Cooking Assistant with Nutritional Tracking App

By

Ngang Chi Khim

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology
(Kampar Campus)

June 2025

COPYRIGHT STATEMENT

This Final Year Project proposal is submitted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science (Honours) at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project proposal represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project proposal may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Ts Dr Lim Seng Poh who supervised my progress and gave me a lot of advice and guidelines. This marks my first step in developing a mobile application, and his support has been instrumental in helping me build confidence and grow in this field.

Finally, I would also like to thank my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

This project will develop a mobile application to address the need for accessible nutritional tracking and convenient cooking guidance. This project integrates computer vision and machine learning techniques into the mobile app. The problem lies in the complexity of maintaining a balanced diet while juggling busy lifestyles, with the challenge of accurately tracking nutritional intake. The technique and methodology adopted include the computer vision and convolutional neural network (CNNs) for ingredient recognition from user captured images. Moreover, creating the interface using Figma. In conclusion, this project demonstrates how artificial intelligence driven solutions may simplify meal preparation and encourage healthier eating habits. The app's ability to integrate nutritional tracking, ingredient recognition, ingredient management and voice-assisted cooking advice onto a single platform is a creative way to tackle contemporary dietary issues.

Area of Study: Mobile Application Development

Keywords: Mobile Application, Computer Vision, Convolutional Neural Network (CNN), Ingredient Recognition, Machine Learning, user-friendly application

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

This chapter presents the background and motivation of the project, which is to integrate the fields of data integration, database management, and information retrieval.

Cooking assistant which is an accommodate tool designed to guide users to cook their dishes in the kitchen. It is like playing the role of a master chef and guiding novices one-on-one on how to cook. The cooking assistant can propose recipes based on the ingredients and guide people through the steps of the generated recipes to achieve cooking, such as the order of inputting ingredients, the quantity of seasonings, the temperature heat needs and the cooking time. Recipe healthiness is evaluated by adhering to dietary guidelines and the nutrition detail label in the nutrition analysis [1]. The nutritional analysis involves evaluating the cooking recipe to determine their nutritional information such as minerals, vitamins, carbohydrates, and much more. A nutritional analysis tracker is basically a pursuing tool for users to investigate the nutritional value and details of their recipes. Users can use nutritional analysis content to evaluate whether the current cooking recipes are satisfactory and can meet the standards of pursuing a healthy life.

Diet is essential to living a healthy existence. Home-cooked meals are prioritized to support the objective of a healthy diet because they enable people to regulate how much salt and oil they consume, which is essential for maintaining good health and preventing diseases linked to a certain lifestyle [2]. Nowadays, there is a growing demand to swiftly and effectively cook well-balanced meals. However, because most of them are unfamiliar with culinary methods, novices frequently find it difficult to cook effectively, which results in prolonged preparation periods. This will lead to a situation where people will rely more on eating at fast food restaurants or buying processed frozen foods, even though they are much more costly [3]. Eating fast food or eating outside has health risks because there is no guarantee on food control. Only cooking by ourselves will improve the security of our diet, because freely adjustment on the quality and quality control for seasonings and ingredients.

Tools that accommodate specific dietary requirements and preferences are in greater demand as the trend toward personalized health and wellbeing keeps growing. Today's customers look for products that may satisfy a range of dietary requirements, including those caused by allergies, intolerances, and lifestyle decisions like veganism or ketogenic diets. The trend

toward personalized nutrition emphasizes how important it is to have a flexible culinary assistant that can adjust recipe suggestions to fit individual dietary needs and health objectives in addition to offering general recipe guidance [4]. The kitchen assistant can help customers maintain a balanced, fulfilling diet that fits their specific nutritional demands by providing personalized meal plans and ingredient substitutions. Furthermore, the cooking assistant can be extremely helpful in promoting long-term good eating habits by informing users about the nutritional value of various foods and assisting them in making educated dietary decisions.

As information communication technologies (ICTs) have been evolving and developing to offer sophisticated communications and computing competencies form distant settings [5]. The conjunction of these aspects of modernization of information technology (IT) and globalization presents an opportunity to increase health awareness using minimal resources [6]. The world of home cooking is rapidly changing with the combination of cutting-edge information technology and smart kitchen applications. Most image-based cooking recipe systems on the market rely on machine learning (ML) techniques, more specifically deep learning (DL), to identify food types, estimate quantities, and predict the nutritional value of a given recipe [7]. Cooking assistance apps are capitalizing on this trend, introducing features such as using computer vision to capture ingredients to create recipes. Computer vision is used inside the application, and it is a field of artificial intelligence (AI) that uses machine learning and neural networks to teach computers and systems to derive meaningful information from digital images [8]. These technology makes meal preparation easier by letting customers snap photos of available ingredients and get instant recipe suggestions.

Summary, the goal of the project is to create a cutting-edge digital tool that makes meal preparation easier, increases nutritional awareness, and encourages effective ingredient management, all of which help people live healthier lives. The initiative aims to address common issues such as time constraints, food waste and dietary information to promote sustainable cooking practices and personalized nutrition. The project aims to increase the accessibility, convenience and enjoyment of healthy eating for the general population through the integration of cutting-edge technologies.

1.1 Problem Statement and Motivation

In the fast-paced world of today, it can be difficult to keep up a balanced diet and prepare wholesome meals at home. A lot of people find it difficult to organize meals that satisfy their

nutritional needs and don't have a trustworthy way to monitor their intake of nutrients. Meal planning and nutrition tracking are two traditional practices that are frequently done by hand and take a lot of time and work to maintain accuracy and consistency. In addition, manually maintaining ingredient inventories may result in food waste, lost possibilities for well-balanced meals, and higher expenses. A complete solution that streamlines these procedures could lead to better eating practices, less food waste, and an improvement in general health and wellbeing.

Inadequate Support for Time-Constrained Meal Preparation

In today's hectic environment, many people find it difficult to find the time to make nutritious home-cooked meals. Existing meal preparation apps often fall short in addressing this issue effectively. While they may provide recipes or grocery lists, they often lack features that streamline the cooking process or adapt to users' tight schedules. Efficient work schedules, familial obligations, and several other commitments frequently impede the time allocated for deliberate cooking. Because of this, consumers frequently choose quick and easy options like processed frozen foods or fast food, which are frequently heavy in sodium, sugar, and harmful fats. Convenience food addiction can result in several health issues, such as diabetes, cardiovascular high blood pressure, and obesity [9]. Tools that make meal preparation easier are obviously needed so that people may cook wholesome meals at home more quickly and effectively.

Increasing complexity of food health requirement

Dietary requirements are becoming increasingly complex as many people follow certain diets due to health issues, allergies or lifestyle decisions, existing cooking and nutrition apps often fall short of meeting these demands. Dietary requirements and preferences are becoming increasingly complex and the existing app meeting so many needs are difficult. Personalized nutrition solutions that provide customized meal plans and recipes are in high demand. Cooking Assistant apps can produce personalized meals based on certain dietary requirements, such as low-carb, gluten-free or vegan options. This can greatly improve the user's ability to maintain a healthy, balanced diet.

Lack of nutritional knowledge

Many people have ignored nutrition, which can greatly impact their ability to make informed food decisions. A balanced diet requires understanding the proportions of macronutrients such as protein, fat, and carbohydrates, as well as the importance of micronutrients such as vitamins

and minerals. This ignorance about nutrition often results in people eating foods that are either low in important nutrients or high in calories but low in nutritional value. Most of the existing applications have issues of lacking nutritional information which will then cause the user to not know how different foods affect their energy levels, weight, and overall health, or how to portion foods correctly.

Difficulty in managing ingredient inventories causes food waste

Managing a well-stocked kitchen can be difficult, especially for people who don't usually plan their meals or lead hectic lives. Frequently, people purchase groceries without really knowing what they will use or what they already have, which results in overspending and eventually spoiled food. This pattern of overspending on food and allowing it to spoil is not only costly financially but also detrimental to the environment. It makes the issue of food waste worse, which has serious ramifications for both environmental sustainability and global food security. Effective management of ingredient inventories is essential to minimize waste, save money, and ensure that food is utilized efficiently.

1.2 Objectives

The project will develop a comprehensive cooking assistant app that combines intelligent ingredient management, and detailed nutritional tracking to improve the user's experience in the kitchen. The app aims to encourage better eating habits, and make meal plan easier by incorporating these features. The objectives of the project are stated as follows:

- To develop a mobile application that sorts and identifies ingredients using image recognition technology.

  Capture and identify the ingredient image then store them in the digital inventory system for effective kitchen management.
- To implement a personalized recipe system that generates their meal plan based on their nutritional needs.

  Detailed nutritional analysis for each generated dish by allowing users to monitor the intake of the calories, macronutrients, and micronutrients needed and schedule their daily or weekly meals.

1.3 Project Scope and Direction

The project encompasses the development and integration of various software components to create a portable and smart cooking assistant mobile application. The mobile application will

serve as a multifunctional tool to provide ease and assist users in identifying ingredients, managing ingredient's inventory, planning meals, and tracking the nutritional information with precision.

The core components include:

- Image Recognition Technology
  It is used to identify ingredients from photos and save to inventory management.
- Nutritional Analysis Module
  Nutritional details are provided and tracking the consumption, such as calorie counts and macronutrient distribution.
- Daily Meal Planning
  The daily meal planning is based on the nutritional goals of the user.

1.4 Contributions

By utilizing technology to streamline meal preparation, improve nutritional literacy, and maximize food utilization, the project supports the promotion of sustainable practices and healthier eating habits. This application aims to provide a number of creative solutions by tackling common issues people have with controlling their diet and cooking schedules. The Cooking Assistant with Nutritional Tracking App has made the following significant contributions.

Reduction of food waste

This project's capacity to decrease food waste through more intelligent ingredient management is one of its main contributions. The application reduces the possibility of supplies going bad or getting lost in the cupboard by letting users keep track of the components they have on hand and making recipe suggestions based on these items.

Nutritional education

The app aims to provide consumers with comprehensive information on the nutritional content of different foods and meals. The application informs users about important nutrients, portion sizes, and the effects of various foods on their general health through individualized feedback.

Efficient ingredients management

The app's ability to handle ingredients effectively is another important feature that aids users in keeping their kitchens tidy. The inventory management system built into the app, users can keep track of their supplies and get reminders for expiration dates.

1.5 Report Organization

The report described as follows:

Chapter 2 focused on the literature review of the previous work and existing similar system. This was to generate an idea to develop the requirement of the user for the proposed system by identifying the weaknesses and adapting the strengths.

Chapter 3 discussed the methodology of the proposed system. This is the main of this chapter which covers the method and system design. It contains visual representations such as system architecture, use case, and activity diagram which describe the functional flow and interaction within the application.

Chapter 4 outlines the technical framework of the developed application and provides a system block diagram, detailed specifications for each software module, and an explanation of the interface and data structure design. It also highlights the interaction between these components in implementing the core functionality, including account management, transaction monitoring, and budget planning.

Chapter 5 focuses on the actual implementation of the system. It describes the development environment, the software tools and techniques used, the setup steps, and the main implementation strategies. User interface screenshots are provided to illustrate the development process for each feature, and implementation challenges and solutions are discussed.

Chapter 6 evaluates the performance and reliability of the application. It details the testing methodology employed, presents test cases and results, and discusses how effectively the system achieves its objectives. It also reviews the development experience, identifies key technical challenges, and assesses how well the implemented functionality aligns with user needs.

Chapter 7 was the conclusion for the whole proposed system. This section included a summary of the proposed system and suggestions on the future works.

CHAPTER 2

Literature Review

2.1 Existing mobile applications

The literature review discusses five mobile applications designed to enhance the cooking experience: Smart Kitchen Assistant, CookAI, PixFood, SideChef, and Supercook. These apps leverage various technologies to assist users in meal planning, recipe discovery, and kitchen management. Smart Kitchen Assistant uses AI for ingredient recognition and inventory management, while CookAI offers voice-activated recipe suggestions and multilingual support. PixFood employs image recognition to identify ingredients and suggest recipes, complete with video instructions. SideChef provides comprehensive cooking experience with step-by-step guides, meal planning, and integrated grocery shopping. Lastly, Supercook focuses on reducing food waste by generating recipes based on available ingredients. Each app offers unique features, from AI-powered suggestions to visual recognition, catering to different user needs and preferences in the kitchen.

2.1.1 Smart Kitchen Assistant

The Smart Kitchen Assistant is an advanced kitchen companion application that leverages artificial intelligence (AI) to enhance the cooking experience by simplifying kitchen management and recipe planning [10]. It is designed to transform the way users interact with their kitchen, making it easier to manage groceries, reduce food waste, and cook meals with confidence.

The Smart Kitchen Assistant's capacity to identify ingredients through picture capture is one of its most notable capabilities. Users may take pictures of their groceries, and the app will recognize and classify the goods based on AI techniques. This functionality adds recognized ingredients to an app's digital "Refrigerator" inventory automatically, making inventory management precise and efficient. In order to maintain the accuracy of the inventory and account for any mistakes in recognition or user preferences, users can also manually amend records.

Figure 2.1.1.1 User Interface of Ingredient Recognition [10]

The figure 2.1.1.1 state that the inventory system is a central component of the Smart Kitchen Assistant, offering more than just a list of ingredients. It includes functionality for managing expiration dates. The app notifies users of ingredients that are nearing their expiration date, prompting them to use these items in their cooking before they spoil. This proactive approach helps users maximize the freshness of their groceries and minimizes the amount of food that goes to waste.

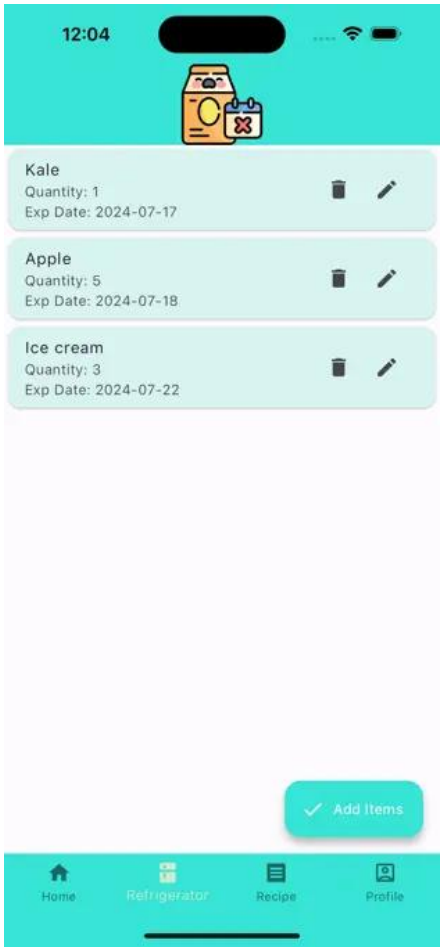


Figure 2.1.1.2 User Interface of Ingredient Management [10]

In addition to inventory management in the figure 2.1.1.2 show the Smart Kitchen Assistant enhances the cooking experience by providing customized recipe suggestions. Users can select ingredients from their inventory that they want to use, and the app generates recipes based on these choices. This feature not only fosters creativity in the kitchen but also ensures that users can make the most of what they have on hand. The generated recipes come with step-by-step text instructions, guiding users through the cooking process with ease and precision.

Figure 2.1.1.3 User Interface of Recipe Instruction [10]

2.1.2 CookAI

CookAI is an innovative personal gourmet guide that leverages the power of Artificial Intelligence (AI) to transform the culinary experience for its users [11]. This application aims to provide an intuitive and personalized cooking experience by utilizing voice-activated features and a diverse recipe database that spans global cuisines.

One of the standout features of CookAI is its voice-activated personalized recipe suggestion system. Users can simply speak to CookAI to receive tailored recipe recommendations. This hands-free approach makes it convenient for users to discover new recipes while they are engaged in other kitchen tasks, providing a seamless cooking experience.

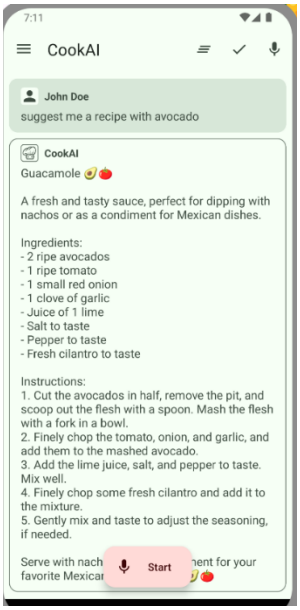


Figure 2.1.2.1 User Interface of voice generated recipe [11]

Another key feature of CookAI is the ability for users to save and organize their favorite recipes in a personalized recipe box. This function allows users to easily access their preferred recipes at any time, making it simple to revisit favorite dishes or plan meals based on previous successful experiences. The personalized recipe box is designed to be user-friendly, enabling quick navigation and easy management of saved recipes.

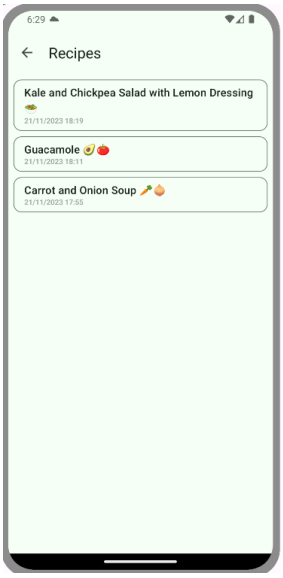


Figure 2.1.2.2 User Interface of Saved Recipe [11]

In order to reach a worldwide audience, CookAI also provides a multilingual experience. A completely worldwide culinary experience is made possible by the ability for users to create a recipe in many languages. This function is especially helpful for people who want to sample

and discover a variety of flavors from around the globe. CookAI promotes a broader understanding of international cuisines and encourages users to try out novel and intriguing foods from many cultures by offering recipes using multiple languages.

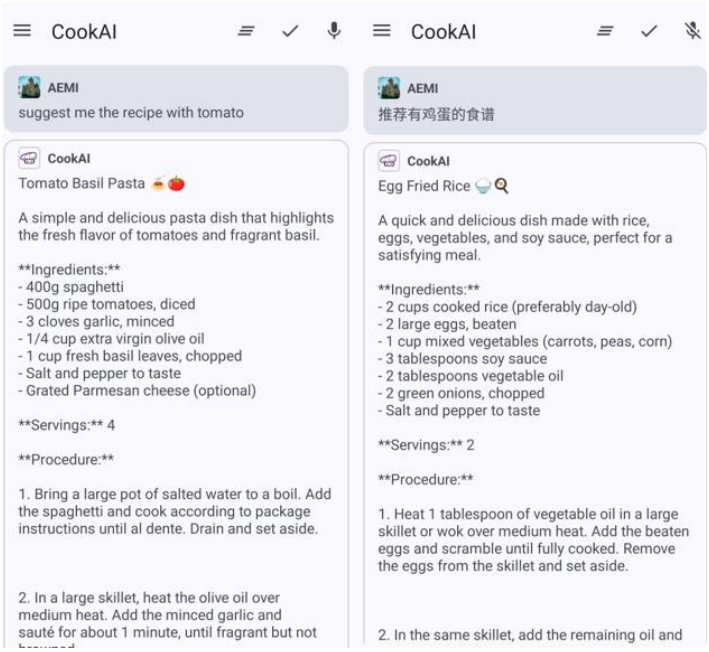


Figure 2.1.2.3 User Interface of multilingual voice support

2.1.3 PixFood

PixFood is a smartphone app that uses image recognition technology to revolutionize how users find and prepare recipes [12]. PixFood's sophisticated picture recognition algorithms make it possible for users to recognize items and get customized recipe recommendations, which makes meal preparation creative and easy.

The primary functionality of PixFood revolves around its visual recognition capabilities. Users can simply take a picture of an ingredient, such as a vegetable, fruit, or pantry item, and the app's sophisticated algorithms quickly identify the item. This feature is particularly useful for those who might be unsure about an ingredient or looking for inspiration on how to use it. The ability to visually recognize ingredients streamlines the cooking process by eliminating the need for manual searching or browsing through recipes, providing a user-friendly and efficient way to explore new culinary possibilities.

Figure 2.1.3.1 User Interface of capture ingredient image [12]

Once an ingredient is identified, PixFood produces a list of recipes that use the found ingredient. This rapid recipe suggestion tools provide ease for user to choose what to prepare depending on what they already have in the kitchen, encouraging creativity and cutting down on food waste. Users can include dietary preferences and allergy information for further customize the recipe results that the app generates, guaranteeing that the dishes are not only quick to prepare but also safe and appropriate for specific dietary requirements. With this degree of personalization, users can adjust their meals to fit particular dietary requirements or health objectives such as low-carb or vegan diets.

In addition to providing recipe lists, PixFood enhances the user experience with step-by-step recipe in videos format. After selecting a recipe, users are guided through the cooking process with detailed video instructions, which can be especially helpful for visual learners or those who prefer video content over text. These videos ensure that users can follow along easily, increasing the likelihood of successfully preparing the chosen dish and enhancing their overall cooking skills.

Figure 2.1.3.2 User Interface of recipe video

2.1.4 SideChef

SideChef is a comprehensive cooking application designed to provide robust and versatile cooking experience, catering to a wide range of users from novices to experienced chefs [13]. The app offers a variety of functionalities to assist users in the kitchen, aiming to simplify the cooking process and inspire culinary creativity.

