user to modify it.	The system displays an no recipes found and asks the user to modify it.	Passed.
3	Toggle Search by Name option, and enter valid input in search box.	The system will display search result underneath the search box.	The system will display search result underneath the search box.	Passed.
4	Toggle Search by Ingredient option, and enter valid input in search box.	The system will display search result underneath the search box.	The system will display search result underneath the search box.	Passed.
5	Toggle Videos option, and enter valid input in search box.	The system will display search result underneath the search box.	The system will display search result underneath the search box.	Passed.
6	Click any recipe category in the “Most Searched By People”.	The system will display a list of recipes that is related to the category name.	The system will display a list of recipes that is related to the category name.	Passed.
7	Click any category in the “More Categories”.	The system will display a list of recipes that is related to the category name.	The system will display a list of recipes that is related to the category name.	Passed.
8	Click “enter” button on the on-screen keyboard after entering valid input.	The system will display a list of recipes that is related to the search input.	The system will display a list of recipes that is related to the search input.	Passed.

Table 6.7 Search Recipes by Name Testing

6.2.8 [F008] View and Modify Saved Recipes and Ingredients Testing

View and Modify Saved Recipe and Ingredients Test Case				
Use Case: View and Modify Saved Recipes and Ingredients				
Function Id: F008				
Date Created: 26/8/2024				
Role: User				
No.	**Input Values**	**Expected Results**	**Actual Results**	**Pass/Fail**
1	Click the ingredient icon on the navigation bar.	The system will display a list of saved recipes and ingredients in the grocery list.	The system will display a list of saved recipes and ingredients in the grocery list.	Passed.
2	Edit the ingredient details and click save.	The system will display the ingredient updated successfully message and conversion service will be called. After conversion, the grocery list UI will be refreshed to update.	The system will display the ingredient updated successfully message and conversion service will be called. After conversion, the grocery list UI will be refreshed to update.	Passed.
3	Swipe left on the recipe or ingredient tile.	The system will prompt a delete confirmation dialog.	The system will prompt a delete confirmation dialog.	Passed.
4	Click “Confirm” in the delete confirmation dialog.	The system will remove the recipe or ingredient from the grocery list.	The system will remove the recipe or ingredient from the grocery list.	Passed.
5	Click the “Categorized view” on the top right corner of the grocery list page.	The system will display the ingredients in combined category view.	The system will display the ingredients in combined category view.	Passed.
6	No recipes added into grocery list.	The system will display message saying “there is no ingredients added into grocery list yet”.	The system will display message saying “there is no ingredients added into grocery list yet”.	Passed.

7	Long press any recipes tile in default view.	The system navigates users to the recipe information page.	The system navigates users to the recipe information page.	Passed.

Table 6.8 View and Modify Saved Recipes and Ingredients Testing

6.2.9 [F009] View Shopping List Testing

View Shopping List Test Case				
Use Case: View Shopping List Testing				
Function Id: F009				
Date Created: 26/8/2024				
Role: User				
No.	**Input Values**	**Expected Results**	**Actual Results**	**Pass/Fail**
1	Click “Cart” icon in grocery list page.	The system will display shopping list page.	The system will display shopping list page.	Passed.
2	Added items are available.	The system will display items under “Available Items”.	The system will display items under “Available Items”.	Passed.
3	Added items are not available.	The system will display items under “Unavailable Items”.	The system will display items under “Unavailable Items”.	Passed.
4	Click “Clear” button.	The system will forward to modify shopping list.	The system will forward to modify shopping list.	Passed.
5	Click “+” or “-” button.	The system will forward to modify and delete shopping list.	The system will forward to modify and delete shopping list.	Passed.

Table 6.9 View Shopping List Testing

6.2.10 [F010] View History Testing

View History Test Case				
Use Case: View History				
Function Id: F010				
Date Created: 26/8/2024				
Role: User				
No.	**Input Values**	**Expected Results**	**Actual Results**	**Pass/Fail**
1	Click the "History" icon in grocery list page.	The system will display delete history page.	The system will display delete history page.	Passed.
2	Swipe left on recipe or ingredient tile.	The system will display delete and restore option on the tile.	The system will display delete and restore option on the tile.	Passed.
3	Click the "delete" option after swiping left on the recipe or ingredient tile.	The system will prompt a delete confirmation dialog.	The system will prompt a delete confirmation dialog.	Passed.
4	Click "Confirm" in the delete confirmation dialog.	The system will remove the recipe or ingredient.	The system will remove the recipe or ingredient.	Passed.
5	Click the "restore" option after swiping left on the recipe or ingredient tile.	The system will recover the removed recipe or ingredients to the grocery list.	The system will recover the removed recipe or ingredients to the grocery list.	Passed.
6	Click "Clear All" button.	The system will prompt a delete confirmation dialog.	The system will prompt a delete confirmation dialog.	Passed.
7	Click "Confirm" in the delete confirmation dialog.	The view history page will become empty.	The view history page will become empty.	
8	Open delete history page without removing any recipes or ingredients.	The system will display no deleted recipes and no deleted ingredients.	The system will display no deleted recipes and no deleted ingredients.	Passed.

Table 6.10 View History Testing

6.2.11 [F011] Add to Shopping List Testing

Add to Shopping List Test Case				
Use Case: Add to Shopping List				
Function Id: F011				
Date Created: 26/8/2024				
Role: User				
No.	**Input Values**	**Expected Results**	**Actual Results**	**Pass/Fail**
1	After selecting ingredients in grocery list page, click “Add to Shopping List” button.	The system will display “Added to shopping list” message.	The system will display “Added to shopping list” message.	Passed.

Table 6.11 Add to Shopping List Testing

6.2.12 [F012] Purchase Testing

Purchase Test Case				
Use Case: Purchase				
Function Id: F012				
Date Created: 26/8/2024				
Role: User				
No.	**Input Values**	**Expected Results**	**Actual Results**	**Pass/Fail**
1	Click “Purchase” button in shopping list page.	The system will prompt a purchase confirmation dialog.	The system will prompt a purchase confirmation dialog.	Passed.
2	Click “Purchase” button in the purchase confirmation dialog.	The system will prompt a share receipt confirmation dialog.	The system will prompt a share receipt confirmation dialog.	Passed.
3	Click “Share” button in the share receipt confirmation dialog.	The system will prompt the media sharing window.	The system will prompt the media sharing window.	Passed.

Table 6.12 Purchase Testing

6.2.13 [F013] Modify and Delete Shopping List Testing

Modify and Delete Shopping List Test Case				
Use Case: Modify and Delete Shopping List				
Function Id: F013				
Date Created: 26/8/2024				
Role: User				
No.	**Input Values**	**Expected Results**	**Actual Results**	**Pass/Fail**
1	Click “Clear” button in shopping list page.	The system will prompt a delete confirmation dialog.	The system will prompt a delete confirmation dialog.	Passed.
2	Click “Clear” button in the delete confirmation dialog.	The shopping list will be empty.	The shopping list will be empty.	Passed.
3	Click “+” button in shopping list page.	The system will increment the quantity of the item in shopping list.	The system will increment the quantity of the item in shopping list.	Passed.
4	Click “-” button in shopping list page.	The system will decrement the quantity of the item in shopping list.	The system will decrement the quantity of the item in shopping list.	Passed.
5	Item quantity is lesser than 1.	The system will remove the ingredient in the shopping list.	The system will remove the ingredient in the shopping list.	Passed.
6	Item quantity is larger than stock quantity.	The quantity amount will stop incrementing.	The quantity amount will stop incrementing.	Passed.

Table 6.13 Modify and Delete Shopping List Testing

6.2.14 [F014] Edit Profile Picture Testing

Edit Profile Picture Test Case				
Use Case: Edit Profile Picture				
Function Id: F014				
Date Created: 26/8/2024				
Role: User				
No.	**Input Values**	**Expected Results**	**Actual Results**	**Pass/Fail**
1	Click “Change Profile Picture” button.	The system will prompt media selection window to let the user select a picture to set as profile picture.	The system will prompt media selection window to let the user select a picture to set as profile picture.	Passed.

Table 6.14 Edit Profile Picture Testing

6.3 Project Challenges

Implementing the mobile food recipe application presented several challenges, highlighting the steep learning curve associated with working with new technologies and integrating external resources. Although Flutter is known for its ease of learning, adapting to its environment was a significant challenge, especially for someone new to the framework. The limited availability of resources related to integrating the Spoonacular API with Flutter projects further compounded the difficulty, necessitating extensive research and experimentation to navigate effectively.

One significant hurdle was the requirement to implement numerous image widgets, with all images being hardcoded and pulled from the API. This process demanded meticulous attention to detail and rigorous testing to ensure accurate display and functionality within the application. Integrating external Flutter libraries such as flutter_bloc, dio, and persistent_bottom_nav_bar also introduced additional complexity. Understanding the functionality and proper implementation of these libraries into the system required significant time and effort.

Moreover, encountering various errors during the integration process necessitated extensive debugging and troubleshooting. Overcoming these challenges involved thorough research, consultation with documentation and community forums, and hands-on experimentation to ensure seamless integration and optimal performance of the application.

The next challenge is to prevent unauthorized users from gaining access, stealing information which could potentially damage the system, and implementing more user customization options. Ensuring the security of user data and protecting the system from potential threats will require the implementation of robust authentication mechanisms and access controls. Additionally, providing users with a greater degree of customization options will enhance the user experience and make the application more adaptable to individual preferences.

6.4 Objectives Evaluation

Objective 1: To Develop a Recipe Exploration Platform by Integrating a Culinary API

This objective aims to revolutionize the culinary experience by seamlessly integrating grocery management and recipe exploration into a single application. The goal is to streamline the meal preparation process, save time and effort, and reduce food wastage.

Achievement and Evaluation:

The application was successfully developed and integrated with a culinary API. This integration is evident in several features of the application:

- **Browse Recipes:** Users can explore a wide range of recipes, as demonstrated by test case [F005] in section 6.2.5. This feature allows users to view detailed recipe information, including ingredients, preparation time, and nutritional information.
- **Save Recipes and Ingredients:** The application allows users to save recipes and add ingredients directly to their grocery list, as shown in test case [F006] in section 6.2.6. This feature directly addresses the goal of streamlining the meal preparation process.
- **Search Recipes:** Users can search for recipes by name, ingredients, or even videos, as evidenced by test case [F007] in section 6.2.7. This comprehensive search functionality enhances the recipe exploration experience.
- **View and Modify Saved Recipes and Ingredients:** Test case [F008] in section 6.2.8 demonstrates that users can view and modify their saved recipes and ingredients, allowing for efficient grocery list management.

The evaluation of these features through various test cases shows that the application successfully integrates recipe exploration with grocery management, achieving the first objective.

Objective 2: Inspire Users to Explore Wide Ranges of Culinary Art

This objective aims to encourage users to diversify their meals, experiment with new dishes, and embrace a more varied and nutritious diet by providing an extensive recipe database and advanced search features.