A standout feature of SideChef is its vast selection of recipes, which are accompanied by video and audio guides, as well as detailed step-by-step instructions. The inclusion of step-by-step instructions is particularly beneficial for beginners, as it helps simplify complex cooking techniques and ensure a smooth learning curve.

Figure 2.1.4.1 User Interface of Step-by-step cooking video instruction

SideChef also offers a meal planning feature, enabling users to create personalized meal plans that can extend over multiple days or weeks. This function allows users to plan their meals in advance, reducing the daily decision-making burden and helping to streamline grocery shopping and meal preparation. This is especially useful for those who wish to maintain a consistent meal schedule or follow a specific diet. Additionally, the app provides daily recipe inspirations through its "Daily Inspiration" feature, offering personalized suggestions that help users discover new dishes. For those looking to utilize ingredients they already have, SideChef includes a search function that allows users to search for recipes based on specific ingredients.

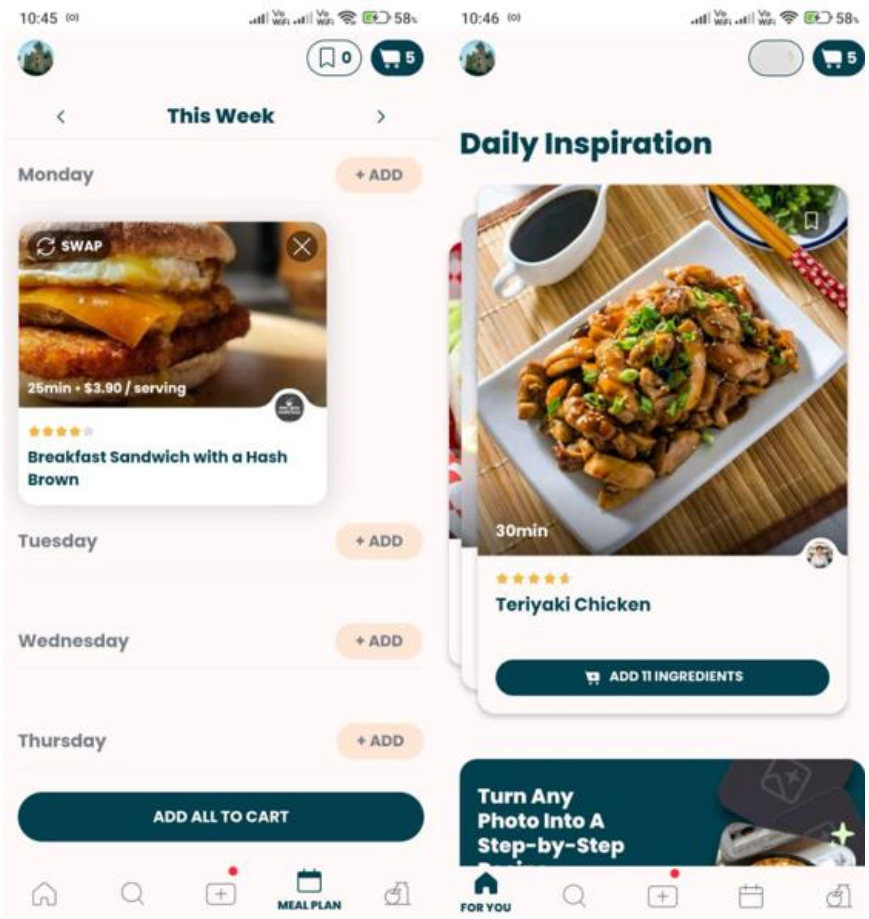


Figure 2.1.4.2 User Interface of Weekly Plan and Daily Recommendation Recipe

One of the unique aspects of SideChef is its integration with grocery shopping. Users can easily create a grocery list based on their selected recipes and even shop for ingredients directly within the app, making the process of meal planning and preparation more seamless. This feature enhances the overall user experience by bridging the gap between recipe selection and ingredient procurement.

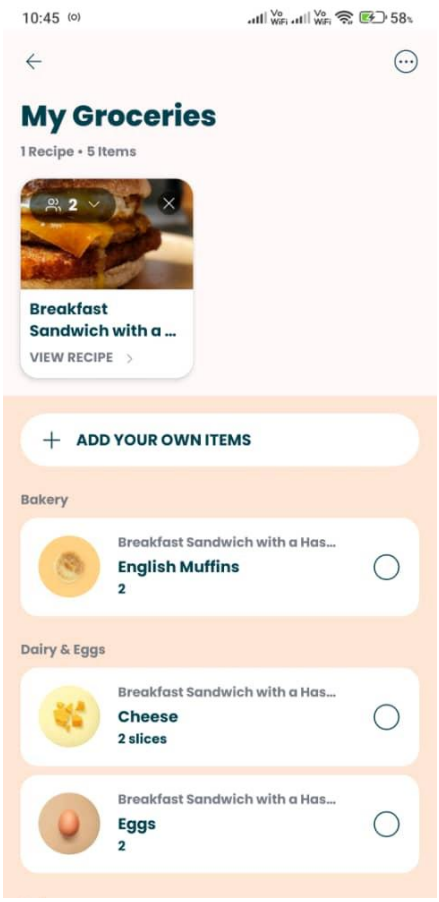


Figure 2.1.4.3 User Interface of grocery shopping list

2.1.5 Supercook

Supercook is a cooking application designed to help users optimize the use of ingredients they already have at home, significantly reducing food waste and maximizing resource utilization [14]. The app's primary feature is its ability to generate recipes based on a list of ingredients input by the user, making it particularly useful for those who wish to avoid unnecessary grocery shopping and make the most out of their existing pantry items.

The process begins with users manually entering the ingredients they currently have on hand into the app. Supercook then uses this information to search for any type of food is available and provide a variety of recipes that can be made using those ingredients.

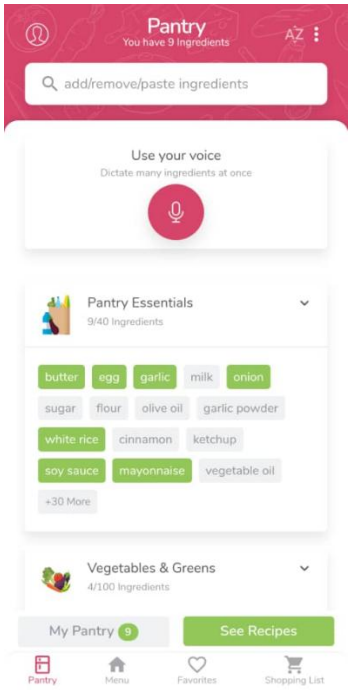


Figure 2.1.5.1 User Interface of search recipe by tag

The app also allows users to save their preferred recipes to a favorites list for future reference, providing easy access to go-to dishes that suit their taste and available ingredients.

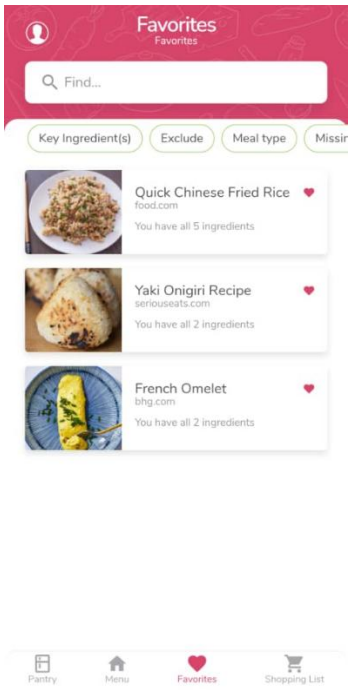


Figure 2.1.5.2 User Interface of Saved Favorite Recipe

Supercook's focus on minimizing food waste and optimizing existing resources positions it as a highly practical application, especially for individuals or families looking to cut down on

food expenses and make the most of their groceries. It's also a valuable tool for those living in areas where frequent grocery shopping may not be feasible or for users who wish to plan meals around what they already have rather than making new purchases.

2.2 Limitation of the previous mobile application

2.2.1 Smart Kitchen Assistant

Despite these innovative features, the Smart Kitchen Assistant focuses primarily on inventory management and basic recipe generation. While it is highly effective for users looking to manage their kitchen stock and reduce food waste, it lacks more advanced capabilities such as detailed nutritional analysis, personalized meal planning based on dietary needs. This gap suggests room for improvement, especially for users seeking a more comprehensive solution that includes health-focused features or more personalized culinary guidance. Besides, the application does not contain a picture or video tutorial for review. This situation will cause users have no confidence on what they cook.

2.2.2 CookAI

CookAI is primarily focused on personalized recipe suggestions and basic culinary management. Although it is excellent at providing user-specific, voice-activated recipes, it is devoid of thorough ingredient management and nutritional information. These shortcomings point to potential areas for growth, especially for consumers looking for more comprehensive cooking aids that balance practicality with dietary and health concerns.

As an example, it emphasizes recipes rather than nutritional analysis or sophisticated meal planning tools. Consideration of nutritional analysis and meal planning tools will allow future growth to better serve the needs of consumers seeking more comprehensive cooking and meal management solutions.

2.2.3 PixFood

PixFood primarily focuses on these two areas of ingredient identification and recipe suggestion and lacks broader functionalities such as comprehensive meal planning, nutritional analysis. The app's strengths lie in its ability to quickly recognize ingredients and suggest recipes, but there is no contains the nutrition management tool.

2.2.4 SideChef

Despite its many strengths, SideChef has some limitations. The app lacks a detailed nutritional tracking feature, which limits its appeal for users who require specific nutritional information or guidance, such as those managing dietary conditions or adhering to specific health goals. Additionally, SideChef does not offer a robust inventory management system. Users must manually track their ingredients evertimes, which can be cumbersome and lead to inefficiencies in meal planning. This absence of automated inventory management might hinder users who rely on precise ingredient tracking for effective kitchen management and meal planning.

2.2.5 Supercook

Supercook excels in offering ingredient-based recipe suggestions, it has several limitations compared to more feature-rich cooking apps. The app does not support personalized meal planning, which limits its usefulness for users who want to structure their meals around specific dietary needs or health goals over a longer period. It also lacks a detailed nutritional analysis feature, which could be a drawback for users who require comprehensive nutritional information to manage health conditions or adhere to particular diets.

Additionally, Supercook does not offer the functionality to capture ingredient images and automatically identify them, which means that users must manually input all ingredients. This can be time-consuming and may lead to errors or omissions. Moreover, Supercook lacks an inventory management system to track ingredient quantities and expiration dates. Apart from that, the application does not provide voice-guided cooking instructions, which can be a valuable feature for users who prefer hands-free guidance while cooking.

2.3 Comparison of Application with the Proposed System

Table 2-3-1 below had shown the comparison of features present in the application.

Table 2.3.1 Compare Existing Application with Proposed System

Application / Feature	Smart Kitchen Assistant	CookAI	PixFood	SideChef	Supercook	The Proposed Application
identify ingredients through picture capture	√		√			√
create personalized meal plans		√	√	√		√
save their preferred recipes to a favorites list					√	√
notifies users of ingredients that are nearing their expiration date	√					√

The proposed application involves an application that enhances meal preparation and nutritional management. It features advanced ingredient recognition using image recognition technology to manage inventory, including tracking expiration dates. The app offers personalized meal planning and ingredients management. It also provides comprehensive nutritional analysis, allowing users to set health goals and receive tailored meal recommendations.

Advanced Ingredient Recognition and Management

The proposed application will use AI-driven image recognition technology to identify the ingredients from the image that users capture. Then, the recognized ingredients will be automatically updated and stored in the inventory system. The data that will be stored is ingredient names, quantities, and expiration dates. Additionally, the application will provide

real-time inventory tracking and send alert notification messages to users when the ingredients are nearing their expiration date.

Personalized Meal Planning with Suggestion

The proposed application aims to offer personalized meal plans that consider the nutritional needs, or available ingredients. Users can add their daily recipe with healthy purpose for meal nutrition planning.

Comprehensive Nutritional Analysis

The proposed application aims to cover a detailed nutritional analysis by calculating the total calories, macronutrients, micronutrients, etcetera for each recipe and ingredient. Besides that, the application will provide a nutrition suggestion based on the user's recorded nutrient intake data. As an example, weight loss objectives prefer to ingest micronutrient ingredients with fewer macronutrients, while muscle gain needs to consume more protein and a balanced carbohydrate.

CHAPTER 3

System Methodology

3.1 System Requirements

This chapter provides an overview of the methods and techniques used in mobile application development. It outlines the development approach chosen, the tools and frameworks used, and the design approach. By detailing these aspects, this chapter aims to show how the selected methods and techniques are consistent with the project goals and contribute to the effective implementation of user-friendly mobile applications.

Development Methodology

Agile methods are incremental development methods for increment that are small and able to get rapid feedback on changing requirements [15]. The design and implementation are the central activities during the software development process as shown below Figure 3.1.1.1. Therefore, the mobile application development follows the Agile methodology which ensures that it aligned with the user needs and evolving market demands throughout the development process.

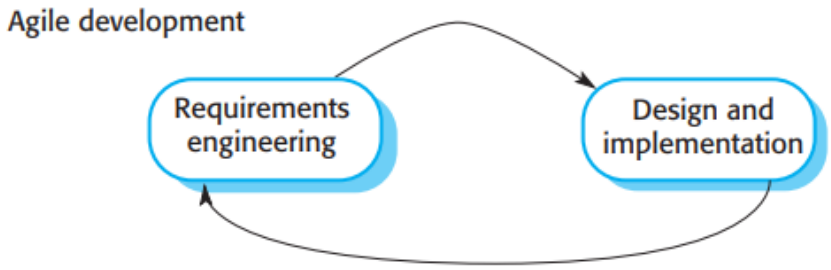


Figure 3.1.1 Agile development flows [15]

Planning: In this initial phase, defines the overall vision of the product, creates a product backlog, and prioritizes features. This phase ensures alignment with business goals and user needs.

Design: In this phase is to create wireframes, mockups, and prototypes of the application. It focuses on user experience (UX) and user interface (UI) design, ensuring the app is intuitive and visually appealing.

Development: During this phase is to start coding the features based on the prioritized backlog. This phase involves writing clean, efficient code and implementing the designed features and functionalities.

Testing: Quality assurance (QA) testers and developers perform various tests to identify and fix bugs, ensure functionality, and validate that the app meets the defined requirements.

Deployment: The tested and approved features are released to users. This may involve deploying to app stores or updating existing installations.

Review: Then, the project demonstrates the completed work to gathers feedback. This phase allows for adjustments and refinements based on stakeholder input before moving to deployment.

Feedback: User feedback is collected through various channels (e.g., app reviews, user surveys, analytics). This information is crucial for identifying areas of improvement and planning future iterations.

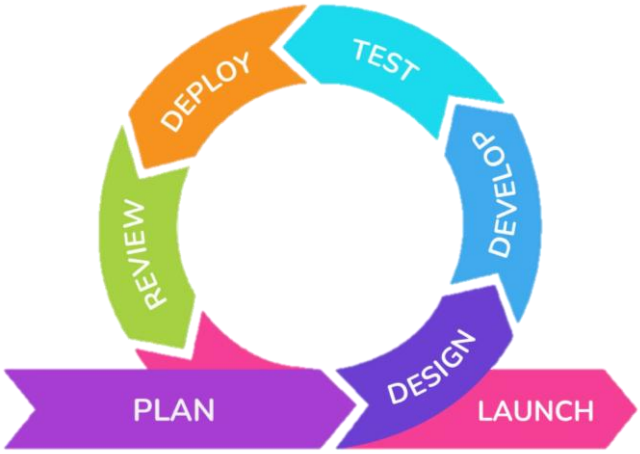


Figure 3.1.2 Software Development Circle [16]

3.2 System Design Diagram

This Figure 3.2.1 provides an overview of the system architecture that outlines a Cooking Assistant with Nutritional Tracking App.

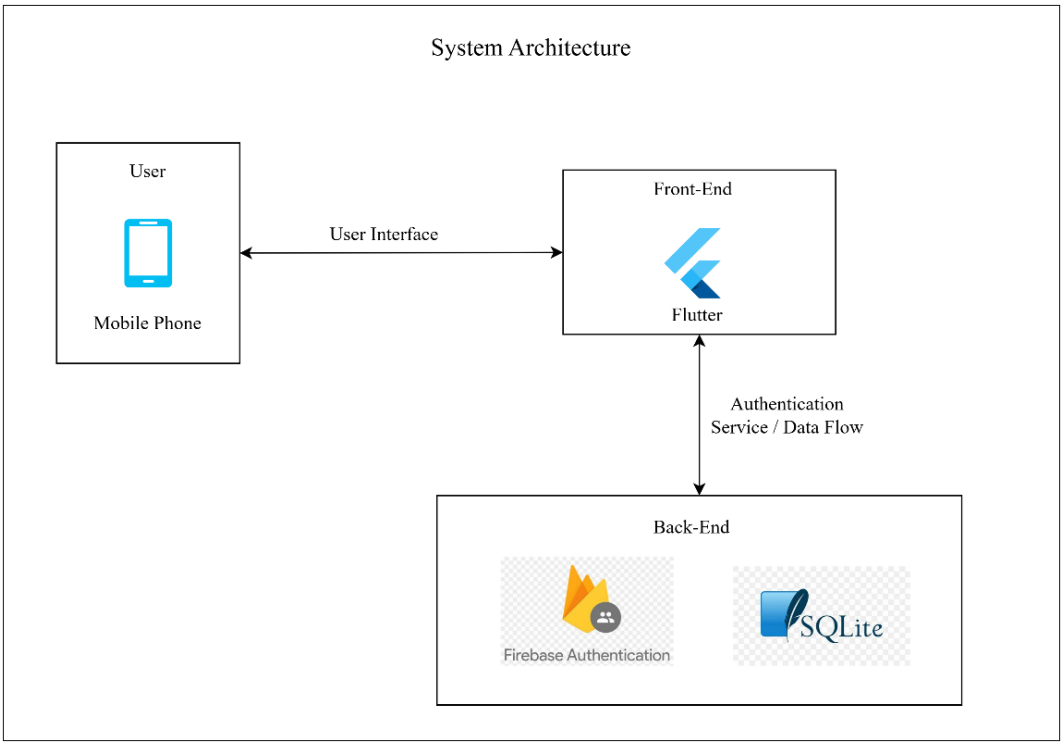


Figure 3.2.1 System Architecture Design for Mobile App

The user interacts with the application through a mobile phone interface, which connects to a Flutter front-end. The front-end communicates with a back-end system powered by Firebase Authentication and utilizes SQLite as the database. Data flows bidirectionally between the front-end and back-end, ensuring real-time updates and synchronization. This architecture allows for responsive user experience and efficient data management to provide personalized cooking and nutritional tracking assistance to users.

3.3 System Design Overview

The proposed Cooking Assistant with Nutritional Tracking application is designed as a mobile solution developed using the Flutter framework with Firebase integration for cloud services. It provides users with an intelligent and user-friendly platform to manage food ingredients, save recipes, and track nutritional information effectively. The application aims to reduce food waste and plan their daily meals while supporting healthier dietary decisions through nutritional analysis.

3.3.1 System Architecture Diagram

The application is designed using a layered architecture that separates concerns across different responsibilities. This design approach improves the maintainability, extensibility, and testability of the code.

The architecture is divided into four main layers:

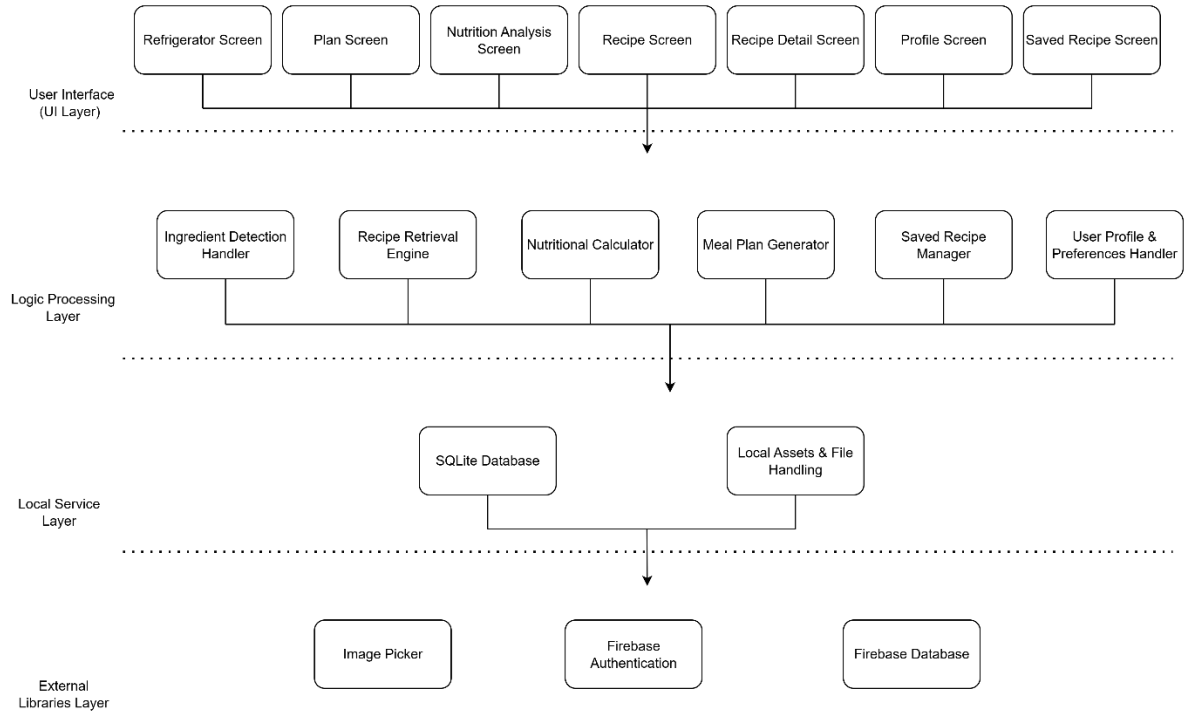


Figure 3.3.1.1 System Architecture Diagram

1. UI Layer

The UI layer consists of all user-facing screens that enable interaction with the application. Each screen serves a dedicated role in providing a seamless and engaging user experience:

- Refrigerator Screen: Displays a list of available ingredients, including those detected from images or added manually.
- Plan Screen: Allows users to create and manage meal plans based on nutritional goals.
- Nutrition Analysis Screen: Presents detailed nutritional breakdowns such as calories, macronutrients for selected recipes or daily meal plans.
- Recipe Screen: Provides a browsable catalog of recipes, searchable by keyword, or filter the recipe with cooking time and calories.