Achievement and Evaluation:

The application successfully implements features that inspire culinary exploration:

- **Diverse Recipe Database:** The browse recipes feature (test case [F005], section 6.2.5) allows users to explore a wide range of recipes, encouraging meal diversity.
- **Advanced Search Features:** Test case [F007] in section 6.2.7 demonstrates the application's ability to search recipes by name, ingredients, and even videos. This versatility in search options promotes exploration of new culinary ideas.
- **Categorized Recipes:** The search functionality also includes categorized recipes and a "Most Searched By People" section, as shown in test case [F007]. This feature helps users discover popular and diverse recipe categories.

The successful implementation and positive evaluation of these features indicate that the application effectively inspires users to explore a wide range of culinary arts, achieving the second objective.

Objective 3: Improve User's Shopping Efficiency

This objective focuses on streamlining the shopping process by providing a digital grocery list that can be populated directly from recipes and modified as needed. The goal is to reduce stress associated with grocery shopping and empower users to make more efficient and informed purchasing decisions.

Achievement and Evaluation:

The application includes several features that contribute to improved shopping efficiency:

- **Digital Grocery List:** Users can view and modify their saved recipes and ingredients in a digital grocery list, as demonstrated by test case [F008] in section 6.2.8.
- **Add to Shopping List:** Test case [F011] in section 6.2.11 shows that users can easily add selected ingredients to their shopping list.
- **View Shopping List:** Users can access a dedicated shopping list view, as shown in test case [F009] in section 6.2.9. This feature allows users to see available and unavailable items, improving shopping efficiency.
- **Modify and Delete Shopping List:** Test case [F013] in section 6.2.13 demonstrates that users can easily modify quantities or remove items from their shopping list, allowing for flexible and efficient list management.
- **Purchase Feature:** The application includes a purchase feature (test case [F012], section 6.2.12) that allows users to complete their shopping process within the app.

The successful implementation and positive evaluation of these features through various test cases indicate that the application effectively improves users' shopping efficiency, achieving the third objective.

In conclusion, the objective evaluations demonstrate that all three main objectives of the project have been successfully achieved. The application effectively integrates recipe exploration with grocery management, inspires culinary exploration, and improves shopping efficiency. The comprehensive test cases provided in Chapter 6 serve as evidence of the successful implementation and functionality of these features.

6.5 Concluding Remark

This chapter focuses on evaluating the system to measure its performance against the predefined requirements and goals. The evaluation determines whether the developed system meets the intended objectives and functions as expected. Use case testing is performed to assess the system's functionality thoroughly. Each use case is systematically evaluated and verified to ensure that the system behaves correctly and delivers the anticipated results. This process confirms that the system operates as intended and fulfills its purpose. The test results from the use case testing were highly satisfactory, indicating that the system performed exceptionally well. The actual outcomes closely matched the expected results, demonstrating that the system met the predefined requirements and goals. This successful evaluation validates the system's effectiveness, reliability, and adherence to the specified functionalities.

CHAPTER 7 Conclusion and Recommendation

7.1 Conclusion

The development of the mobile food recipe application has been a successful endeavor, meeting the project objectives and delivering a functional, user-friendly solution for recipe exploration and grocery management. Through the integration of a culinary API, the application offers users a diverse range of recipes, inspiring them to explore various culinary arts and expand their cooking repertoire.

The application's features, such as recipe browsing, saving recipes and ingredients, advanced search functionality, and digital grocery list management, have been thoroughly tested and validated. The satisfactory results of the use case testing demonstrate the system's effectiveness, reliability, and adherence to the specified requirements.

The project has successfully addressed the challenges faced by university students in meal preparation, including disorganized shopping practices and limited culinary exploration. By streamlining the process from recipe selection to ingredient acquisition, the application promotes efficient grocery shopping, reduces food wastage, and encourages healthier, more diverse eating habits.

Throughout the development process, valuable lessons were learned in overcoming technical challenges, integrating external libraries, and ensuring seamless functionality. The project showcases the potential of mobile technology in transforming traditional cooking and grocery management practices.

In conclusion, the mobile food recipe application stands as a testament to the successful application of software development principles, the effective use of the Flutter framework, and the integration of external APIs to deliver a comprehensive solution. The project has laid a solid foundation for future enhancements and serves as a valuable contribution to the culinary and technology domains.

7.2 Recommendation

While the mobile food recipe application has achieved its primary objectives, there are several recommendations for future enhancements and improvements:

- **User Feedback and Reviews:** Implement a user feedback and review system within the application to gather insights from users regarding their experience, favorite features, and areas for improvement. This feedback can guide future development efforts and ensure the application remains user-centric.
- **Personalized Recommendations:** Leverage user behavior data and machine learning algorithms to provide personalized recipe recommendations based on individual preferences, dietary restrictions, and cooking history. This feature would enhance the user experience and foster continued engagement with the application.
- **Social Sharing and Community Features:** Introduce social sharing capabilities, allowing users to share their favorite recipes, cooking tips, and culinary creations with friends and the wider community. Encouraging user-generated content and facilitating interactions among users can create a vibrant and supportive cooking community within the application.
- **Meal Planning and Nutritional Tracking:** Expand the application's functionality to include meal planning features, enabling users to create weekly or monthly meal plans based on their preferences and nutritional goals. Integrate a nutritional tracking system that helps users monitor their calorie intake, macronutrient balance, and overall dietary habits.
- **Partnership with Local Grocery Stores:** Explore partnerships with local grocery stores to provide real-time information on ingredient availability, pricing, and promotions. This integration would enhance the shopping experience for users and offer a more seamless transition from recipe selection to ingredient acquisition.
- **Multilingual Support:** Consider implementing multilingual support to cater to a wider user base. Offering the application in multiple languages can expand its reach and make it more inclusive for users from diverse backgrounds.
- **Continuous Performance Optimization:** Regularly assess and optimize the application's performance, focusing on areas such as loading times, resource

utilization, and battery consumption. Continuous performance improvements will ensure a smooth and efficient user experience.

By implementing these recommendations, the mobile food recipe application can evolve into a more comprehensive, user-centric, and socially engaging platform. Continued development and refinement based on user feedback and emerging technologies will position the application as a leader in the culinary and mobile app market.

REFERENCES

[1] M. Hillyer, "How has technology changed - and changed us - in the past 20 years?," World Economic Forum, [Online]. Available: https://www.weforum.org/agenda/2020/11/heres-how-technology-has-changed-and-changed-us-over-the-past-20-years/. [Accessed 18 Nov 2020].

[2] Jovarie, "Role of Technology in Culinary Arts," Technolufy World's News, [Online]. Available: https://importanceoftechnology.net/role-of-technology-in-culinary-arts/. [Accessed 2 May 2021].

[3] L. K. Ozanne, P. Ballantine and A. McMaster, "Understanding Food Waste Produced by University Students: A Social Practice Approach," MDPI, [Online]. Available: https://www.mdpi.com/2071-1050/14/17/10653. [Accessed 26 August 2022].

[4] R. Alpert, "Meal Preparation Constraints in Emerging Adults in College," BRIDGEWATER STATE UNIVERSITY, [Online]. Available: https://vc.bridgew.edu/cgi/viewcontent.cgi?article=1505&context=undergrad_rev. [Accessed 2021].

[5] C. Lin Phooi, E. A. Azman, R. Ismail, J. A. Shah and E. S. Rou Koay, "Food Waste Behaviour and Awareness of Malaysian," PMC PubMed Central, [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9444424/. [Accessed 29 August 2022].

[6] K. Ahuja, "Benefits of cooking apps in kitchen," ArticlesFACTORY, [Online]. Available: https://www.articlesfactory.com/articles/food/benefits-of-cooking-apps-in-kitchen.html. [Accessed 29 January 2015].

[7] A. Gomstyn, "4 surprising health benefits of a home-cooked meal," aetna, [Online]. Available: https://www.aetna.com/health-guide/health-benefits-of-home-cooking.html#:~:text=1.,nutritious%20and%20contain%20fewer%20calories..

[8] K. T. Hanna, "COPPA (Children's Online Privacy Protection Act)," TechTarget, [Online]. Available: https://www.techtarget.com/searchcio/definition/COPPA-Childrens-Online-Privacy-Protection-Act. [Accessed March 2022].

[9] J. Chen, "Android Operating System (OS): Definition and How It Works," Investopedia, [Online]. Available: https://www.investopedia.com/terms/a/android-operating-system.asp. [Accessed 5 June 2022].

[10] Coopossum, "How Open-Source Is Android Really?," Medium, [Online]. Available: https://medium.com/@coopossum/how-open-source-is-android-8d1815b9a42d. [Accessed 26 November 2022].

[11] A. Turner, "How Many Android Users Are There? Global and US Statistics (2024)," bankmycell, [Online]. Available: https://www.bankmycell.com/blog/how-many-android-users-are-there. [Accessed 13 March 2024].

[12] "Why iOS is not Open-Source like Android?," TheTechTop10, [Online]. Available: https://thephilosophyoftech.com/why-ios-is-not-open-source-like-android/. [Accessed 2 July 2020].

[13] GrowSolutions, "What is Flutter? A Complete Guide to Google's Framework," Medium, [Online]. Available: https://medium.com/@growsolutions/what-is-flutter-a-complete-guide-to-googles-framework-b51ae5e2fbd5. [Accessed 15 September 2023].

[14] G. Andrades, "What Is Flutter Framework? A Guide to Flutter App Testing," ACCELQ, [Online]. Available: https://www.accelq.com/blog/flutter-framework/. [Accessed 3 June 2022].

[15] P. Paterska, "What is React Native And When to Use It For Your App in 2024 (Updated)," ELPassion, [Online]. Available: https://www.elpassion.com/blog/what-is-react-native-and-when-to-use-it. [Accessed 23 February 2024].

[16] M. Budziński, "What Is React Native? Complex Guide for 2023," Netguru, [Online]. Available: https://www.netguru.com/glossary/react-native#what-is-cross-platform-development.

[17] L. Bedosky, "Meal Planning 101: A Complete Beginner's Guide to Meal Prep," [Online]. Available: https://www.everydayhealth.com/diet-nutrition/meal-planning/. [Accessed 28 Mar 2022].

[18] "Epicurious Software," Justuseapp.com, [Online]. Available: https://justuseapp.com/en/app/312101965/epicurious. [Accessed 11 January 2024].

[19] L. Stinson, "How a Food Blog Shaped the Way We Use the Internet," WIRED, [Online]. Available: https://www.wired.com/2015/08/epicurious/. [Accessed 20 August 2015].

[20] similarweb, [Online]. Available: https://www.similarweb.com/website/epicurious.com/#overview.

[21] [Online]. Available: https://apps.apple.com/us/app/epicurious/id312101965.