- Recipe Detail Screen: Displays step-by-step cooking instructions, ingredients, and nutritional values for a chosen recipe.
- Profile Screen: Allows users to update their account settings.
- Saved Recipe Screen: Stores and manages user-favorited recipes for quick access.

Each screen is implemented using Flutter widgets, ensuring responsive rendering and smooth navigation across the application.

2. Logic Processing Layer

This layer represents the application's functional core. It manages data processing, enforces system rules, and facilitates communication between the UI and local storage:

- Ingredient Detection Handler: Processes captured or uploaded food images and identified ingredients using the pre-trained model.
- Recipe Retrieval Engine: Searches the database to retrieve recipes that match keyword and user-defined filters.
- Nutritional Calculator: Computes and aggregates nutritional information for recipes and meal plans.
- Meal Plan Generator: Add meal plans tailored to dietary preferences, and nutrition targets.
- Saved Recipe Manager: Handles CRUD operations for saving, updating, or deleting recipes in the user's favorites.
- User Profile & Preferences Handler: Updates user dietary preferences, and profile details.

This layer ensures that data is validated, processed efficiently, and delivered consistently to the UI for real-time updates.

3. Local Services

Persistent data storage and local resources are managed through this layer:

- SQLite Database: Stores recipes, nutritional data, user preferences, ingredient lists, and cooking history.
- Local Assets & File Handling: Provides access to static resources such as icons, default images, pre-trained model files and manages local read and write operations.

This approach supports offline access, enabling users to browse recipes, manage ingredients, and track nutrition without an internet connection.

4. External Libraries and Tools

External packages are integrated to provide advanced functionality and streamline development:

- Image Picker: Enables users to capture or select food images from their device.
- Firebase Authentication: Secures user login and identity management.
- Firebase Database: Provides cloud storage and synchronization for recipes and user data.

These tools extend the core system, adding intelligent detection, secure user management, and enhanced usability.

3.3.2 Use Case Diagram

Figure 3.1.2.1 below shows the use case diagram containing 2 actors which were User and Admin. Each of them has their respective role and functionality. Admin has the full right to the system including managing user and recipe. Users are only able to access all the functionality except the user management due to the access control done by Admin. All the actors can manage their ingredients, plan their meal, track the nutrition status and search for the recipe.

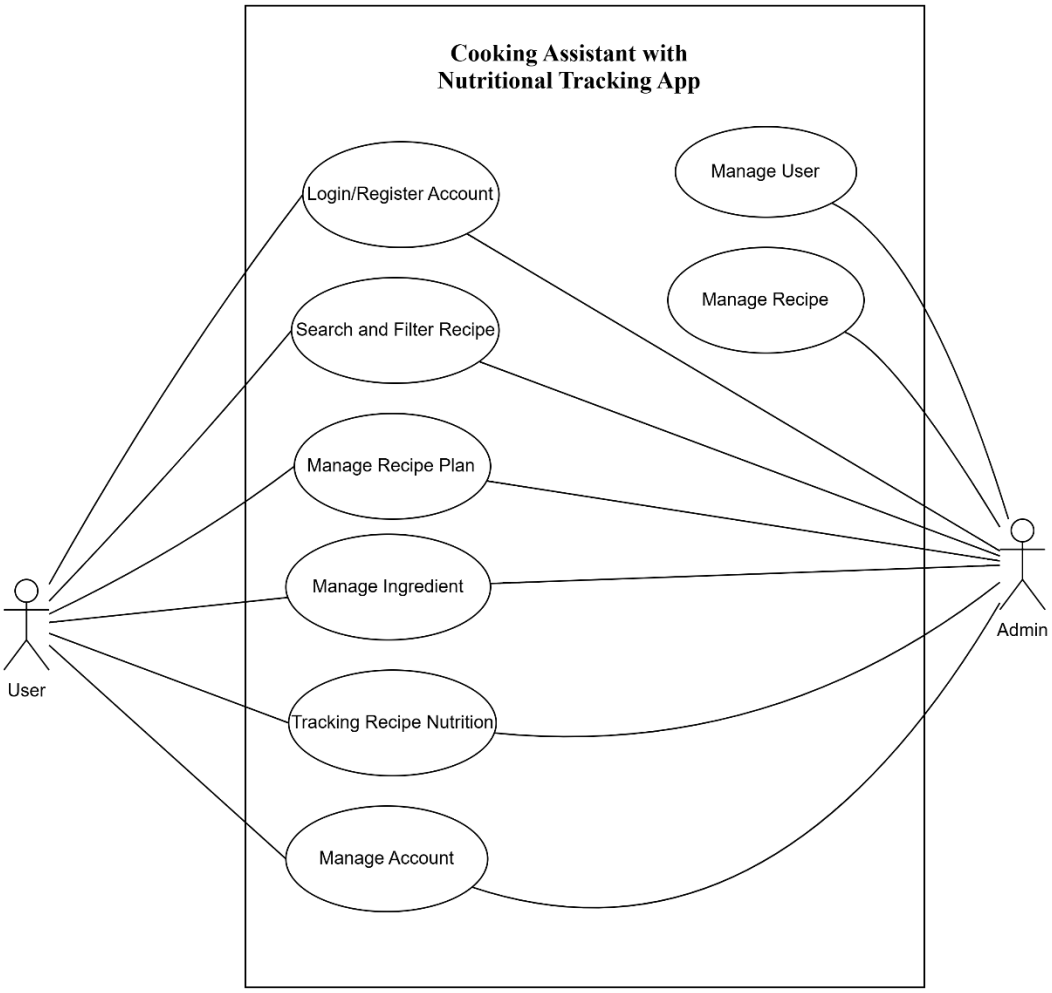


Figure 3.3.2.1 Use Case Diagram

3.3.3 Activity Diagram

Activity Diagram for Login and Registration

Figure 3.1.3.2 below shows the Activity Diagram for Login and Registration. During registration, Users need to enter their personal information such as email addresses, and the password twice. Then, the system will validate the information format and data. If the input is valid, the account is created otherwise the user requires to enter the information again. During the login process, the user requires providing correct email and password to redirected to home page of the application. There is an option to reset password when users forget their password.

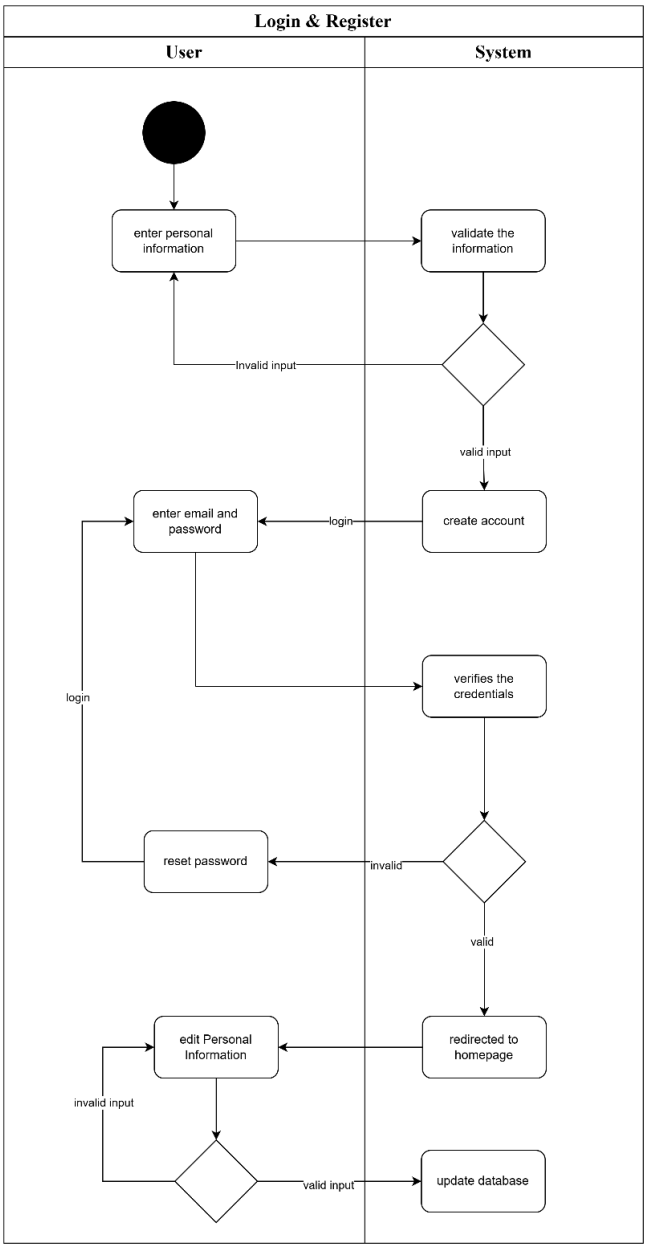


Figure 3.3.3.2 Activity Diagram for Login and Registration

Activity Diagram for Ingredient Management

The food ingredients management activity diagram below shows the interaction between the user and the system when processing food ingredients. Users can view the ingredients list, add ingredients, edit existing ingredients, or delete ingredients. Then, the system will handed the responds by updating the database with adding, modifying, or deleting entries as necessary. This process ensures that the ingredients library remains accurate and up-to-date. Morever, it allows for seamless recipe management and meal planning. This figure 3.1.3.3 highlights a clear bidirectional flow where user actions trigger specific system responses to emphasize efficiency of ingredient processing and the data integrity.

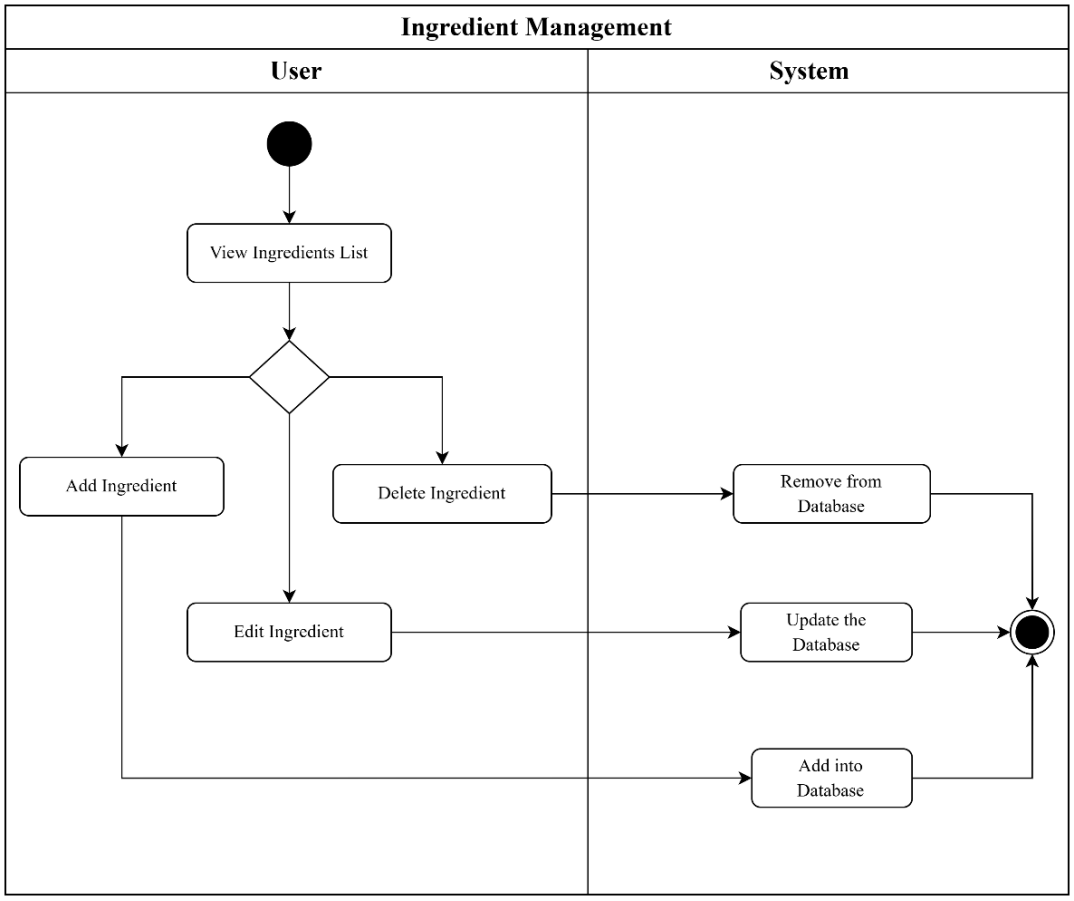


Figure 3.3.3.3 Activity Diagram for Ingredient Management

Activity Diagram for Recipe Browsing Module

The recipe management activity diagram shows the workflow of users searching, selecting and managing recipes. The user initiates a search and the system checks the available ingredients and then matches them with recipes. The system will then display to the user the results of the food ingredients contained in that particular recipe. In recipe management, users can select a recipe, view its details, switch it to favorites, or add it to a meal plan. Figure 3.1.3.4 show the activity diagram highlights the role of the system in connecting user preferences with available ingredients, ensuring a personalized and efficient recipe discovery experience.

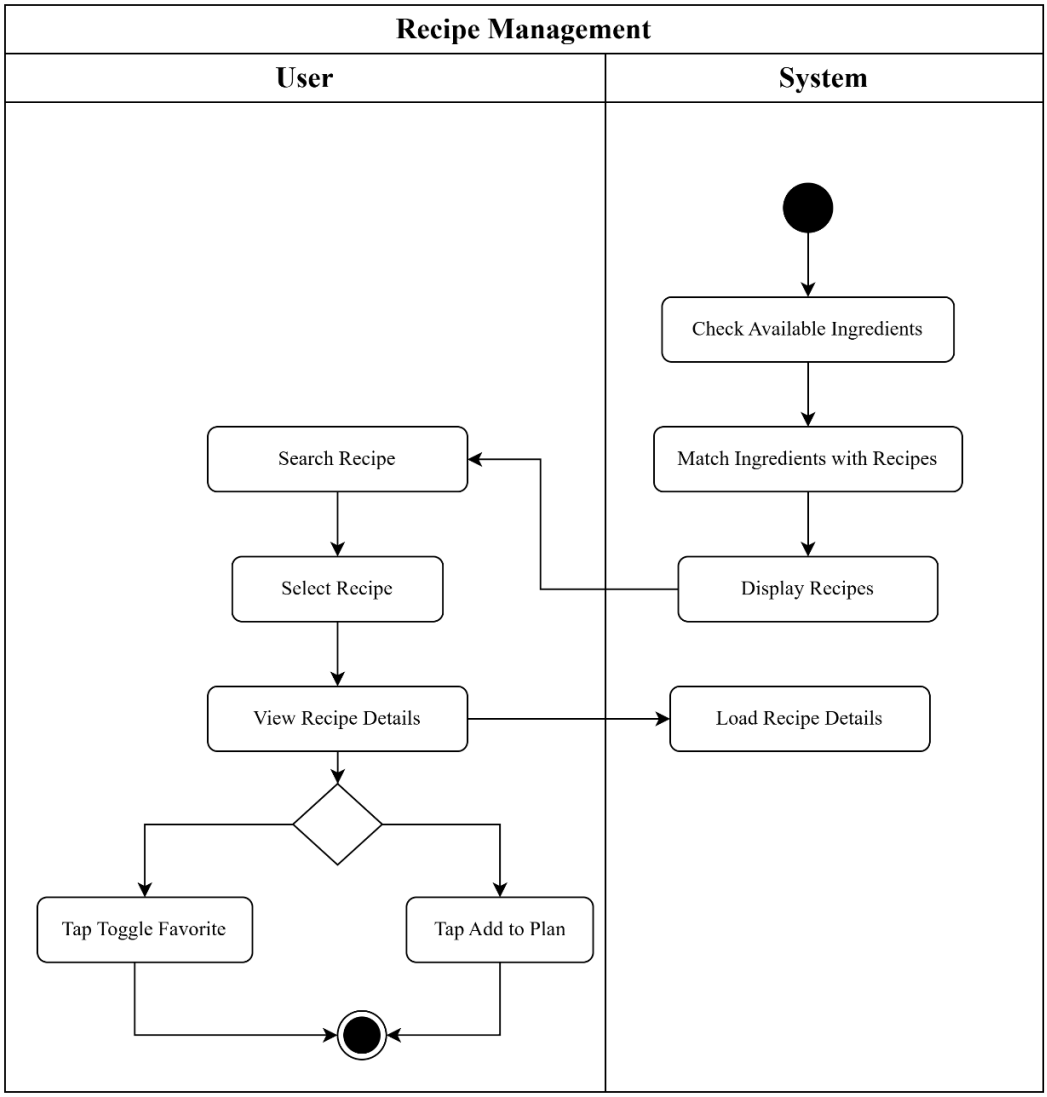


Figure 3.3.3.4 Activity Diagram for Recipe Browsing Module

Activity Diagram for Meal Nutrition Plan Module

The Meal Nutrition Planning Activity Diagram outlines the process of developing and managing a daily meal plan. The user selects the date and meal type. This will trigger the date picker to appear and load the relevant nutritional information. When the user adds a new recipe, the system will pop up a selection dialog box for the user to update the diet plan and recalculate the nutritional information and output it to the user. The activity diagram below highlights the system's ability to provide instant feedback, such as a nutritional overview and updated meal plan, to ensure users can make informed dietary choices while maintaining a structured and intuitive planning process.

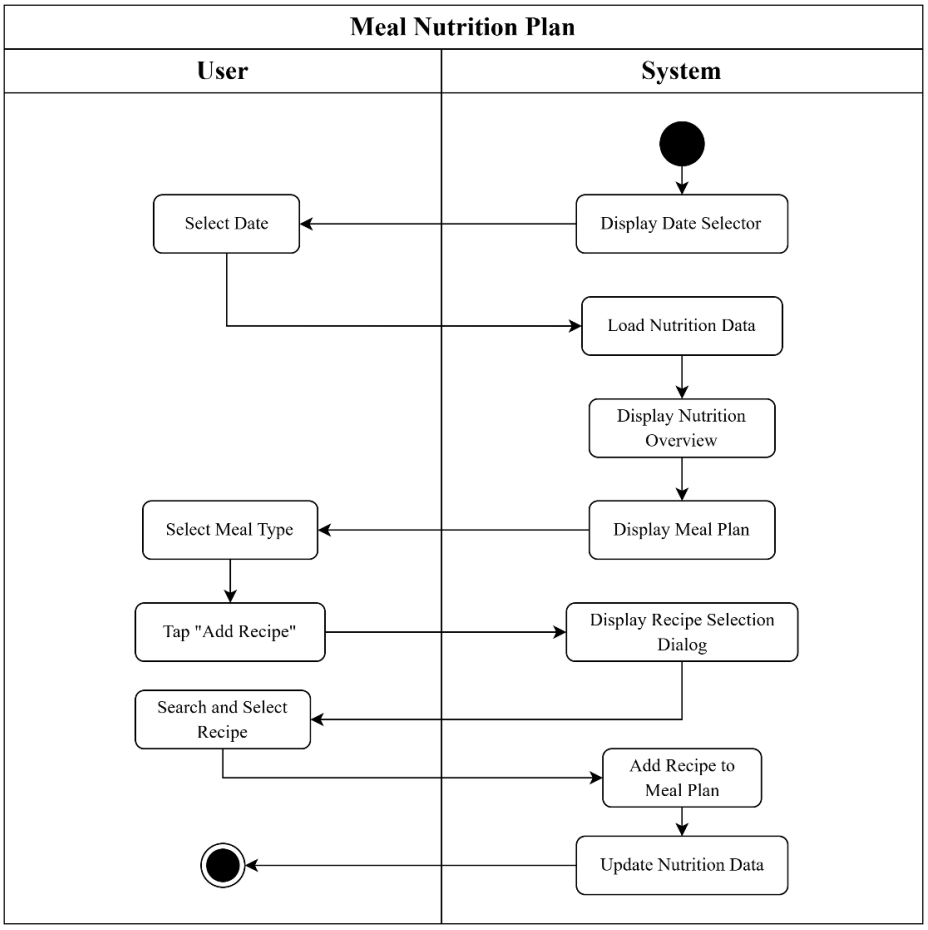


Figure 3.3.3.5 Activity Diagram for Meal Nutrition Plan Module

3.4 Timeline

Figure 3.4.1 shows the iterative timeline of the proposed application during the FYP1 stage, which follows agile principles. Unlike the rigid structure of the waterfall model, agile development consists of sprints where tasks are continually refined based on feedback. There is total 12 sprints and the timeline shows flexible and overlapping phases rather than strictly sequential steps, allowing adjustments to be made at each iteration.