[22] M. A. Fernandez, RD, S. Descroches, M. Marquis and V. Provencher, "Meal planning as a strategy to support healthy eating," erudit, [Online]. Available: https://www.erudit.org/en/journals/nutrition/2020-v17-n3-nutrition05242/1068830ar.pdf. [Accessed 30 April 2020].

[23] C. India, "10 REASONS WHY COOKPAD APP IS A MUST ON YOUR PHONE," Medium, [Online]. Available: https://blog.cookpad.in/10-reasons-why-cookpad-app-is-a-must-on-your-phone-dabd326d53be. [Accessed 9 June 2017].

[24] Rexan, "Cookpad App Review: The Social Network for Home Chefs," [Online]. Available: https://cellularnews.com/mobile-apps/cookpad-app-review/. [Accessed 13 Jul 2023].

[25] Paul, "CookBook App Review (Here's What I Think)," Meal Matchmaker, [Online]. Available: https://mealmatchmaker.com/cookbook-app-review.

[26] "What is SDLC (Software Development Lifecycle)?," AWS, [Online]. Available: https://aws.amazon.com/what-is/sdlc/#:~:text=The%20software%20development%20lifecycle%20(SDLC,expectations%20during%20production%20and%20beyond..

[27] H. Clark, "The Software Development Life Cycle (SDLC): 7 Phases and 5 Models," [Online]. Available: https://theproductmanager.com/topics/software-development-life-cycle/.

[28] "Software Prototyping Model and Phases," Geeksforgeeks, [Online]. Available: https://www.geeksforgeeks.org/software-prototyping-model-and-phases/. [Accessed 30 March 2023].

[29] "SDLC - Iterative Model," tutorialspoint, [Online]. Available: https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm.

[30] "Android Studio," TechTarget, [Online]. Available: https://www.techtarget.com/searchmobilecomputing/definition/Android-Studio.

[31] P. Biswas, "Introduction to Android Studio," loginradius, [Online]. Available: https://www.loginradius.com/blog/engineering/introduction-to-android-studio/.

[32] M. Heller, "What is Visual Studio Code? Microsoft's extensible code editor," InfoWorld, [Online]. Available: https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html. [Accessed 8 July 2022].

[33] D. Stevenson, "What is Firebase? The complete story, abridged.," Medium, [Online]. Available: https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0. [Accessed 25 September 2018].

[34] "Is Firebase Free? Everything you Need to Know in 2024," Lido, [Online]. Available: https://www.lido.app/firebase/is-firebase-free.

[35] A. Olawale, "Firebase Authentication," Medium, [Online]. Available: https://medium.com/@Adekola_Olawale/firebase-authentication-413626c5234d. [Accessed 17 December 2022].

[36] Duggu, "Understanding Real-time Firebase Database in Android," Medium, [Online]. Available: https://medium.com/@dugguRK/understanding-real-time-firebase-database-in-android-eec9967745c8. [Accessed 28 September 2023].

[37] L. Rebecca, "PROS & CONS of Physical and Digital Books," [Online]. Available: https://becsbookshelf.wordpress.com/2018/09/26/pros-cons-of-physical-and-digital-books/. [Accessed 26 Sept 2018].

[38] A. Laskowski, "Student Dreams of a Meal Planning App to Avoid Waste," BU Today, [Online]. Available: https://www.bu.edu/articles/2022/student-dreams-of-a-meal-planning-app-to-avoid-waste/. [Accessed 20 April 2022].

[39] M. Saltzman, "5 Delicious Reasons to Use Recipe Apps and Websites," [Online]. Available: https://www.aarp.org/home-family/personal-technology/info-2022/recipe-apps.html. [Accessed 31 Jan 2022].

[40] V. Sharma, "How To Make a Meal Planning App like Whisk," [Online]. Available: https://ideausher.com/blog/how-to-make-app-like-whisk/. [Accessed 3 Aug 2023].

[41] Tecno, "Smartphone innovation in the third decade of the 21st century," [Online]. Available: https://www.technologyreview.com/2020/03/05/905500/smartphone-innovation-in-the-third-decade-of-the-21st-century/#:~:text=Thanks%20to%20the%20development%20of%20networks%2C%20technology%2C%20and,music%2C%20and%20play%20games%2C%20among%20many%20other%20fu. [Accessed 5 Mar 2020].

[42] C. Romans, "Why Kitchen Stories Is the Best Cooking App for Learning Delicious Recipes," [Online]. Available: https://www.makeuseof.com/kitchen-stories-best-app-learning-delicious-recipes/. [Accessed 16 Jul 2022].

[43] P. Bhanderi, "Flutter’s Benefits and Drawbacks," Medium, [Online]. Available: https://medium.com/@parthbhanderi01/flutters-benefits-and-drawbacks-b268c1fe0f7f. [Accessed July 10 2023].

[44] G. Mastorakis, I. Kopanakis, J. Makridis, C. Chroni, K. Synani, K. Lasaridi, K. Abeliotis and I. Louloudakis, "Managing Household Food Waste with the FoodSaveShare Mobile Application," MDPI, [Online]. Available: https://www.mdpi.com/2071-1050/16/7/2800. [Accessed 27 March 2024].

[45] "Health benefits of meal planning," Corewell Health, [Online]. Available: https://www.beaumont.org/health-wellness/blogs/health-benefits-of-meal-planning.

APPENDIX

Appendix A: Biweekly Report

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: Y3S2	**Study week no.: Week 2**
Student Name & ID: Vickern Tan 21ACB05679	
Supervisor: Ms. Tan Lyk Yin	
Project Title: An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning	

1. WORK DONE

- Review and refresh memories on what I've done on IIPSPW.
- Installed Android Studio, Visual Studio.