Figure 3.4.2 illustrates the FYP2 timeline, which extends the agile sprint structure into the implementation and evaluation phases of a project. This timeline consists of 12 sprints, covering activities ranging from detailed design, module development, and system integration to progressive testing and final report preparation.

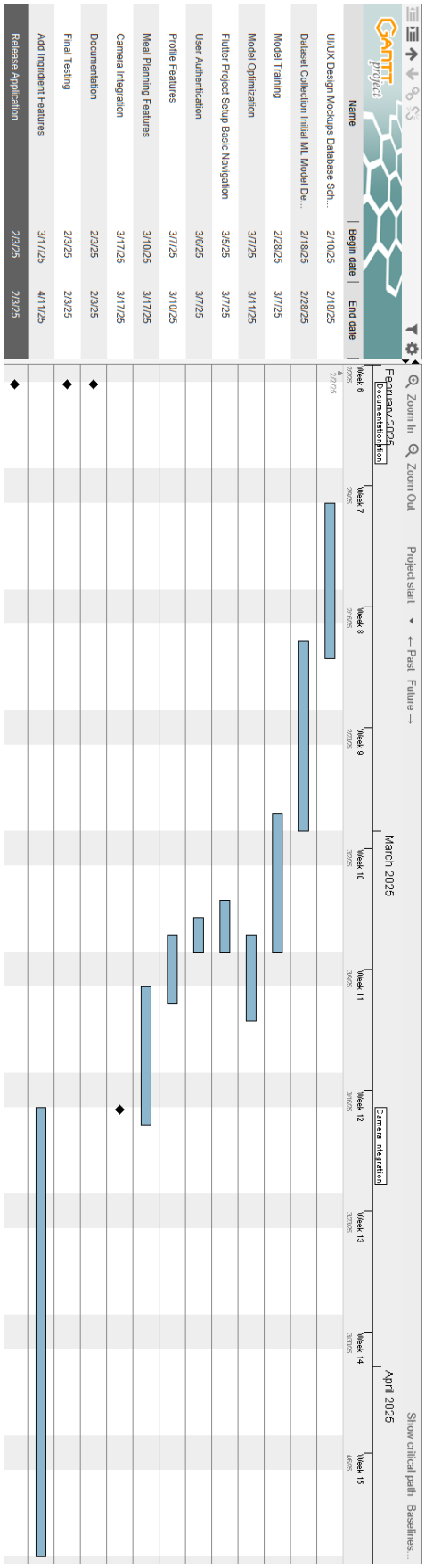


Figure 3.4.1 Gantt Chart FYP1 timeline

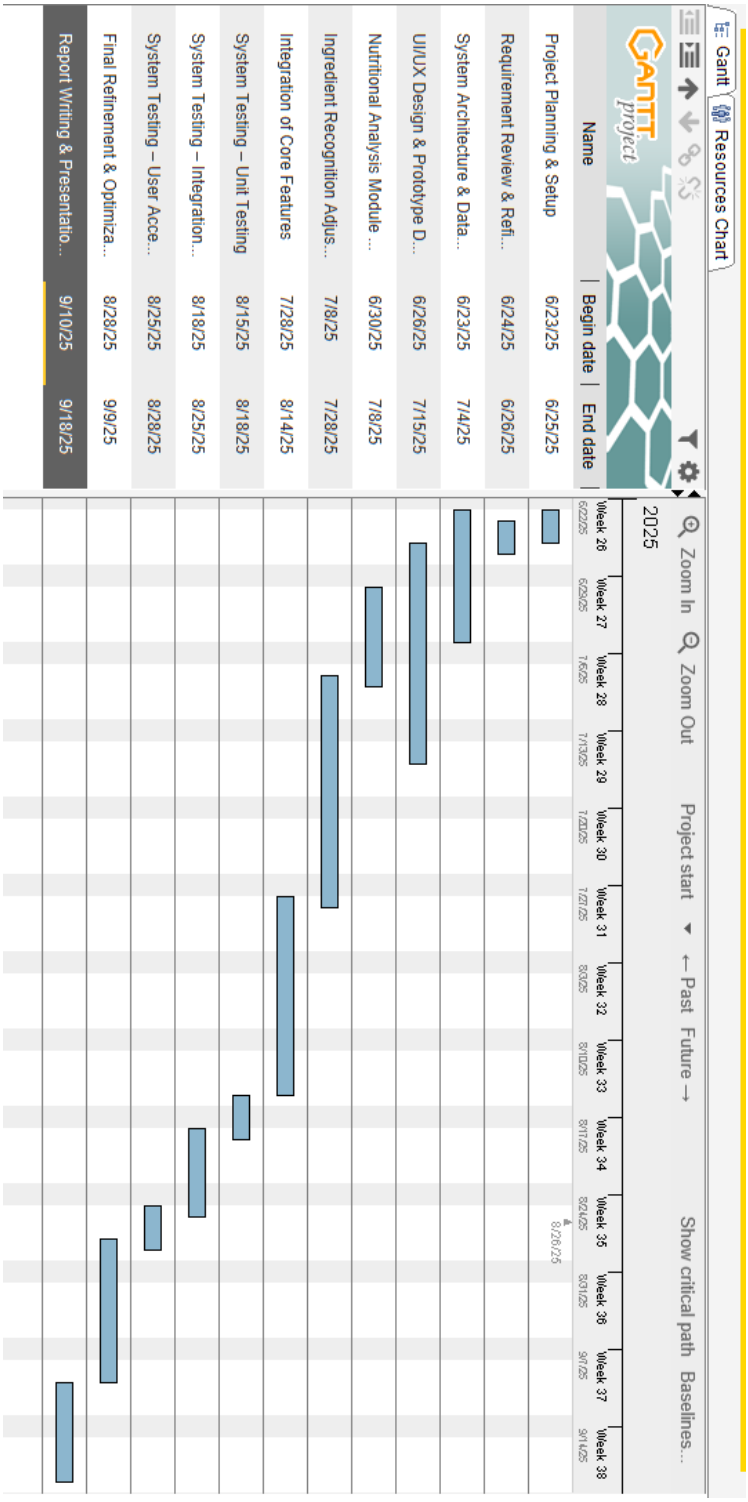


Figure 3.4.2 Gantt Chart FYP2 timeline

CHAPTER 4
System Design

4.1 System Block Diagram

The system block diagram illustrates the overall structure and interaction between the key components within the application. It outlines the flow of data and the connection between the user interface, logic processing, and the storage layer.

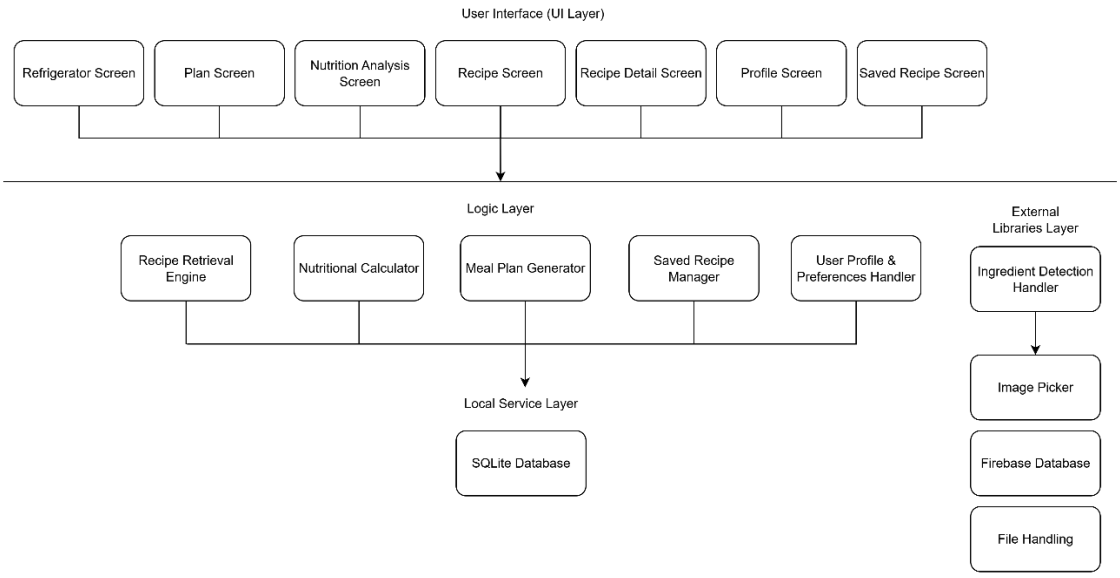


Figure 4.1.1 System Block Diagram

The application consists of the following key components:

1. User Interface (UI):

The UI layer manages all user interactions and provides access to the system's core features. It is implemented using Flutter widgets to ensure responsiveness and usability. The key screens include:

- Refrigerator Screen: Displays and manages ingredients, including those detected ingredients from images or added manually.
- Plan Screen: Allows users to create and manage meal plans.
- Nutrition Analysis Screen: Provides nutritional insights, including calories and macronutrient breakdowns.
- Recipe Screen: Enables browsing of recipes.

- Recipe Detail Screen: Presents detailed cooking instructions, ingredient lists, and nutritional information.
- Profile Screen: Manages user profiles, and preferences.
- Saved Recipe Screen: Stores recipes that the user has marked as favorites for quick retrieval.

2. Logic Layer:

This layer processes user input enforces application rules, and coordinates communication between the UI and database. It includes the following components:

- Ingredient Detection Handler: Identifies food items in images using the pre-trained model.
- Recipe Retrieval Engine: Searches and retrieves recipes that match available keyword or user filters.
- Nutritional Calculator: Computes nutritional values for recipes and meal plans.
- Meal Plan Generator: Create personalized meal plans based on dietary preferences.
- Saved Recipe Manager: Handles operations for storing and retrieving user-favorite recipes.
- User Profile & Preferences Handler: Updates and maintains user preferences.

3. SQLite Database:

The local SQLite database serves as the persistent data store for all application data. It stores recipes, nutritional information, ingredient inventories, and user preferences. The use of a local database ensures that the application functions effectively in offline mode.

4. External Libraries and Tools:

Several external libraries are integrated to extend the core functionality of the system:

- Image Picker: Captures or selects images of food items from the device.
- Firebase Database: Provides cloud storage and synchronization if online access is enabled.
- File Handling: Manages access to local file storage and pre-trained model files.

5. Block Diagram Explanation:

The system architecture follows a layered design to ensure modularity and scalability:

The UI layer initiates user actions, such as scanning ingredients, browsing recipes, or analyzing nutrition.

The Logic layer processes these requests by detecting ingredients, creating meal plans, or calculating nutritional values.

Processed data is stored in or retrieved from the SQLite database, ensuring persistence and offline usability.

External libraries and tools provide specialized support, including image capture, ingredient detection, and data synchronization.

This modular separation enhances maintainability, scalability, and offline reliability, enabling the Cooking Assistant with Nutritional Tracking App to deliver smooth and efficient user experience.

4.2 System Components Specifications

This section outlines the core components of the cooking assistant application with integrated nutrition tracking. The application is developed using Flutter and Dart follows modular architecture to ensure scalability, maintainability, and enhanced user experience. The application is primarily designed for offline functionality, with a local SQLite database serving as the main storage solution. It also incorporates advanced features such as ingredient recognition, recipe recommendations, and nutrition tracking to support users in efficiently planning meals and developing healthier eating habits.

User Interface Components

The user interface (UI) serves as the interaction point between users and the system's functionalities. Designed with usability and clarity in mind, each screen is tailored to support a specific task while ensuring smooth navigation and responsiveness. The main UI components are:

Table 4.2.1 user interface components

UI Screen	**Functionality Description**
Refrigerator Screen	Allows users to manage ingredient inventories by adding, editing, or deleting items. Detected ingredients from image scans are also displayed here.
Plan Screen	Provides tools for generating and managing meal plans, tailored to user preferences.
Nutrition Analysis Screen	Displays nutritional breakdowns, such as calories and macronutrient values, for meals and individual recipes.
Recipe Screen	Enables users to browse and filter recipes.
Recipe Detail Screen	Presents full recipe information including preparation steps, ingredients, and nutrition data.
Profile Screen	Manages user profile details, and personalization preferences.
Saved Recipe Screen	Stores recipes bookmarked by the user for quick access and reuse.

Logic Layer Components

The logic layer manages the system's core processing and business rules. It ensures that data from the UI is properly processed, validated, and synchronized with storage. Key components include:

Table 4.2.2 Logic Layer Components

Logic Component	**Role and Functionality**
Ingredient Detection Handler	Identifies and extracts ingredient information from uploaded or captured images using an integrated detection model.
Recipe Retrieval Engine	Search and retrieves recipes that match available keyword or user filters.
Nutritional Calculator	Computes nutritional information for recipes and meal plans, supporting informed dietary decisions.
Meal Plan Generator	Creates personalized meal schedules that align with dietary goals.
Saved Recipe Manager	Handles operations for storing, updating, and retrieving user-favorited recipes.
User Profile & Preferences Handler	Maintains user data such as favorite cuisines.

Local Storage Layer

The application employs SQLite as its primary storage solution, ensuring offline functionality and structured data management. It maintains:

- Recipes and Nutritional Information
- Ingredient Inventories
- User Profiles and Preferences
- Saved Recipes and Meal Plans

This offline approach allows users to access and manage their data seamlessly without depending on internet connectivity.

External Libraries and Tools

Several third-party libraries are integrated to extend the system's capabilities:

Table 4.2.3 External Libraries and Tools

Library / Tool	**Purpose and Contribution**
Image Picker	Enables users to capture images with the camera or select from the gallery.
Firebase Database	Provides online synchronization of user data when internet access is available.
File Handling	Supports local file access, storage, and management of assets such as images and pre-trained models.

By combining these, the system provides a comprehensive cooking assistant that not only manages ingredients and recipes but also supports nutrition tracking and personalized meal planning. Its modular design ensures extensibility, allowing future integration of advanced features without significant architectural changes.

4.3 Components Design

This section outlines the design of the main component in the application, organized according to the architectural layers. Its purpose is to describe the role of each component and how they integrate to build a cohesive, maintainable, and scalable mobile application. The system is structured into four main layers: the User Interface (UI) layer, the Logic layer, Local Services, and External Libraries.

4.3.1 UI Layer Components

The User Interface (UI) layer serves as the primary interaction point between the user and the system. Each screen is designed to be intuitive, responsive, and consistent with user expectations for managing ingredients, recipes, and nutritional information.

1.Refrigerator Screen

Displays and manages the list of available ingredients. Users can add or remove items manually or update the inventory using image-based ingredient detection.

2.Plan Screen

Provides functionality for creating and viewing meal plans. Users can generate customized plans based on stored ingredients and dietary preferences.

3.Nutrition Analysis Screen

Presents nutritional information such as calorie content, macronutrient breakdowns, and dietary suitability for selected recipes or meal plans.

4.Recipe Screen

Allows users to browse and filter recipes by ingredient availability, categories, or personal preferences.

5.Recipe Detail Screen

Displays detailed recipe information including preparation steps, required ingredients, and nutritional values.

6.Profile Screen

Manages user information, dietary goals, and personal preferences. Supports customization to improve recipe recommendations.

7.Saved Recipe Screen

Stores and displays recipes bookmarked by the user for easy access and reuse.

4.3.2 Logic Layer Components

The Logic layer processes data, enforces business rules, and ensures that user actions in the UI are translated into consistent system behavior. It acts as the application's core processing engine.

1.Ingredient Detection Handler

Utilizes the integrated pre-trained model to identify ingredients from captured or uploaded images. Extracted items are forwarded to the refrigerator inventory.

2.Recipe Retrieval Engine

Search and retrieves recipes from the local database that match keyword, dietary constraints, and user preferences.

3.Nutritional Calculator

Computes calorie counts, macronutrient breakdowns, and dietary suitability for individual recipes or entire meal plans.

4.Meal Plan Generator

Create meal plans tailored to user-defined goal. Ensure balance across nutrition targets.

5.Saved Recipe Manager

Handles operations for saving, retrieving, and updating user-favorited recipes.

6.User Profile & Preferences Handler

Maintains user preferences, including dietary restrictions, nutritional goals, and personalization data.

4.3.3 Local Services

This layer provides persistent data storage and ensures offline usability of the system.

SQLite Database

Acts as the primary storage solution for recipes, nutritional data, ingredient inventories, user profiles, preferences, and saved recipes. Supports structured queries and relational consistency.

4.3.4 External Libraries

The system integrates external libraries and tools to extend functionality beyond the native capabilities of Flutter and Dart.

1.Image Picker

Allows users to capture food images from the device camera or select from the gallery. Works in conjunction with the ingredient detection module.

2.Firebase Database

Provides cloud-based synchronization of user data when online connectivity is available.

3.File Handling

Supports local asset management, preprocessing of images, and storage of model files required for detection and analysis.

4.4 System Components Interaction Operation

This section explains how the major components interact during key operations of the cooking assistant application. It describes the flow of data and the collaboration between the UI, logic, local services, and external libraries to deliver essential functionalities such as managing ingredients, scanning items, generating recipes, tracking nutrition, and providing personalized recommendations. These interactions ensure the application operates smoothly and offer intuitive user experience.

4.4.1 Adding Ingredients

User Input (UI Layer)

The user navigates to the Add Ingredient screen and enters details such as ingredient name, quantity, and expiration date. Alternatively, the user can capture an image of the ingredient for recognition.

Image Recognition (External Library + Logic Layer)

The image is captured using the camera input. The pre-trained model processes the image to detect and classify the ingredient. Detected results are validated and linked to ingredient records in the database.

Ingredient Processing (Logic Layer)

The recognized or manually entered ingredient data is sent to the Ingredient Handler.

The system updates the local SQLite database with the new or updated ingredients.

4.4.2 Managing Ingredient Inventory

Loading Ingredient Data (UI + Local Services)

The Refrigerator screen retrieves stored ingredient data from SQLite. Details such as quantity and expiration are displayed in a structured list.

Updating Inventory (Logic Layer)

When the user edits or removes ingredients, changes are validated and saved back into the database. Expired items are flagged, and notifications are triggered if necessary.

4.4.3 Nutrition Tracking

Nutritional Data Retrieval (Logic + External Libraries)

Each recipe or ingredient is linked to nutritional information stored in the database or retrieved from integrated nutrition APIs.

Daily Log Update (Logic Layer)

When a recipe is marked as “cooked” or “consumed,” the nutritional values are added to the user’s daily intake record. Progress against daily nutritional goals is updated accordingly.

Visualization (UI Layer)

The system presents daily summaries using charts or progress bars, highlighting calorie intake, macronutrient distribution, and other tracked values.

4.4.4 Ingredient Expiry Alerts

Expiry Monitoring (Logic Layer)

Each time the application is launched, or ingredients are updated, the system checks expiration dates.

These interaction flows demonstrate how the different components of the cooking assistant app work together to support ingredient management, recipe recommendations, nutritional tracking, and user notifications. By tightly integrating the UI, logic, local storage, and external models, the system ensures both usability and reliability for daily cooking assistance.

4.5 Model Training Setup

4.5.1 Recipe Data Process

Recipe database from Kaggle is extracted and downloaded as the project recipe data with the table below code. The downloaded dataset is provided in .csv format and it is then converted into JSON type through a Python script because it is easier to parse and integrate into Flutter project apps with a structured processing.

Table 4.5.1 Import dataset code

```
import kagglehub
path = kagglehub.dataset_download("irkaal/foodcom-recipes-and-reviews")
print("Path to dataset files:", path)
```

4.5.2 Ingredient Recognition Model Training

4.5.2.1 Importing Dataset and libraries

The food ingredients dataset is retrieved from Kaggle dataset where it contains 36 different classes of label image. The dataset is separated into 3 different folders which are train, test, and validation. Each of the classes contains 100 images inside the train and test file while 10 images for each class in the validation file.

Training recognition model in the google colab platform, there are 3 imported libraries which are NumPy, TensorFlow, and Matplotlib.pyplot. The NumPy will be used for processing image data as numerical arrays and performing mathematical operations on pixel values, and handling batch processing of multiple images. Next, the TensorFlow is used for building the convolutional neural network architecture, evaluating model performance with built-in metrics, and saving trained models. In the training platform, Matplotlib.pyplot is used for visualizing training and validation accuracy curve.