2. WORK TO BE DONE

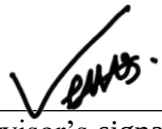
- Figure out how to setup Flutter in my system.
- Figure out how Spoonacular API, and Flutter framework works.
- Design application wireframe to have a first view of application UI.

3. PROBLEMS ENCOUNTERED

- No problem encountered in this week.

4. SELF EVALUATION OF THE PROGRESS

- Self-assigned tasks are completed within expected timeframe.



Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: Y3S2	**Study week no.: Week 4**
Student Name & ID: Vickern Tan 21ACB05679	
Supervisor: Ms. Tan Lyk Yin	
Project Title: An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning	

1. WORK DONE

- Installed flutter, and extensions.
- Completed app UI design wireframe.

2. WORK TO BE DONE

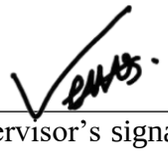
- Start implementing project UI (Welcome ,Login & Register page).
- Implement login, and sign-up function.

3. PROBLEMS ENCOUNTERED

- Flutter is too new to me, have to spend more time learning on types of widgets.

4. SELF EVALUATION OF THE PROGRESS

- Self-assigned tasks are completed within expected timeframe.



Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: Y3S2	**Study week no.: Week 6**
Student Name & ID: Vickern Tan 21ACB05679	
Supervisor: Ms. Tan Lyk Yin	
Project Title: An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning	

1. WORK DONE

- Added Welcome, Login, and Sign-up page UI.
- Implemented login and sign-up function including input validation and database linking.

2. WORK TO BE DONE

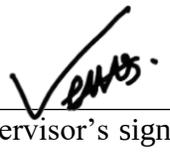
- Integrate Spoonacular API.
- Retrieve information from Spoonacular API.

3. PROBLEMS ENCOUNTERED

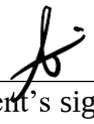
- Unable to retrieve Spoonacular API data, json converter library needed.

4. SELF EVALUATION OF THE PROGRESS

- Self-assigned tasks are completed within expected timeframe.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: Y3S2	**Study week no.: Week 8**
Student Name & ID: Vickern Tan 21ACB05679	
Supervisor: Ms. Tan Lyk Yin	
Project Title: An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning	

1. WORK DONE

- Integrated Spoonacular API.
- Able to retrieve data from Spoonacular API.

2. WORK TO BE DONE

- Home page (UI design, random recipes calling from API)
- Search page (UI design, recipe data extract)
- Food recipe information page

3. PROBLEMS ENCOUNTERED

- Lack of online resources regarding to relevant API and Flutter.
- Encountered many errors when trying to retrieve multiple data from the API.

4. SELF EVALUATION OF THE PROGRESS

- Self-assigned tasks are completed in expected timeframe.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: Y3S2	**Study week no.: Week10**
Student Name & ID: Vickern Tan 21ACB05679	
Supervisor: Ms. Tan Lyk Yin	
Project Title: An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning	

1. WORK DONE

- Home page (UI design, random recipes calling from API).
- Search page (UI design, search function partially done).

2. WORK TO BE DONE

- Search page
- Food recipe information page
- More page
- Profile page

3. PROBLEMS ENCOUNTERED

- Unable to make the search function working dynamically, errors occurred while testing.

4. SELF EVALUATION OF THE PROGRESS

- Self-assigned tasks are completed in expected time frame.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I)

Trimester, Year: Y3S2	**Study week no.: Week 12**
Student Name & ID: Vickern Tan 21ACB05679	
Supervisor: Ms. Tan Lyk Yin	
Project Title: An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning	

1. WORK DONE

- Search page
- Food recipe information display
- More page
- Profile page

2. WORK TO BE DONE

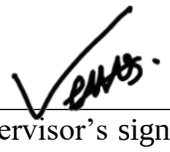
- Improve UI design, user-interaction.
- FYP 1 report.

3. PROBLEMS ENCOUNTERED

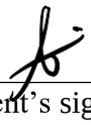
- No problems encountered for this week.

4. SELF EVALUATION OF THE PROGRESS

- Self-assigned tasks are completed in expected time frame.



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: Week 2**
Student Name & ID: Vickern Tan 21ACB05679	
Supervisor: Ms. Tan Lyk Yin	
Project Title: An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning	

1. WORK DONE

2. WORK TO BE DONE

- System implementation.

3. PROBLEMS ENCOUNTERED

- No problems encountered for this week.

4. SELF EVALUATION OF THE PROGRESS

- Self-assigned tasks are completed in expected time frame.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: Week 4**
Student Name & ID: Vickern Tan 21ACB05679	
Supervisor: Ms. Tan Lyk Yin	
Project Title: An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning	

1. WORK DONE

- Some parts of features implementation are done.

2. WORK TO BE DONE

- Finish the rest of features implementation.

3. PROBLEMS ENCOUNTERED

- No problems encountered for this week.

4. SELF EVALUATION OF THE PROGRESS

- Self-assigned tasks are completed in expected time frame.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: Week 6**
Student Name & ID: Vickern Tan 21ACB05679	
Supervisor: Ms. Tan Lyk Yin	
Project Title: An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning	

1. WORK DONE

- The system implementation is completed.

2. WORK TO BE DONE

- System debugging.

3. PROBLEMS ENCOUNTERED

- No problems encountered for this week.

4. SELF EVALUATION OF THE PROGRESS

- Self-assigned tasks are completed in expected time frame.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: Week 8**
Student Name & ID: Vickern Tan 21ACB05679	
Supervisor: Ms. Tan Lyk Yin	
Project Title: An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning	

1. WORK DONE

- Chapter 3 and Chapter 4 report writing is completed.

2. WORK TO BE DONE

- Write Chapters 5 of the report.

3. PROBLEMS ENCOUNTERED

- No problems encountered for this week.

4. SELF EVALUATION OF THE PROGRESS

- Self-assigned tasks are completed in expected time frame.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: Week 10**
Student Name & ID: Vickern Tan 21ACB05679	
Supervisor: Ms. Tan Lyk Yin	
Project Title: An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning	

1. WORK DONE

- Chapter 5 report writing is completed.

2. WORK TO BE DONE

- Write test cases to test the system.

3. PROBLEMS ENCOUNTERED

- No problems encountered for this week.

4. SELF EVALUATION OF THE PROGRESS

- Self-assigned tasks are completed in expected time frame.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: Week 12**
Student Name & ID: Vickern Tan 21ACB05679	
Supervisor: Ms. Tan Lyk Yin	
Project Title: An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning	

1. WORK DONE

- The system testing is completing.
- Improved UI design, user-interaction.

2. WORK TO BE DONE

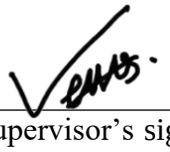
- Finish the report.
- Double check all fields.

3. PROBLEMS ENCOUNTERED

- No problems encountered for this week.

4. SELF EVALUATION OF THE PROGRESS

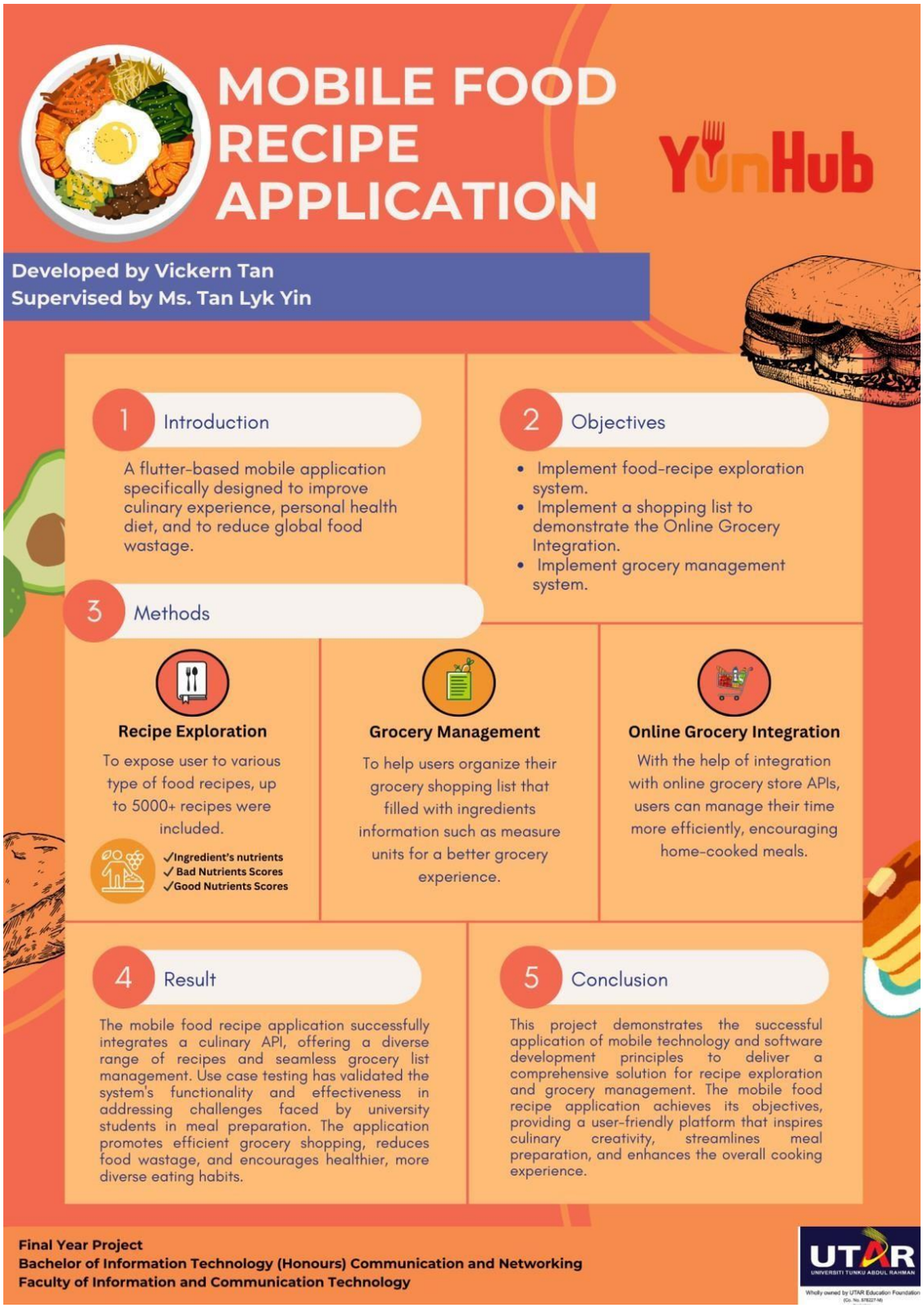
- Self-assigned tasks are completed in expected time frame.