Table 4.5.2 Import library code

```
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
```

4.5.2.2 Data Image Preprocessing

The key parameters for the data image preprocessing:

Table 4.5.3 Key parameter used for the image preprocessing

Key Parameter	Description
labels= “inferred”	The system will automatically determine class labels based on the subdirectory structure with each type of ingredient having its own folders.
label_mode = “categorical”	This configures one-hot encoded labels especially for multiple class classification of different ingredient types.
batch_size = 32	This batch size can optimize memory usage and training efficiency.
image_size= (64,64)	Resized all the image to have a uniform dimension of (64, 64) pixels.
shuffle= true	This parameter is used to randomize the order of images during training and validation.

4.5.2.3 Building Model

tf.keras.models.Sequential() is used to build a linear stack of layers where each layer has exactly one input and output tensor. There are two main convolutional blocks. In the first blocks uses two Conv2D layers with 32 filters to detect simple features like edges and colors. Next, the second blocks added two more layers with 64 filters to learn more detailed and complex patterns. Both of the convolutional blocks used a MaxPooling layer reduces the size of the image, and a Dropout layer helps prevent overfitting.

After that, the image features are flattened into a one-dimensional vector. Then, it goes through two Dense layers which is 512 and 256 units with ReLU, followed by a higher Dropout for better regularization. Finally, a Dense layer with 36 units and softmax activation gives the prediction one of the 36 ingredient classes. Next, compiling and evaluating models until the desired accuracy is achieved. The model is then saved in h5 file type.

Table 4.5.4 Code for building mode

```
cnn.add(tf.keras.layers.Conv2D(filters=32,kernel_size=3,padding='same',activation='relu',i
nput_shape=[64,64,3]))
cnn.add(tf.keras.layers.Conv2D(filters=32,kernel_size=3,activation='relu'))
```

```
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))
cnn.add(tf.keras.layers.Dropout(0.25))
cnn.add(tf.keras.layers.Conv2D(filters=64,kernel_size=3,padding='same',activation='relu'))
cnn.add(tf.keras.layers.Conv2D(filters=64,kernel_size=3,activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))
cnn.add(tf.keras.layers.Dropout(0.25))
cnn.add(tf.keras.layers.Flatten())
cnn.add(tf.keras.layers.Dense(units=512,activation='relu'))
cnn.add(tf.keras.layers.Dense(units=256,activation='relu'))
cnn.add(tf.keras.layers.Dropout(0.5)) #To avoid overfitting
#Output Layer
cnn.add(tf.keras.layers.Dense(units=36,activation='softmax'))
cnn.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
cnn.save('trained_model.h5')
```

CHAPTER 5
System Implementation

This chapter provides an overview of the methods and techniques used in mobile application development. It outlines the development approach chosen, the tools and frameworks used, and the design approach. By detailing these aspects, this chapter aims to show how the selected methods and techniques are consistent with the project goals and contribute to the effective implementation of user-friendly mobile applications.

5.1 Hardware Setup

The hardware involved in this project consists of a laptop computer and an Android mobile device. The laptop is used for development, testing, and deployment of the application, including the implementation of the React-based front-end, Node.js back-end, and integration with the Firebase database and Gemini API. The Android mobile device is used for testing and running the final application, allowing users to interact with the mobile application features in a portable format.

The specifications of the hardware used in this project are as follows:

Table 5.1.1 Specifications of laptop

Description	Specifications
Model	Asus VivoBook 14 series
Processor	Intel Core i5-1135G7
Operating System	Windows 11
Graphic	Intel Iris® Xe Graphics
Memory	8GB RAM
Storage	512GB M.2 NVMe™PCIe® 3.0 SSD

Table 5.1.2 Specifications of android mobile devices

Description	Specifications
Model	Redmi2201117TG
Processor	Snapdragon 680
Operating System	1.0.7.0 TGCMIXM
Memory	6.0 GB RAM
Storage	128GB

5.2 Software Setup

Table 5.2.1 Specifications of Software

Description	Specifications
Operating System	Windows 11 Version
Programming Environment	Visual Studio Code
Database System	Firebase

Table 5.2.2 Specification of Tools

Description	Specifications
Framework	Flutter (with Dart)
UI Design	Figma
Model Training Platform	Google Colab

5.3 Setting and Configuration

This section outlines the key configuration and setup steps required to ensure a cooking assistant app with nutrition tracking functionality works properly in development and testing environments. This setup includes configuring the Flutter environment, Android permissions, initializing an SQLite database, and integrating an external library for ingredient recognition.

5.3.1 Flutter Environment Configuration

Before development, install the Flutter SDK and integrate it with the Android Studio IDE, configure environment variables, and use the AVD Manager to prepare an emulator for app testing.

- **Flutter SDK Path**: Added to system environment variables for global access.
- **Dart SDK**: Bundled with Flutter and used for core application logic.
- **AVD Manager**: Configured to create and run Android emulators for testing the app on multiple devices.
- **Flutter Doctor**: Executed to verify the environment and detect missing dependencies.

5.3.2 Android Manifest Configuration

Several permissions and services were declared in the AndroidManifest.xml file to support the application’s features. These ensure proper access to hardware and system-level operations:

Table 5.3.2.1 Android Manifest Configuration

Feature	Permission/Service Added
Camera Access	<uses-permission android:name="android.permission.CAMERA"/>
Storage Access	<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/> <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

5.3.3 SQLite Database Initialization

This application uses SQLite as a lightweight local database to ensure user data such as ingredients, recipes, and nutritional logs is securely stored on the device. The database schema is initialized using the sqflite package in Flutter when the app first launches. The _onCreate method defines several relational tables to support inventory management, recipe recommendations, and nutritional tracking.

The schema is normalized to ensure consistency and efficient data retrieval. The following is a breakdown of the core tables:

Table 5.3.3.1 ingredients

Field	Type	Description
name	TEXT	Ingredient name
quantity	REAL	Amount available
category	TEXT	Ingredient category
expiryDate	TEXT	Expiration date of the ingredient
imagePath	TEXT	File path to stored ingredient image

Table 5.3.3.2 recipes

Field	Type	Description
id	TEXT	Unique identifier for each recipe.
name	TEXT	The name of the recipe.
image	TEXT	Path or URL reference to the recipe's image.
cookingTime	INTEGER	Estimated cooking time in minutes.
calories	INTEGER	Total calorie count of the recipe.
isFavorite	BOOLEAN	Indicates whether the recipe is marked as a favorite by the user.
ingredients	RELATION	Links to a list of ingredients required for the recipe (via Ingredients table).
steps	RELATION	Defines the ordered cooking steps (via CookingSteps table).
nutritionInfo	RELATION	Links to nutritional details of the recipe (via NutritionInfo table).

Table 5.3.3.3 nutrition

Field	Data Type	Description
id	INTEGER	Unique identifier for each nutritional log entry.
date	DATE	The date of the logged nutritional record.

calorieGoal	INTEGER	Target daily calorie intake, default set to 2000 kcal.
fatGoal	REAL	Target daily fat intake in grams, default 78 g.
saturatedFatGoal	REAL	Target daily saturated fat intake in grams, default 20 g.
cholesterolGoal	REAL	Target daily cholesterol intake in milligrams, default 300 mg.
sodiumGoal	REAL	Target daily sodium intake in milligrams, default 2300 mg.
carbsGoal	REAL	Target daily carbohydrate intake in grams, default 275 g.
fiberGoal	REAL	Target daily fiber intake in grams, default 28 g.
sugarGoal	REAL	Target daily sugar intake in grams, default 50 g.
proteinGoal	REAL	Target daily protein intake in grams, default 50 g.

Table 5.3.3.4 user

Field	**Data Type**	**Description**
userId	INTEGER	Unique identifier for each user (primary key).
email	TEXT	User's registered email address, used for authentication and communication.
passwordHash	TEXT	Encrypted password string for secure authentication.

savedRecipes	TEXT/JSON	Stores a list of recipe IDs saved by the user as favorites.
createdAt	DATETIME	Timestamp indicating when the account was created.

This scheme supports the application core functionality, including ingredient tracking, recipe browsing, and nutrition monitoring, while maintaining scalability and offline performance.

5.4 System Operation

5.4.1 User Authentication Module Page

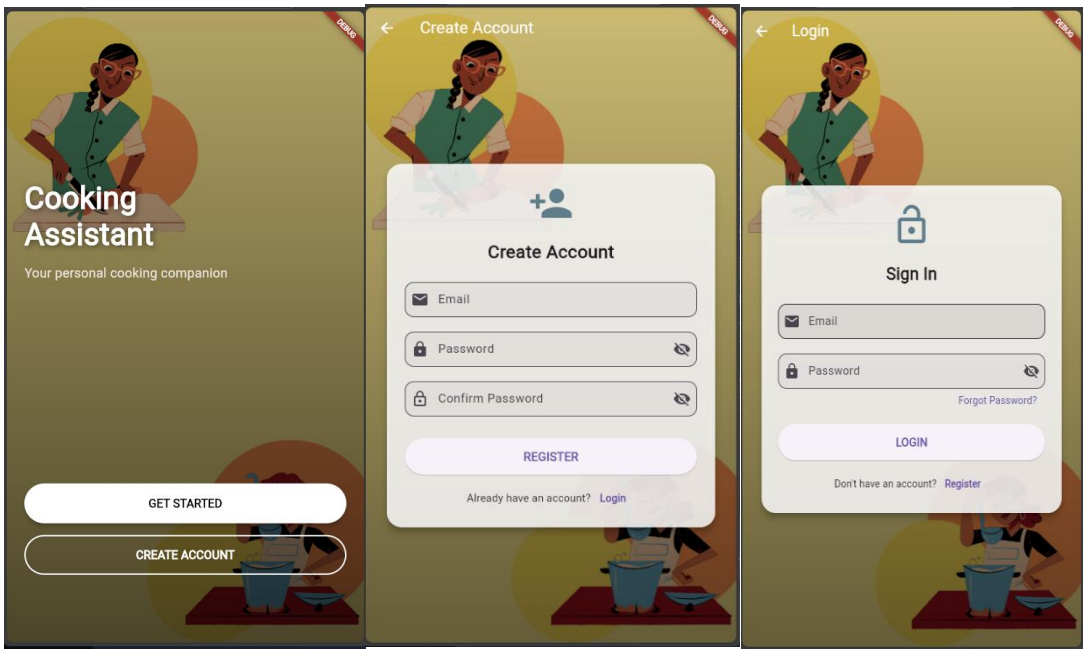


Figure 5.4.1.1 User Authentication Module Page

The authentication process consists of three main screens. When users enter cooking assistant with nutritional tracking app it will navigate to a welcome page. Users have 2 selection either get started to process log in if they already have an account or create a new account by tap on the create account. New users require to enter a valid email address, and a secure password. A valid email address have its purpose when user forget their password, a reset password link will be sending to the email address entered. Existing users log in using their email address and password to access more features provided by the application.

5.4.2 Recipe Browsing Module Page

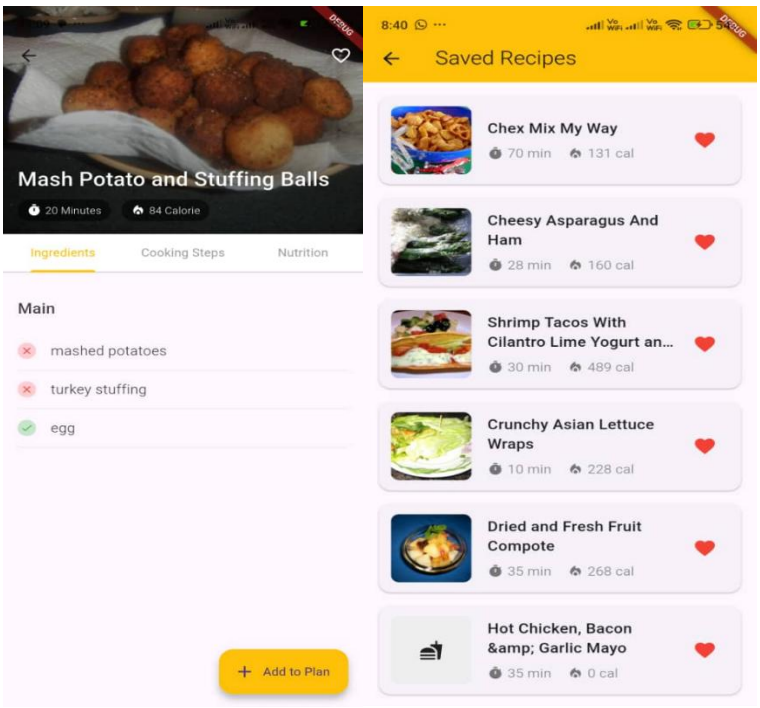


Figure 5.4.2.1 Recipe Browsing Module Page

After successfully log into the account, users can start to explore, save and manage recipes. Users can search the recipes by entering a keyword and then tap on desired recipes to browse full recipe information including preparation time, calories, and ingredients. The recipe detail page contains 3 tabs which are ingredients, cooking steps and nutrition information. At the ingredients tab it shows green tick marks if there are available ingredients that users have into their ingredient's library while a red cross mark shows unavailable ingredients. The Add to Plan button is for users to add to their meal plan with the purpose of further track nutrition. At the right top corner have favorite toggle, users can choose whether to save this recipe for future review.

5.4.3 Add Ingredient Module Page

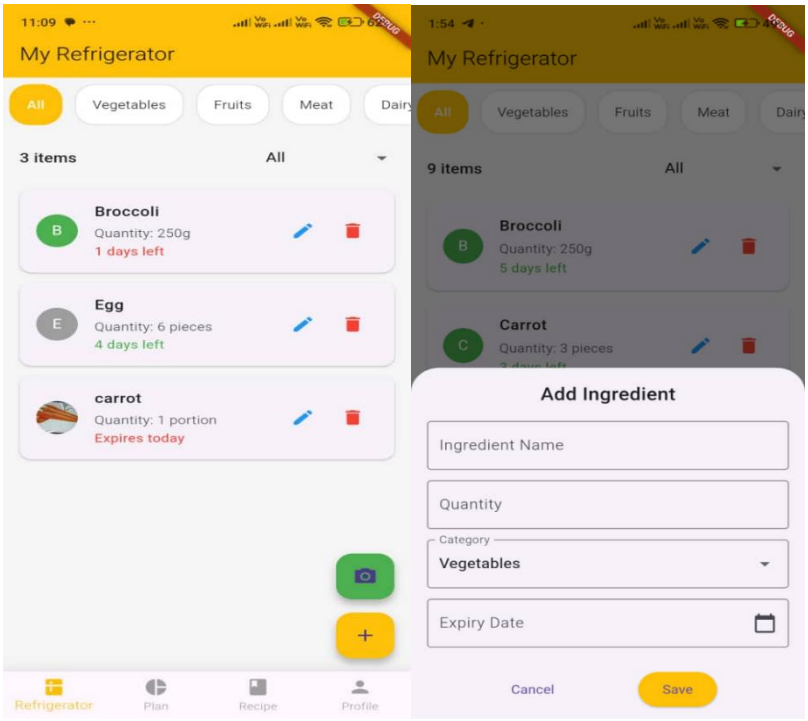


Figure 5.4.3.1 Add Ingredient Module Page

The bottom navigator Refrigerator label controls the ingredient management module. It helps users track food ingredients inventory by category, quantity, and expiry dates. There are category tabs for users to filter items by type such as all, vegetables, fruits, meat, dairy and others. Users have 2 options to add their ingredients through manual typing the ingredients name, quantity, category and expiry date or just tap on the camera icon to capture the ingredients picture for recognition.

5.4.4 Nutrition Plan Module

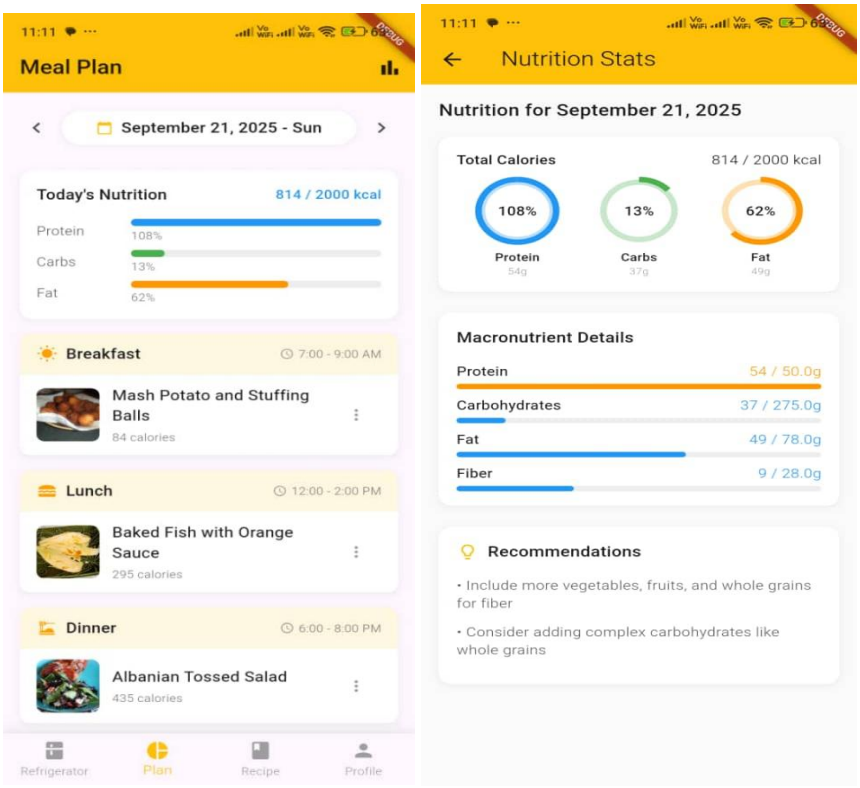


Figure 5.4.4.1 Nutrition Plan Module

The meal plan screen provides a comprehensive daily overview of nutrition and meal schedule. The top section displays the current date; it allows users to navigate between different days to view or plan meals. Users can tap on the right top icon to navigate to nutrition statistics page. It offers detailed analysis of your nutritional intake patterns and status. At the bottom of the statistics page contains nutrition recommendations which provide personalized nutrition advice based on users' consumption.

5.5 Trained Model Result

Figure 5.5.1 below shows the test accuracy of the train model with about 0.9275. The line graph shows that the validation accuracy increases steadily over time. This indicates that the model is learning and improving its performance on the validation set.

```
test_loss,test_acc = cnn.evaluate(test_set)
print('Test accuracy:', test_acc)
```

```
12/12 ━━━━━━━━━━━━━━━━━━━━ 63s 5s/step - accuracy: 0.9288 - loss: 0.6365
Test accuracy: 0.9275766015052795
```

Figure 5.5.1 Validation Test Accuracy

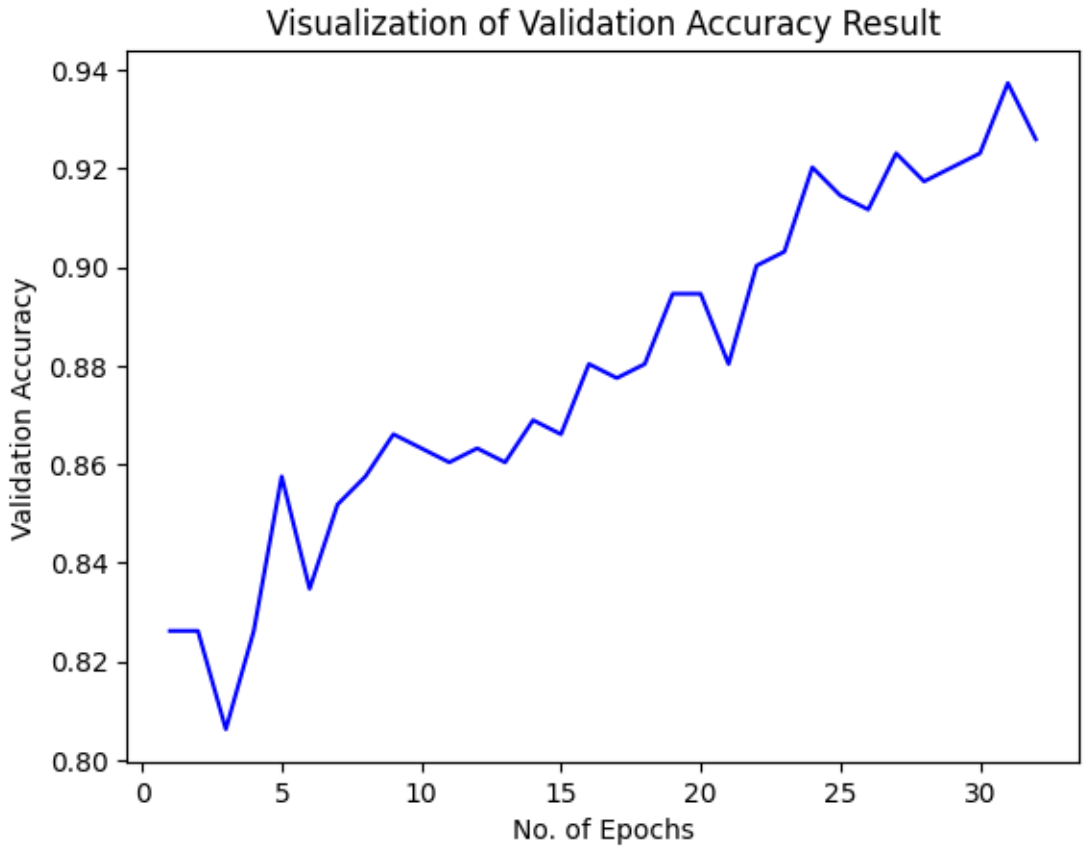


Figure 5.5.2 Visualization Line Graph of the Accuracy Result

5.5 Implementation Issues and Challenges

During implementation of the ingredient recognition module, there are significant challenges when transitioning from the model development environment to the mobile application deployment. The model has excelled accuracy performance about 0.93 during the development phase on Google Colab using the TensorFlow framework. Accuracy degradation happens when it implements to the flutter project. The situation frequently shows “Unknow” classifications for ingredients that were previously correctly identified and misclassified the ingredient into incorrect categories.