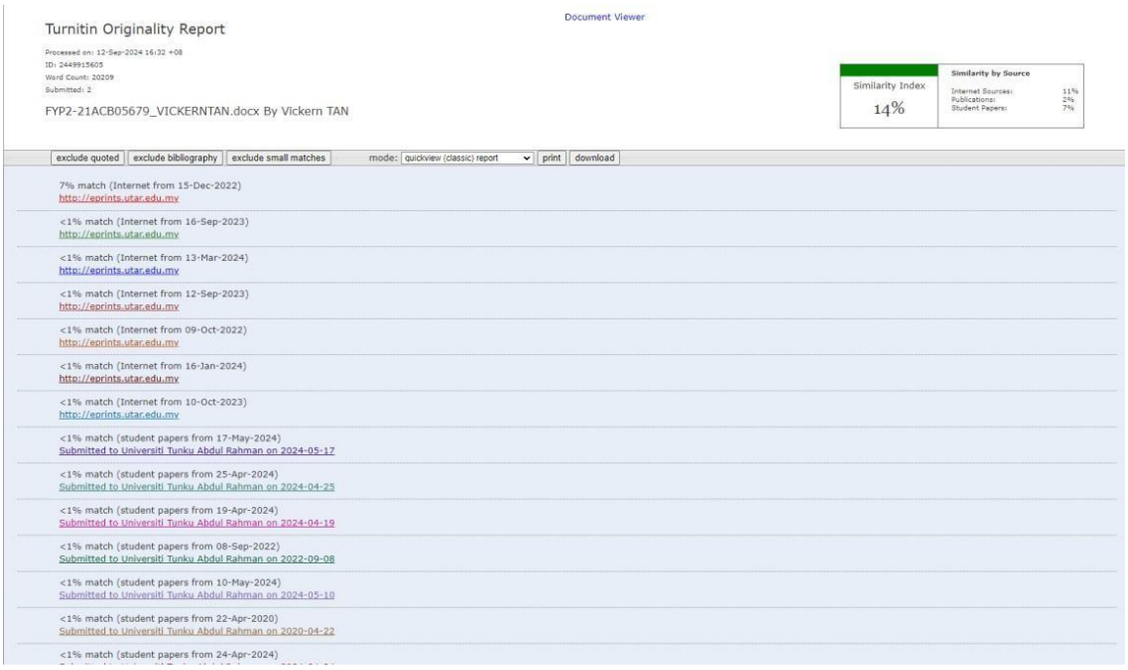
Supervisor's signature

Student's signature

Appendix B: Poster



PLAGIARISM CHECK RESULT



Turnitin Originality Report

Processed on: 12-Sep-2024 16:32 +08
ID: 2449915605
Word Count: 20209
Submitted: 2

FYP2-21ACB05679_VICKERNTAN.docx By Vickern TAN

Document Viewer

Similarity Index 14%

Similarity by Source
Internet Sources: 11%
Publications: 2%
Student Papers: 7%

exclude quoted | exclude bibliography | exclude small matches | mode: quickview (classic) report | print | download

7% match (Internet from 15-Dec-2022)
http://eprints.utar.edu.my

<1% match (Internet from 16-Sep-2023)
http://eprints.utar.edu.my

<1% match (Internet from 13-Mar-2024)
http://eprints.utar.edu.my

<1% match (Internet from 12-Sep-2023)
http://eprints.utar.edu.my

<1% match (Internet from 09-Oct-2022)
http://eprints.utar.edu.my

<1% match (Internet from 16-Jan-2024)
http://eprints.utar.edu.my

<1% match (Internet from 10-Oct-2023)
http://eprints.utar.edu.my

<1% match (student papers from 17-May-2024)
Submitted to Universiti Tunku Abdul Rahman on 2024-05-17

<1% match (student papers from 25-Apr-2024)
Submitted to Universiti Tunku Abdul Rahman on 2024-04-25

<1% match (student papers from 19-Apr-2024)
Submitted to Universiti Tunku Abdul Rahman on 2024-04-19

<1% match (student papers from 08-Sep-2022)
Submitted to Universiti Tunku Abdul Rahman on 2022-09-08

<1% match (student papers from 10-May-2024)
Submitted to Universiti Tunku Abdul Rahman on 2024-05-10

<1% match (student papers from 22-Apr-2020)
Submitted to Universiti Tunku Abdul Rahman on 2020-04-22

<1% match (student papers from 24-Apr-2024)

Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Vickern Tan
ID Number(s)	21ACB05679
Programme / Course	Bachelor of Information Technology (Honours) Communication and Networking
Title of Final Year Project	An Integrated Mobile Application for Personalized Grocery Management, Recipe Exploration, and Meal Planning

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: 14 % **Similarity by source** Internet Sources: 11 % Publications: 2 % Student Papers: 7 %	
Number of individual sources listed of more than 3% similarity: 1	The sources with more than 3% similarity are from the UTAR repository (using the same report template).
Parameters of originality required, and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Ms. Tan Lyk Yin

Signature of Co-Supervisor

Name:

PLAGIARARISM CHECK RESULT

Date:	Date:
12/9/2024	

FYP 2 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP1 THESIS SUBMISSION

Student ID	21ACB05679
Student Name	Vickern Tan
Supervisor Name	Ms. Tan Lyk Yin

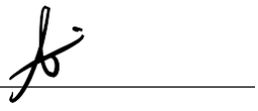
TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed form of the Declaration of Originality
√	Acknowledgment
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Poster

FYP 1 CHECKLIST

√	Signed Turnitin Report (Plagiarism Check Result – Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.



(Signature of Student)

Date: 12/9/2024