Several potential factors were identified through investigation. First, model conversion issues such as the conversion process from TensorFlow's .h5 format to TensorFlow Lite's .tflite format may introduce quantization errors. The large difference in archive size between the original model (~77MB) and the converted TFLite model (~25MB) suggests that aggressive quantization may hurt the representation power of the model. Therefore, this implementation issue highlights the huge gap that can exist between the performance of models in a controlled development environment and real-world deployment scenarios.

5.6 Concluding Remark

The system implementation phase marked a key milestone in transforming the initial design of a cooking assistant app with nutrition tracking functionality into a fully functional, interactive mobile application. This chapter systematically outlines the development process of each component, covering the integration of the user interface, local storage, machine learning capabilities, and supporting services.

The application successfully integrates essential features such as user account management, ingredient recognition, recipe browsing, nutrition tracking, and meal planning. These features are designed to provide users with a personalized and practical cooking experience while promoting healthier eating habits. To ensure offline access, the system uses a local SQLite database to store recipes, user profiles, and nutrition logs, improving data reliability and protecting user privacy without relying on cloud connectivity.

The system incorporates core innovations, including ingredient detection and automatic recipe recommendations using pre-trained models, to improve convenience and reduce food waste. Furthermore, a nutrition analysis module and interactive charts were developed to help users track their dietary goals. Implementing these features presented challenges, such as optimizing

model inference on mobile devices, ensuring smooth database operation, and managing UI responsiveness for dynamic data. These challenges were addressed through iterative testing, code improvements, and meticulous system optimization.

In summary, this chapter not only documented the application technical foundation but also highlighted the problem-solving strategies employed during the development phase. The successful implementation of the cooking assistant and nutrition tracking app laid a solid foundation for further evaluation, user testing, and refinement, which will be explored in the next chapter.

CHAPTER 6
SYSTEM EVALUATION AND DISCUSSION

6.1 System Testing and Performance Metrics

System testing is a critical step in verifying that a cooking assistant app with nutrition tracking meets both functional and non-functional requirements. Extensive testing is conducted to verify the correctness of core features such as ingredient detection, recipe browsing, nutrition log updates, and database reliability. Additionally, application usability, performance, and stability are evaluated under various operating conditions.

Because the application is designed for offline use, all testing was conducted in a standalone environment using both simulators and physical devices. The tests simulated real-world usage scenarios, including scanning ingredients, creating recipes, tracking nutrition intake, and browsing saved recipes.

6.1.1 Testing Approach

The following methodologies were adopted:

Manual Functional Testing

- Each feature was manually tested using real-life cases:
- Adding, editing, and deleting ingredients and recipes.
- Capturing ingredient images and verifying model detection results.
- Generating nutritional logs and comparing them with predefined goals.
- Saving and retrieving data from the local SQLite database.

Unit Testing

Core logic such as nutrition calculation, ingredient-to-recipe matching, and goal tracking were modularized and tested independently using Flutter’s built-in test framework.

Integration Testing

- End-to-end workflows were tested to ensure:
- Ingredient scans correctly updated the recipe suggestion module.
- Nutritional data from recipes was logged into the database.
- Recipe saving and retrieval worked seamlessly with the UI.

UI/UX Testing

- Verified layout responsiveness across multiple screen sizes and orientations.
- Ensured smooth navigation through bottom navigation and tab views.
- Tested form inputs, dropdowns, and search functionality in the recipe screens.

Performance Testing

- Measured app cold and warm start times.
- Evaluated model inference time for ingredient detection.
- Assessed database query performance with increasing data volume.
- Checked memory and CPU usage during simultaneous scanning and logging.

6.1.2 Performance Metrics and Results

Table 6.1.2.1 Performance Metrics and Results

Metric	Target Value	Observed Result	Status
App Initial Launch Time	< 2 seconds	~1.2 seconds	Passed
Warm Start (Resume)	< 1.5 seconds	< 1 second	Passed
Ingredient Detection Time	< 5 seconds	3 seconds	Passed
Recipe Recommendation Delay	< 2 seconds	< 1 second	Passed
Nutritional Log Update Time	< 1 second	< 1 second	Passed
Detection Accuracy	≥ 80%	~80%	Passed
Chart Rendering (Nutrition UI)	< 2 seconds	~1 second	Passed
App Crashes	0	None observed	Passed

6.1.3 Usability Observations

1. Navigation and Layout: All major functions such as scan, recipes, nutrition logs were reachable within two taps from the home screen.
2. Manual Adjustments: After detection, users could manually adjust ingredients and nutritional data to correct mismatches.
3. Data Visualization: Charts and progress bars for nutritional goals provided intuitive insights for daily tracking.

6.1.4 Limitations and Constraints

- Device-Specific Detection Accuracy: Lower-end devices with weaker cameras reduce detection models' ingredient detection reliability.
- Database Lag with Large Data: When more than 1000 recipe records were stored, minor delays were observed in search queries.
- Single Device Limitation: As an offline-first system, cloud sync and backup features were not supported.
- Model Context Limitation: The detection model worked best with common ingredients but was less accurate with rare or poorly captured items.

6.1.5 Summary

Overall, the application passed all major testing benchmarks. Core modules such as ingredient detection, recipe plan generation, and nutritional logging worked reliably under offline conditions. The detection model integration provided effective real-time detection, and the SQLite database ensured consistent data persistence. These results validate that the system is ready for real-world usage, with future improvements focusing on cloud support and enhanced AI features.

6.2 Testing Setup and Result

6.2.1 Testing Environment Overview

Table 6.2.1.1 Testing Environment Overview

Component	**Specification / Tool Used**
IDE	Android Studio Giraffe (2022.3.1)
Framework	Flutter SDK 3.19.5

Testing Tools	Flutter Test, Manual Testing
Database Inspection	Android Studio Inspector, DB Browser for SQLite
Version Control	GitHub (main branch)

6.2.2 Emulator and Device Setup

Table 6.2.2.1 Device Setup

Device Type	**Details**
Physical	Redmi2201117TG Snapdragon 680 1.0.7.0 TGCMIXM 6.0 GB RAM 128GB

Physical Device Goals:

- UI layout, data persistence, and database interactions.
- Camera testing for detection, performance of nutrition logging, and app responsiveness.

6.2.3 Test Data Setup

Table 6.2.3.1 Test Data Setup

Data Type	**Sample Values**
Ingredients	Tomato, Chicken, Onion, Garlic, Rice
Recipes	Chicken Fried Rice, Tomato Soup, Grilled Chicken Salad
Nutrition Logs	Daily calories, protein, and carb tracking entries
Detection Images	10 ingredient photos captured under varying lighting conditions

6.2.4 Functional Testing Results

Table 6.2.4.1 Functional Testing Results

Feature	**Expected Outcome**	**Result**	**Remarks**
App Launch	Dashboard loads without crash	Passed	Smooth

Ingredient Scanning	Ingredients detected and displayed	Passed	Minor edits for unclear images
Nutritional Log Updates	Logs reflect recipe nutritional info	Passed	Synced instantly
Database Storage	Data saved and retrieved correctly	Passed	Stable
Chart Generation	Nutrition charts render correctly	Passed	No null crashes
Offline Usability	All features usable offline	Passed	Fully functional

6.2.5 Key Observations

1. Ingredient scanning accuracy improved under good lighting conditions.
2. Database queries required indexing for faster retrieval when handling large recipe sets.
3. Manual correction support was essential for ensuring data accuracy after detection.

6.2.6 Summary

The application's functional and non-functional aspects were validated successfully across emulator and physical devices. SQLite and Flutter provided stable offline-first architecture, and detection models ensured real-time ingredient detection. Testing confirmed the system's usability and efficiency.

6.3 Project Challenges

1. Real-Time Ingredient Detection

Challenge: Ensuring detection model worked smoothly on mobile without heavy delays.

Resolution: Optimized inference pipeline with model conversion and resized input images for faster predictions.

2. Database Management for Recipes and Logs

Challenge: Storing recipes plan, nutritional data, and user logs without causing redundancy or lag.

Resolution: Normalized SQLite schema with indexing for faster queries and efficient storage.

3. UI Responsiveness Across Screens

Challenge: Maintaining consistent layout for recipes, charts, and logs across devices.

Resolution: Applied Flutter's adaptive layout widgets such as media query, expanded, flexible for smooth scaling.

4. OCR and Nutrition Data Extraction

Challenge: Extracting accurate nutrition details from detected ingredients and recipes.

Resolution: Implemented structured mapping between ingredients and nutrition tables, with fallback manual entry.

5. Permission and Camera Handling

Challenge: Ensuring smooth camera access and runtime permissions across Android versions.

Resolution: Integrated permission handler and developed fallback messages for denied access.

6. Time and Scope Limitations

Challenge: Advanced features such as cloud sync and multi-device support were beyond the timeline.

Resolution: Focused on building and refining the offline-first core features first, while leaving room for future expansion.

6.4 Objectives Evaluation

Table 6.4.1 Objective 1 and 2 Evaluation

Objective 1	**Evaluation**
Allow users to scan and manage ingredients	Achieved. Users can scan ingredients using trained models and manually adjust results.
Support manual input and correction	Achieved. Users can edit detected data for improved accuracy.
Objective 2	**Evaluation**
Enable nutritional tracking and goal setting	Achieved. Users can log nutrition intake and compare against daily goals.
Visualize nutrition progress with charts	Achieved. Progress is displayed through dynamic bar and pie charts.

6.5 Concluding Remark

All core objectives were successfully achieved. While advanced enhancements such as cloud-based backups and improved AI-powered nutrition predictions remain future work, the project delivered a complete, functional, and extensible Cooking Assistant with Nutritional Tracking App.

CHAPTER 7

CONCLUSION AND RECOMMENDATION

7.1 Conclusion

In today's fast-paced lifestyle, maintaining a healthy eating habit is a common challenge due to time constraints, lack of nutritional awareness, and inefficient kitchen management. The project addresses these issues by proposing a smart mobile application that can help users manage ingredients through image recognition, plan meals according to dietary needs, and track nutritional intake through detailed analysis. The app aims to reduce food waste, encourage healthier eating and provide support for individuals with complex dietary needs by providing personalized meal recommendations and real-time stock alerts.

The app integrates ingredient management, nutritional analysis, and meal planning to help promote sustainable eating habits and improve users' health. It enables users to make informed eating decisions, reduce unnecessary food purchases, and manage kitchen supplies efficiently. Ultimately, cooking assistant apps offer practical and innovative solutions to modern cooking and nutrition challenges so that users can make their daily meal preparation smarter, healthier, and more efficient.

7.2 Recommendation

Although the core objectives of the Cooking Assistant with Nutritional Tracking application have been achieved, several areas remain open for improvement and future enhancement. These directions aim to expand the functionality, improve user experience, and increase the system's adaptability.

7.2.1 Cloud Synchronization and Multi-Device Accessibility

Currently, the application stores all data locally within the SQLite database. Future iterations could integrate cloud-based solutions such as Firebase Firestore or Google Drive to enable seamless synchronization across multiple devices. This would not only facilitate cross-device accessibility but also provide reliable backup and recovery options in the event of device loss or replacement.

7.2.2 Enhanced AI-Powered Ingredient Recognition and Recommendation

While the application currently supports ingredient recognition and recipe suggestions, future enhancements could leverage more advanced machine learning models for improved detection accuracy. Additionally, predictive algorithms could be introduced to recommend meals based on dietary patterns, nutritional goals, or ingredient expiry tracking, thereby increasing personalization and reducing food waste.

7.2.3 Voice Assistant and Hands-Free Cooking Support

Future versions of the system could incorporate a more sophisticated voice assistant to support hand-free interactions during cooking. This would allow users to navigate through recipes, check nutritional values, and follow step-by-step instructions without the need to physically interact with the device, thereby improving convenience in a kitchen environment.

7.2.4 Multi-Language and Localization Support

To broaden accessibility, localization features such as multi-language support and region-specific measurement units could be incorporated. This enhancement would enable the system to cater to users from diverse linguistic and cultural backgrounds, making the application more adaptable and globally relevant.

7.2.5 Integration with Wearable and IoT Devices

Future development could extend the system by connecting it with wearable health devices such as smartwatches, fitness trackers or IoT-enabled kitchen appliances. This kind of integration would allow real-time monitoring of calorie expenditure, automated logging of

nutritional intake, and smart kitchen interactions, creating a more holistic health and cooking ecosystem.

In summary, these proposed improvements would further strengthen the system's capabilities, improve scalability, and enhance overall usability. With continued refinement, the application has the potential to evolve into a comprehensive digital assistant that supports both cooking convenience and long-term nutritional well-being.

REFERENCES

[1] Angelsen, A., Starke, A.D. & Trattner, "Healthiness and environmental impact of dinner recipes vary widely across developed countries," *Nature Food,* vol. 4, no. 5, pp. 407-415, 1 May 2023.

[2] WHO, "Salt Reduction," World Health Organization: WHO, 2020. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/salt-reduction.

[3] Y. Matsushima, N. Funabiki, T. Okada, T. Nakanishi and K. Watanabe, "A cooking guidance function on Android tablet for Homemade Cooking Assistance System," *IEEE Region 10 Humanitarian Technology Conference,* pp. 249-254, 2013.

[4] Shyam, S., Lee, K. X., Tan, A. S. W., Khoo, T. A., Harikrishnan, S., Lalani, S. A., & Ramadas, A, "Effect of Personalized Nutrition on Dietary, Physical Activity, and Health," *Nutrients,* vol. 14, no. 19, p. 4104, 2022.

[5] H. Aldawood, M. Alabadi, O. Alharbi and G. Skinner, "A Contemporary Review of Raising Health Awareness Using ICT for Application in the Cyber Security Domain," pp. 1-8, 2019.

[6] R. Ashar, S. Lewis, D. L. Blazes, and J.-P. Chretien , "Applying information and communications technologies to collect health data from remote settings: a systematic assessment of current technologies," *Journal of Biomedical Informatics,* vol. 43, no. 2, pp. 332-341, 2010.

[7] Floridi, L., Cowls, J., Beltrametti, M. et al., "AI4People—An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations," *Minds and Machines,* vol. 28, no. 4, pp. 689-707, 1 December 2018.

[8] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang, and J. P. Campbell, "Introduction to Machine Learning, Neural Networks, and Deep Learning," *Translational Vision Science & Technology,* vol. 9, no. 2, pp. 14-14, 2020.

[9] M. M. Hossain, "Fast Food Consumption and its Impact on Health," *ResearchGate,* January 2020.

[10] I. Coding Minds, "Smart Kitchen Assistant," 15 7 2024. [Online]. Available: https://apps.apple.com/us/app/smart-kitchen-assistant/id6544808437.

[11] RellowTeam, "CookAI: Recipes Assistant," 2021. [Online]. Available: https://play.google.com/store/apps/details?id=com.rellowteam.cookai&hl=en_US.

[12] PixFood, "PixFood - Take a Photo! Get Tasty Recipes!," [Online]. Available: https://pixfood.com/.

[13] SideChef, "SideChef: Recipes & Meal Plans," 2021. [Online]. Available: https://play.google.com/store/apps/details/?id=com.sidechef.sidechef&hl=en_US.

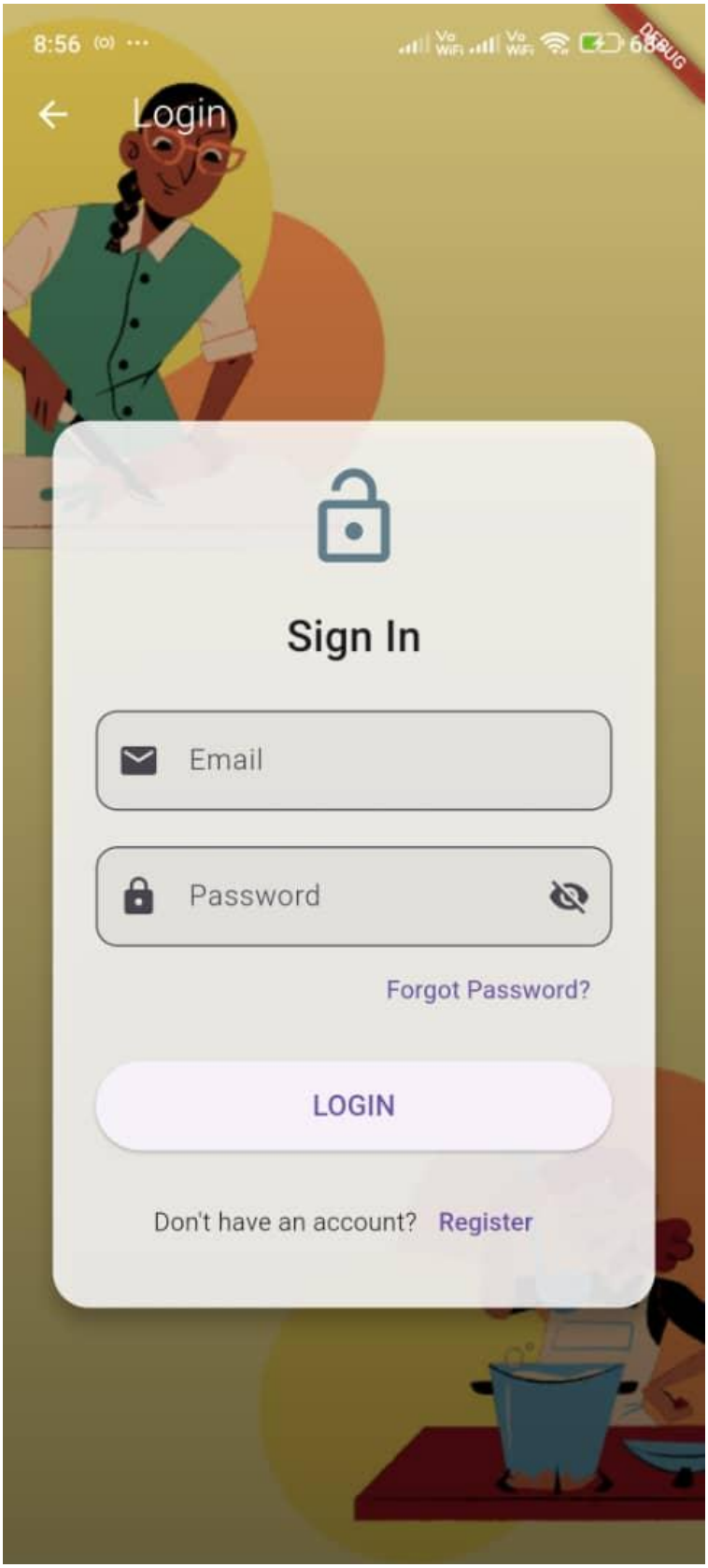
[14] "SuperCook Recipe By Ingredient," [Online]. Available: https://apps.apple.com/us/app/supercook-recipe-by-ingredient/id1477747816.

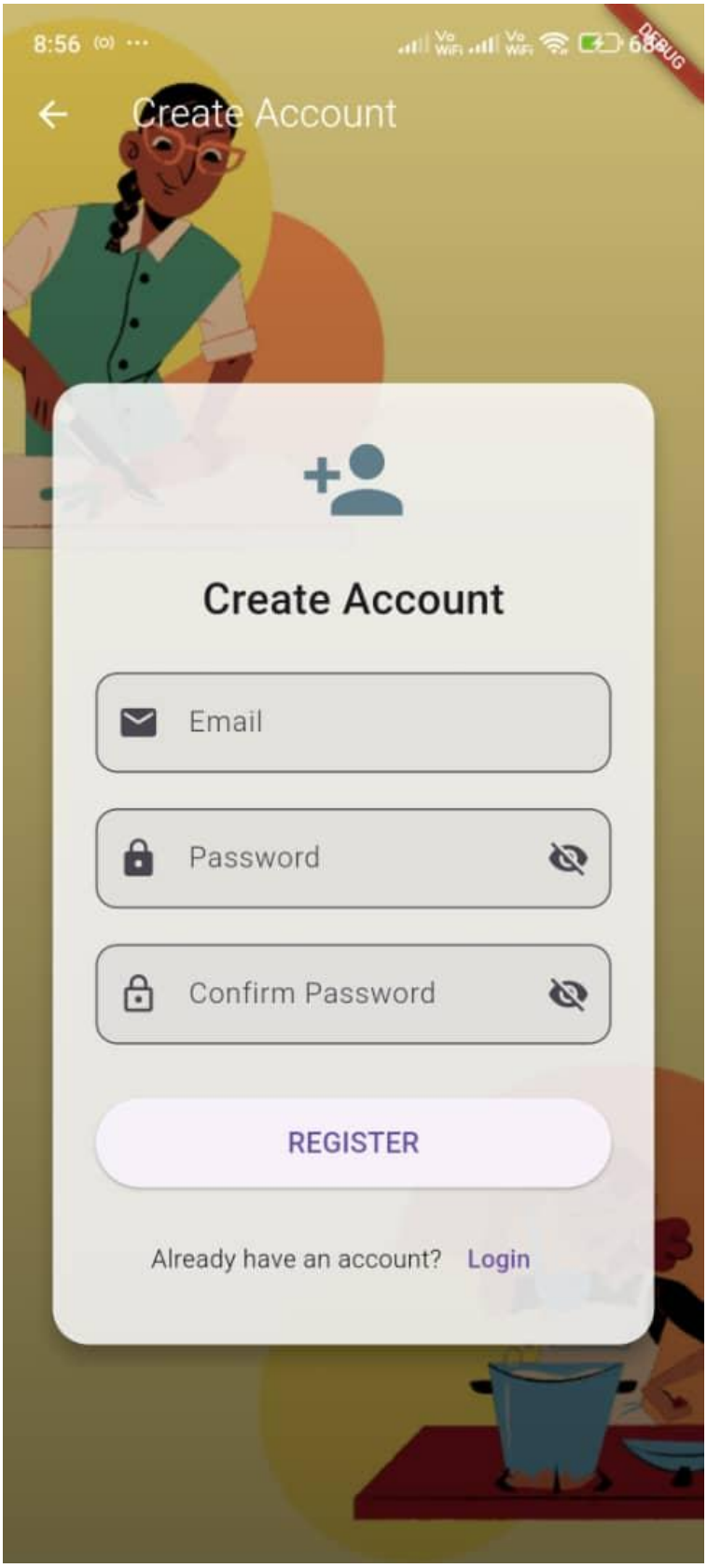
[15] I. Sommerville, Software engineering, 5th ed., Addison Wesley Longman Publishing Co., Inc., 1995.

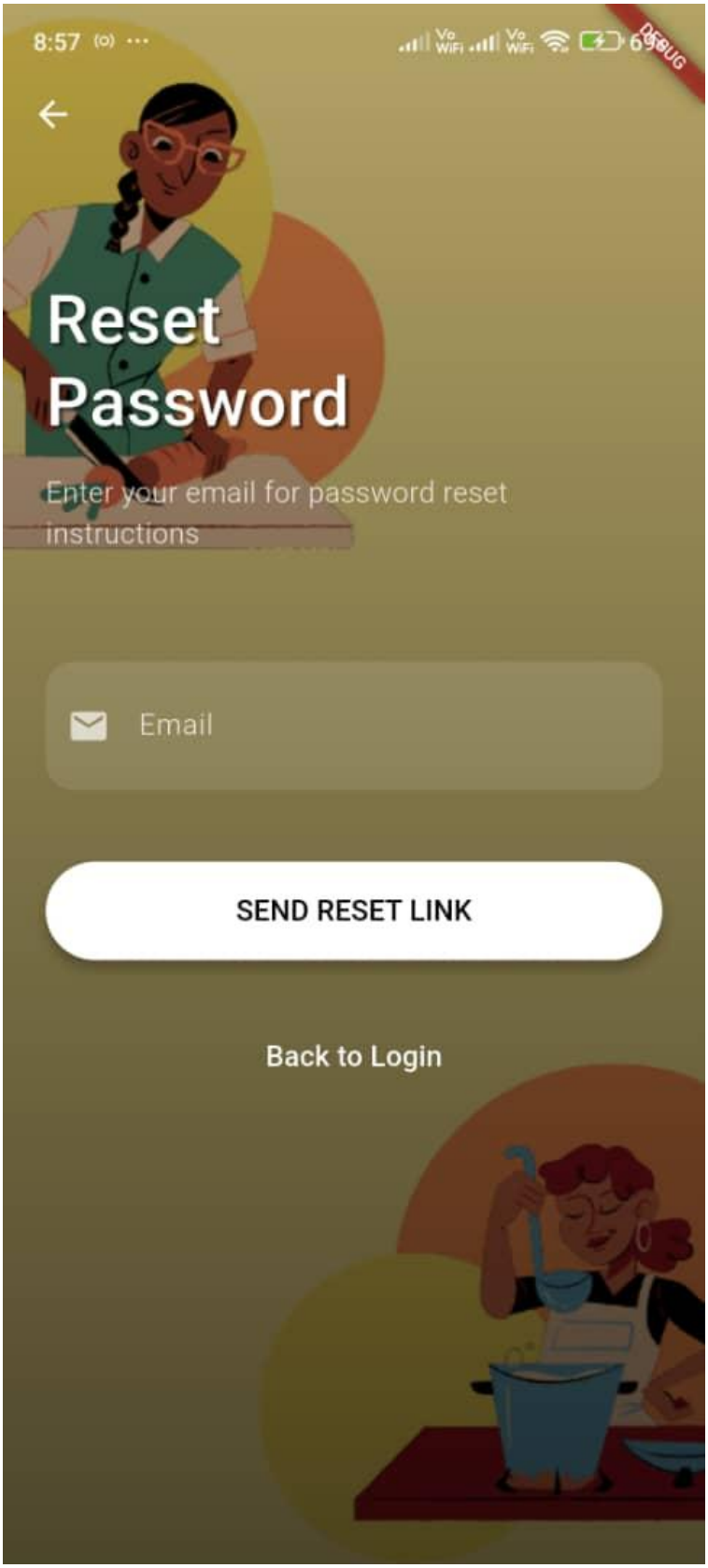
[16] M. Kasbe, "What is Software Development Life-Cycle(SDLC) & what are agile methodologies !?," 26 April 2023. [Online]. Available: https://maheshkasbe.hashnode.dev/what-is-software-development-life-cyclesdlc-what-are-agile-methodologies#heading-scrum.

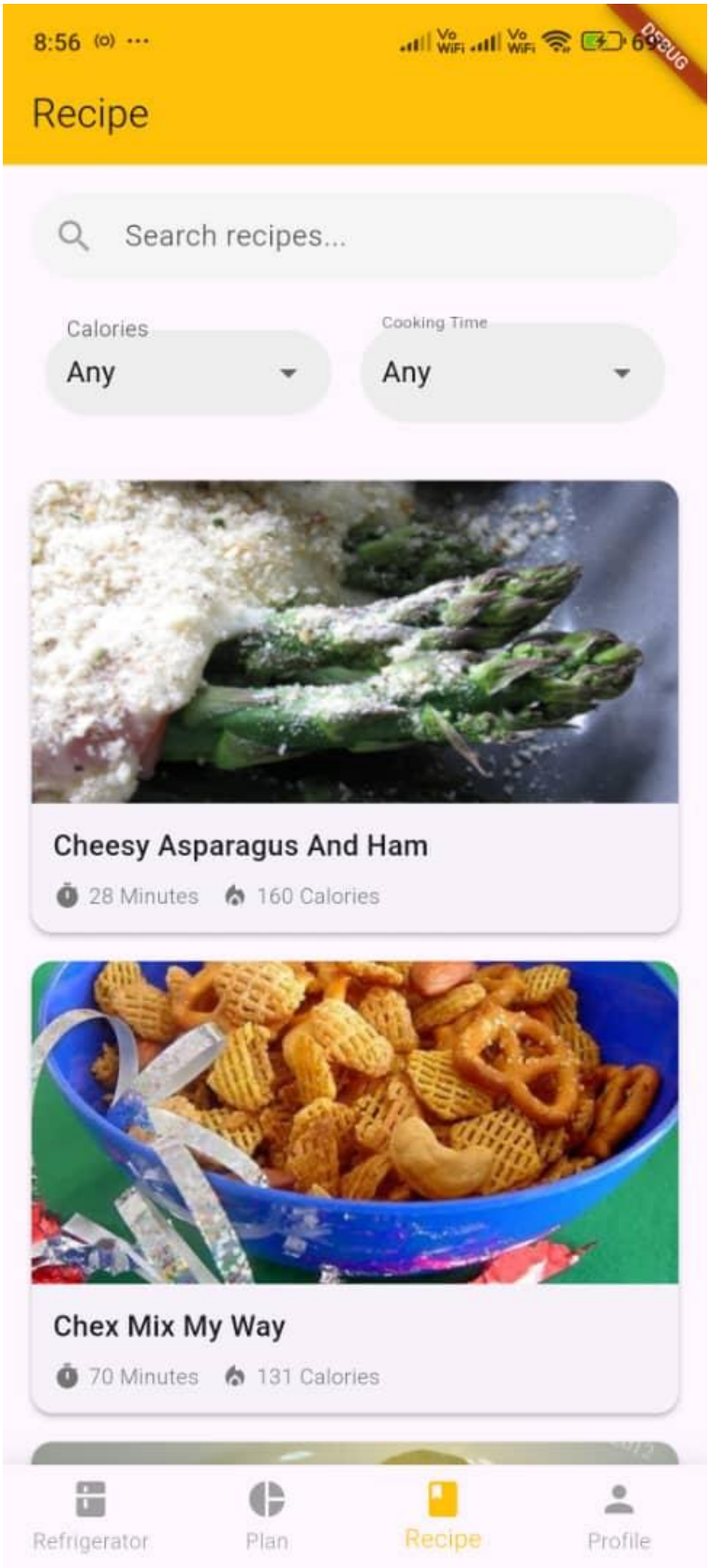
APPENDIX

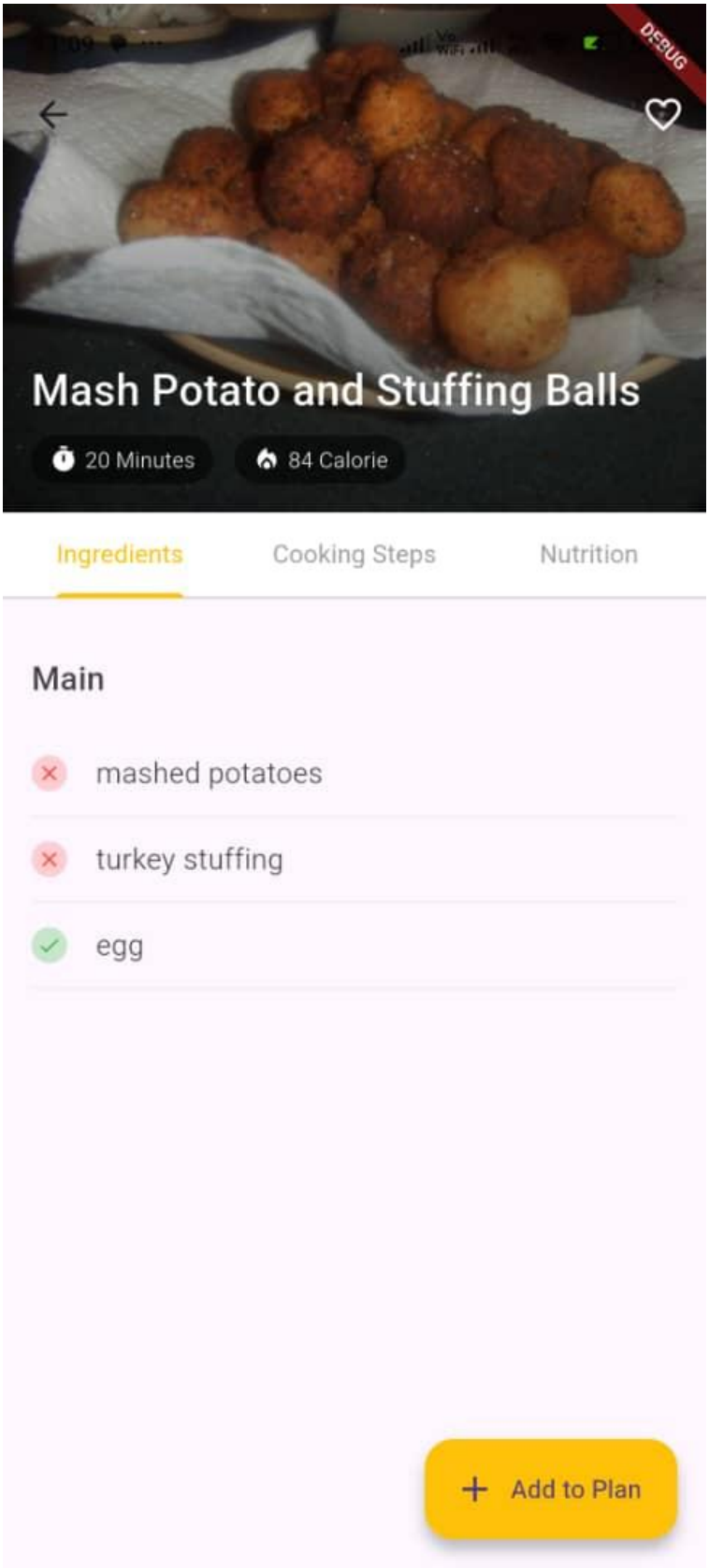
User Interface

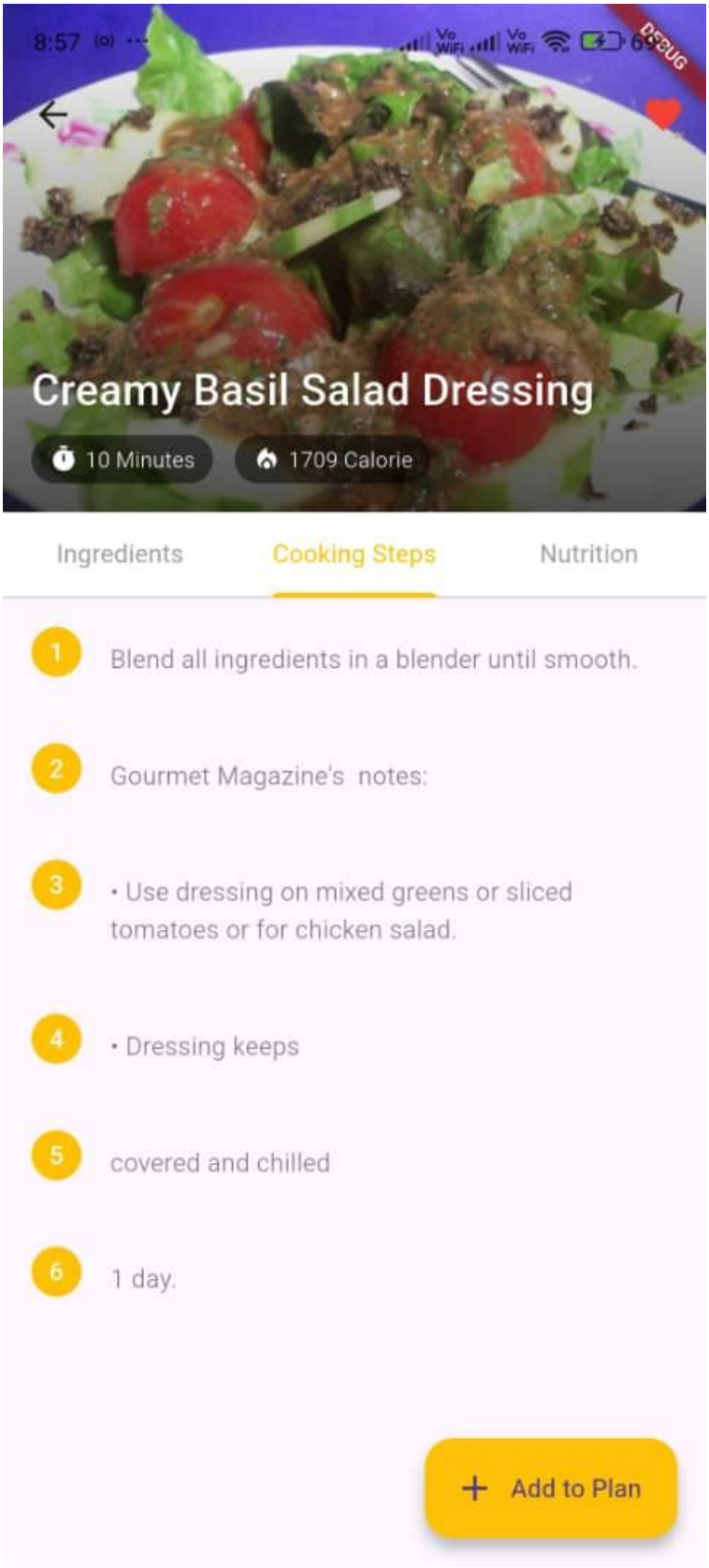

8:56
Login
DEBUG
Sign In
Email
Password
Forgot Password?
LOGIN
Don't have an account? Register

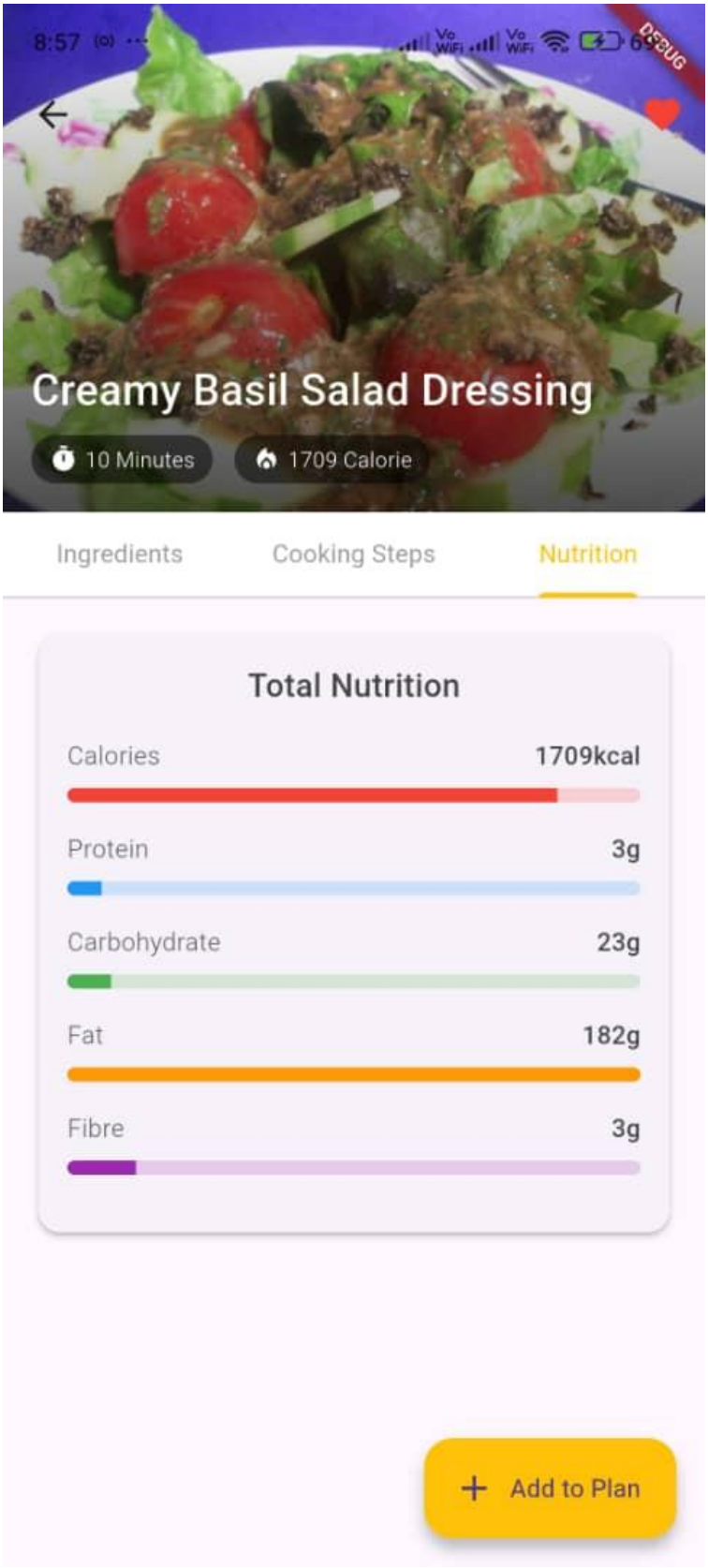

8:56
DEBUG
Create Account
Create Account
Email
Password
Confirm Password
REGISTER
Already have an account?
Login

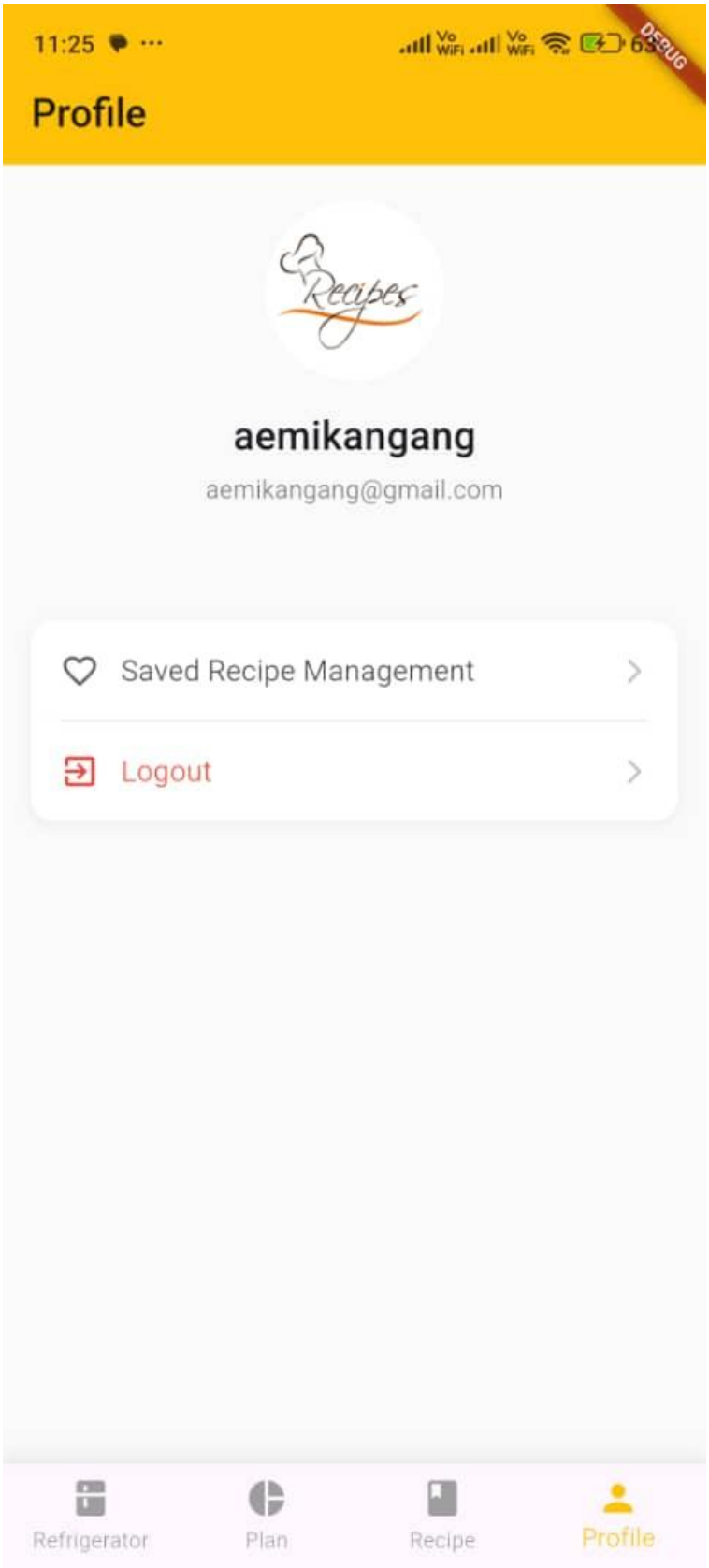

8:57
DEBUG
Reset Password
Enter your email for password reset instructions
Email
SEND RESET LINK
Back to Login

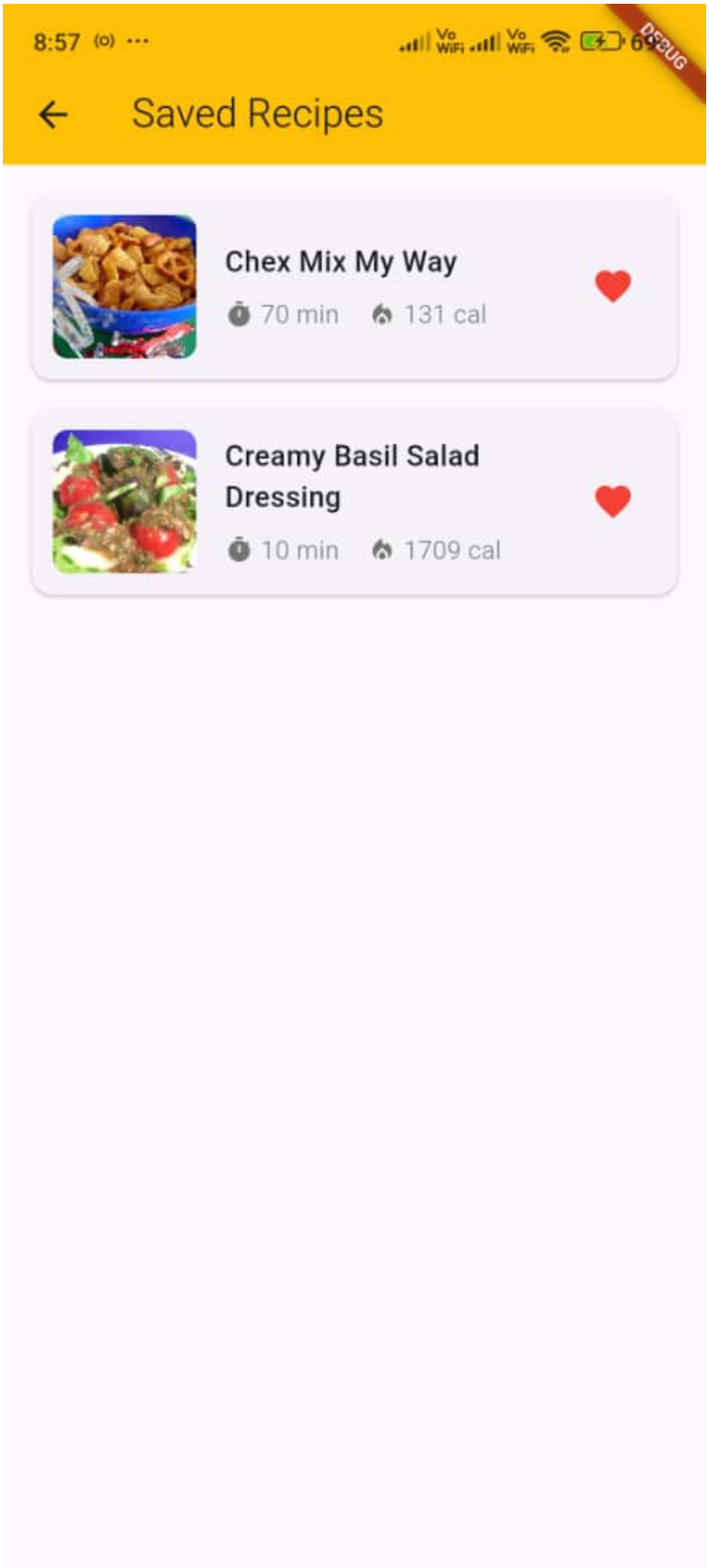

8:56
DEBUG
Recipe
Search recipes...
Calories
Any
Cooking Time
Any
Cheesy Asparagus And Ham
28 Minutes
160 Calories
Chex Mix My Way
70 Minutes
131 Calories
Refrigerator
Plan
Recipe
Profile

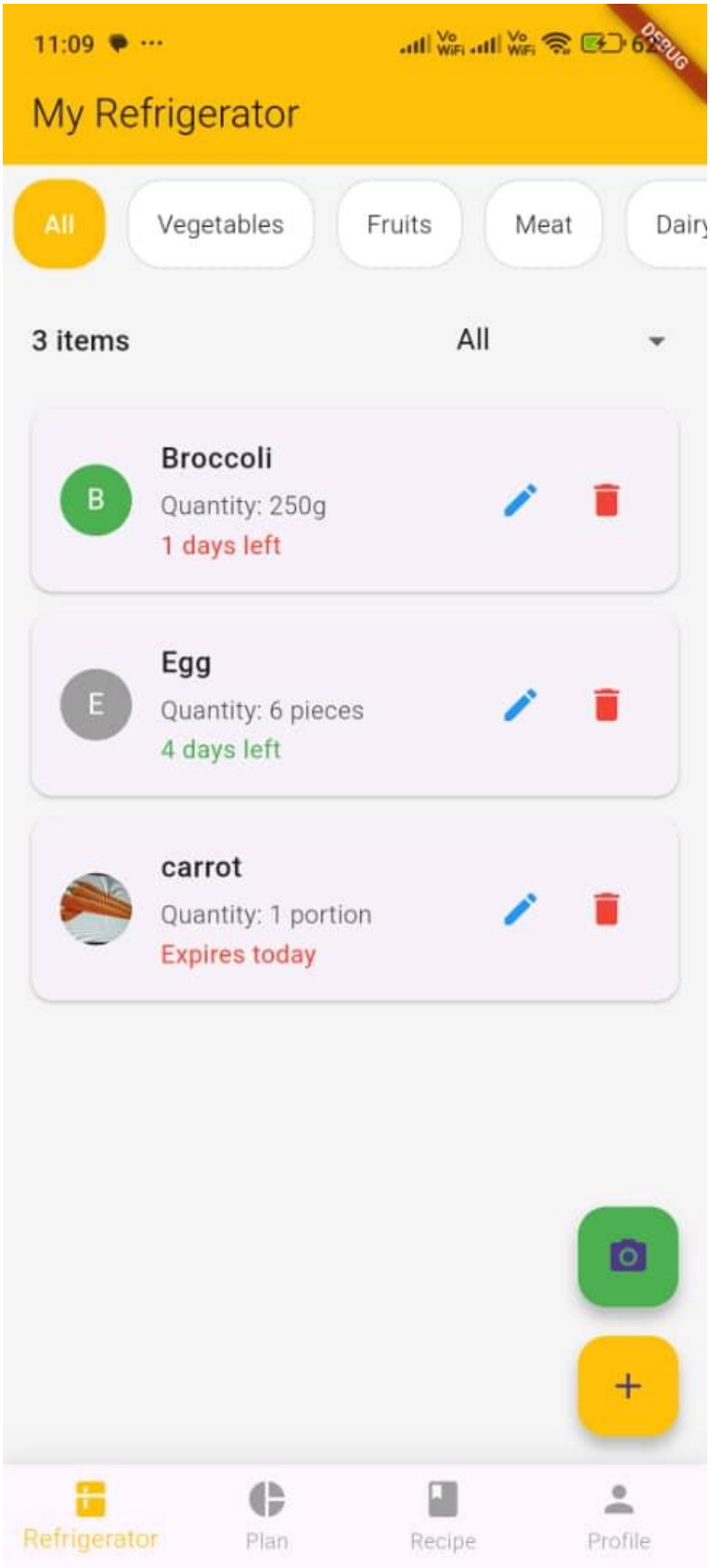

DEBUG
Mash Potato and Stuffing Balls
20 Minutes
84 Calorie
Ingredients
Cooking Steps
Nutrition
Main
mashed potatoes
turkey stuffing
egg
Add to Plan

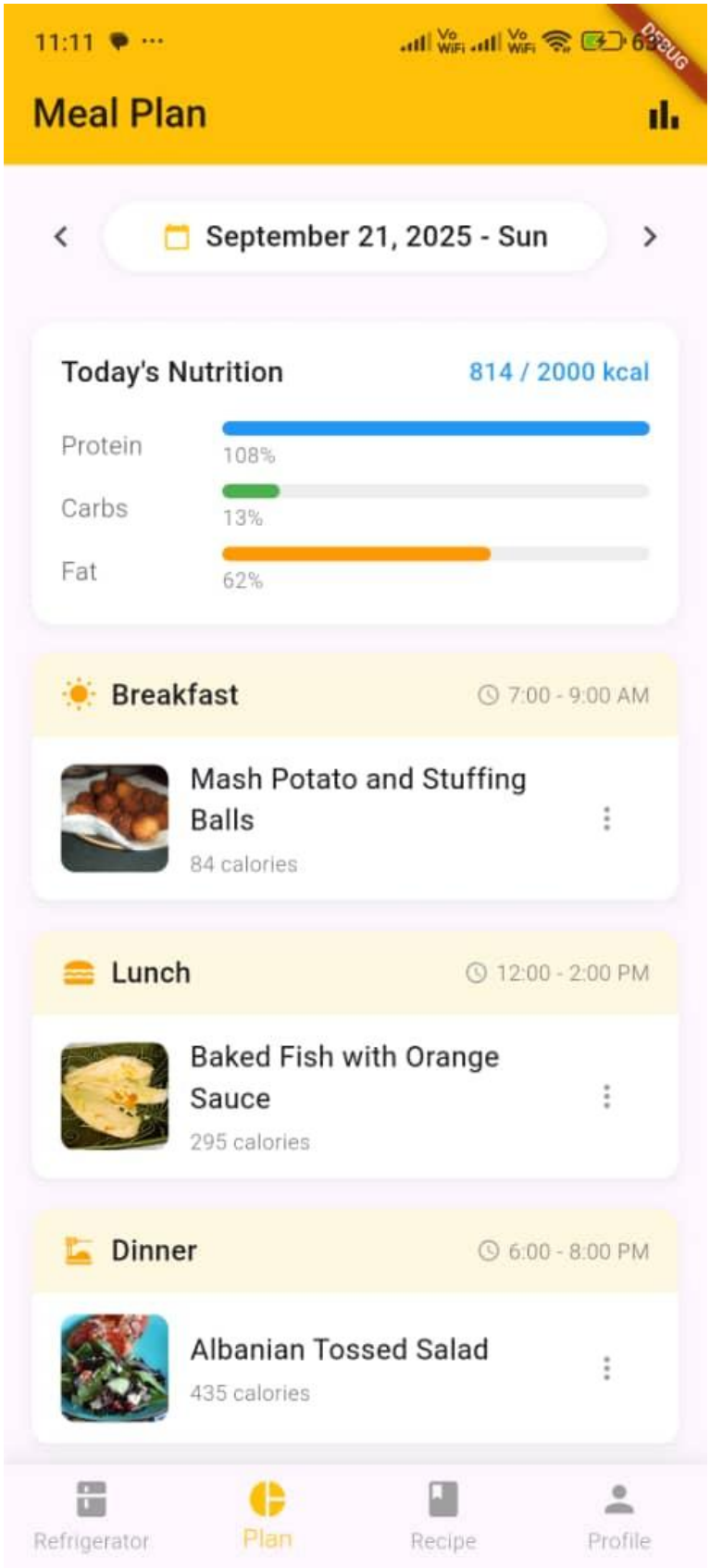

8:57
DEBUG
Creamy Basil Salad Dressing
10 Minutes
1709 Calorie
Ingredients
Cooking Steps
Nutrition
1
Blend all ingredients in a blender until smooth.
2
Gourmet Magazine's notes:
3
• Use dressing on mixed greens or sliced tomatoes or for chicken salad.
4
• Dressing keeps
5
covered and chilled
6
1 day.
Add to Plan

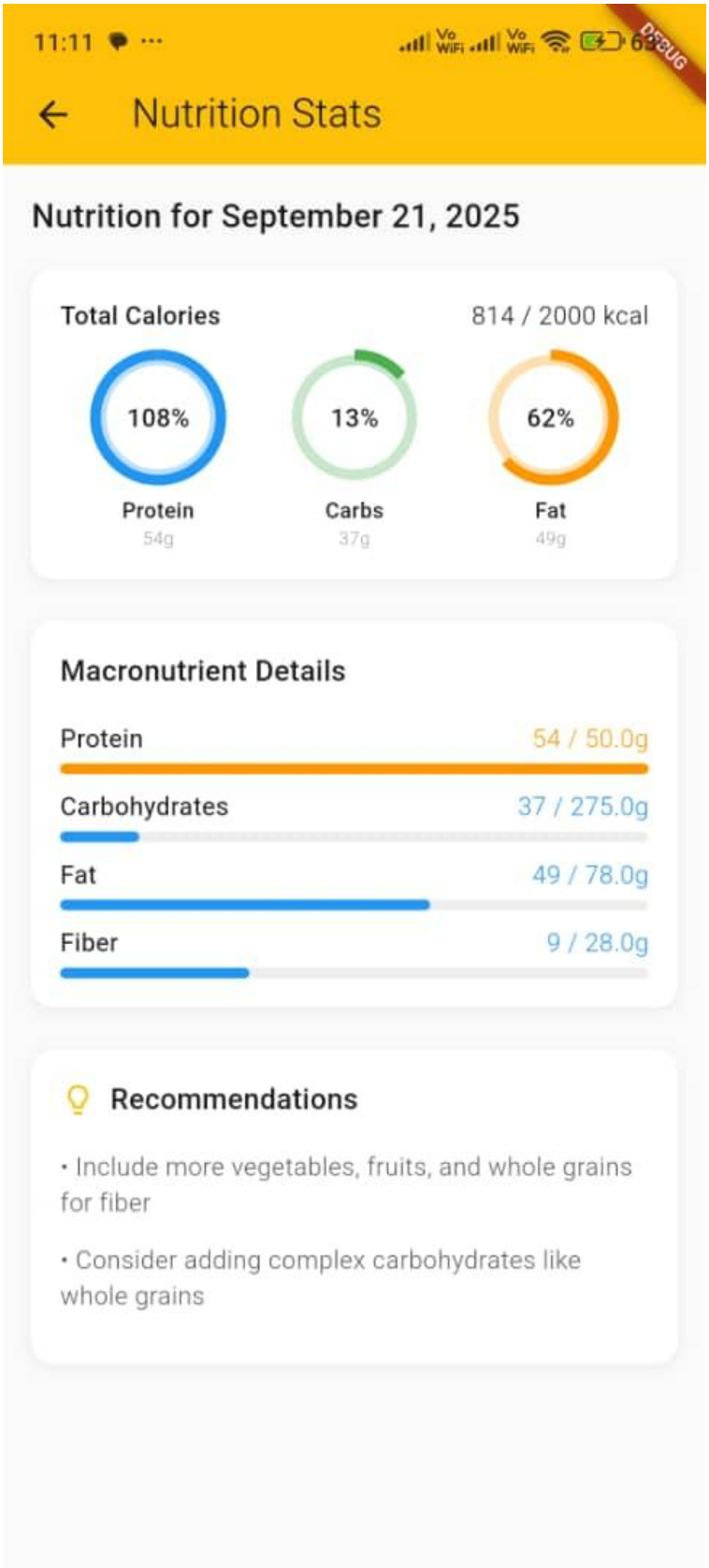

8:57
DEBUG
Creamy Basil Salad Dressing
10 Minutes
1709 Calorie
Ingredients
Cooking Steps
Nutrition
Total Nutrition
Calories
1709kcal
Protein
3g
Carbohydrate
23g
Fat
182g
Fibre
3g
Add to Plan


11:25
DEBUG
Profile
Recipes
aemikangang
aemikangang@gmail.com
Saved Recipe Management
Logout
Refrigerator
Plan
Recipe
Profile


8:57
DEBUG
Saved Recipes
Chex Mix My Way
70 min
131 cal
Creamy Basil Salad Dressing
10 min
1709 cal


11:09
DEBUG
My Refrigerator
All
Vegetables
Fruits
Meat
Dairy
3 items
All
B
Broccoli
Quantity: 250g
1 days left
E
Egg
Quantity: 6 pieces
4 days left
carrot
Quantity: 1 portion
Expires today
Refrigerator
Plan
Recipe
Profile


11:11
DEBUG
Meal Plan
September 21, 2025 - Sun
Today's Nutrition
814 / 2000 kcal
Protein
108%
Carbs
13%
Fat
62%
Breakfast
7:00 - 9:00 AM
Mash Potato and Stuffing Balls
84 calories
Lunch
12:00 - 2:00 PM
Baked Fish with Orange Sauce
295 calories
Dinner
6:00 - 8:00 PM
Albanian Tossed Salad
435 calories
Refrigerator
Plan
Recipe
Profile


11:11
DEBUG
Nutrition Stats
Nutrition for September 21, 2025
Total Calories
814 / 2000 kcal
108%
Protein
54g
13%
Carbs
37g
62%
Fat
49g
Macronutrient Details
Protein
54 / 50.0g
Carbohydrates
37 / 275.0g
Fat
49 / 78.0g
Fiber
9 / 28.0g
Recommendations
• Include more vegetables, fruits, and whole grains for fiber
• Consider adding complex carbohydrates like whole grains

Poster

Cooking Assistant with Nutritional Tracking App

Introduction:

In modern busy lifestyles, people often struggle to prepare healthy meals, manage ingredients, and track nutrition effectively. Manual meal planning and nutrition tracking are time-consuming and error-prone, leading to poor diet quality and food waste. This project proposes a smart mobile application that assists users in identifying ingredients using image recognition, managing kitchen inventory, and planning meals with personalized nutritional tracking. The app aims to improve health, reduce food waste, and support dietary needs through a user-friendly digital solution.

Discussion:

The project tackles several real-world challenges such as food waste, lack of nutritional knowledge, and limited time for meal preparation. Compared to existing apps, this system offers a more integrated and intelligent solution that considers ingredient availability and personalized dietary needs. The image-based ingredient input system increases convenience, while the nutrition analysis tool raises user awareness and supports healthier choices.

Conclusion:

The Cooking Assistant with Nutritional Tracking App helps users manage ingredients more effectively, reduce food waste, and follow personalized nutrition plans. Through the use of image recognition and real-time inventory monitoring, it simplifies daily meal preparation while promoting healthier eating habits. The project demonstrates how technology can be applied to support sustainable, efficient, and informed food management in life.

Methods

1. Ingredient Detection Module
   - Image recognition identifies the ingredient
   - Detected items are saved to the inventory
2. Manage Food Ingredient
   - Tracks current ingredients
3. Personalized Meal Planning with Suggestion
   - View & Get nutritional Analysis

Done By: Ngang Chi Khim
Supervised By: Ts Dr Lim Seng Poh