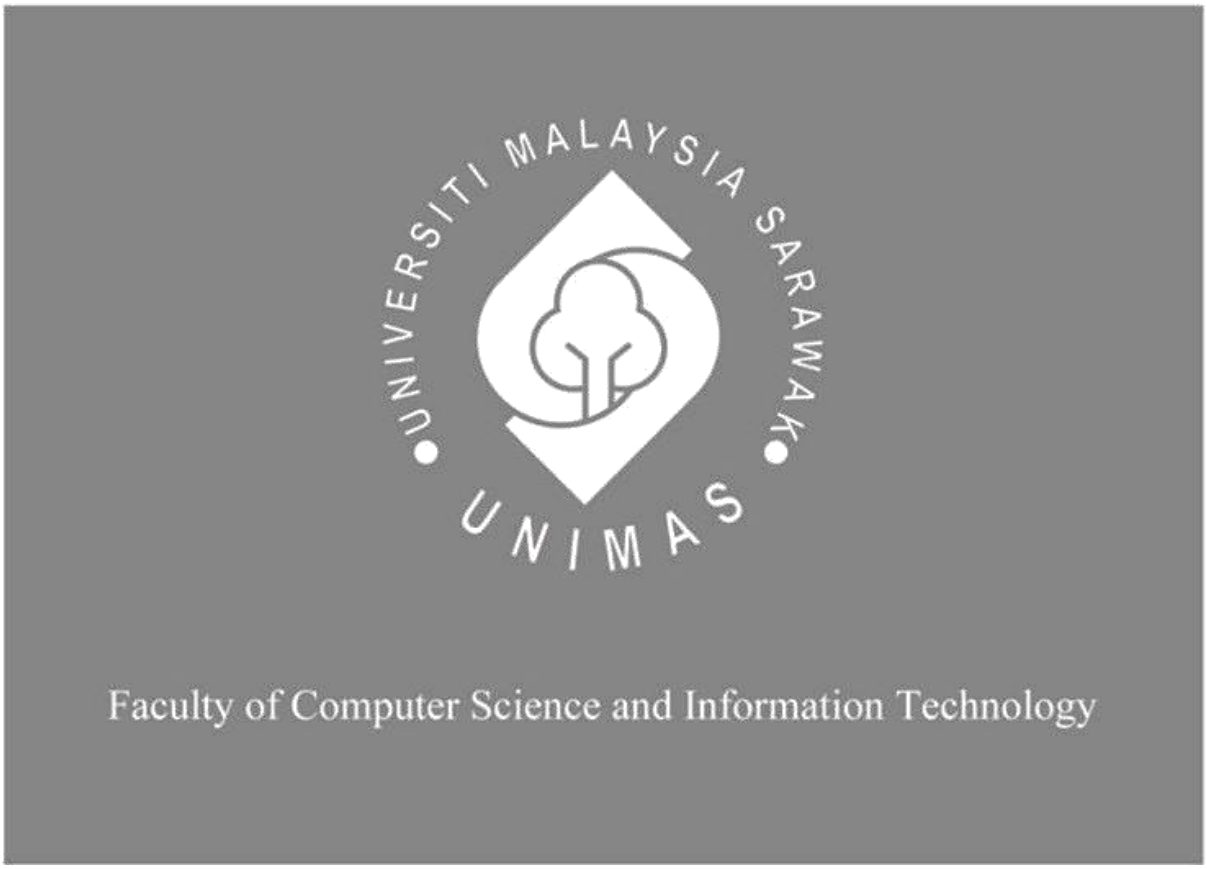


SMART RECIPE GENERATOR: A DEEP LEARNING-POWERED MOBILE APP FOR PERSONALIZED AND WASTE-REDUCING FOOD

Leslie Kong Hao Ong

Bachelor of Computer Science with Honours (Network Computing)

2025

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

TITLE **SMART RECIPE GENERATOR: A DEEP LEARNING-POWERED MOBILE APP FOR PERSONALIZED AND WASTE-REDUCING FOOD**

ACADEMIC SESSION: **2024/2025**

LESLIE KONG HAO ONG

(CAPITAL LETTERS)

hereby agree that this Thesis* shall be kept at the Centre for Academic Information Services, Universiti Malaysia Sarawak, subject to the following terms and conditions:

1. The Thesis is solely owned by Universiti Malaysia Sarawak
2. The Centre for Academic Information Services is given full rights to produce copies for educational purposes only
3. The Centre for Academic Information Services is given full rights to do digitization in order to develop local content database
4. The Centre for Academic Information Services is given full rights to produce copies of this Thesis as part of its exchange item program between Higher Learning Institutions [or for the purpose of interlibrary loan between HLI]
5. ** Please tick (√)

☐	CONFIDENTIAL	(Contains classified information bounded by the OFFICIAL SECRETS ACT 1972)
☐	RESTRICTED	(Contains restricted information as dictated by the body or organization where the research was conducted)
☑	UNRESTRICTED	

Validated by

Assoc. Prof. Dr. Johari Abdullah
Faculty of Computer Science & Information Technology
Universiti Malaysia Sarawak
94300 Kota Samarahan

(AUTHOR'S SIGNATURE)

(SUPERVISOR'S SIGNATURE)

Permanent Address

NO 2J, LORONG 5A2
JALAN KIEW NANG
96000 SIBU, SARAWAK

Date: 18 JUNE 2025

Date: 18 JUNE 2025

SMART RECIPE GENERATOR: A DEEP LEARNING-POWERED MOBILE APP FOR PERSONALIZED AND WASTE-REDUCING FOOD

LESLIE KONG HAO ONG

This project is submitted in partial fulfilment of the
requirements for the degree of
Bachelor of Computer Science with Honours (Network Computing)

Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK
2025

PENJANA RESIPI PINTAR: APLIKASI MUDAH ALIH BERKUASA PEMBELAJARAN MENDALAM UNTUK MAKANAN YANG DIPERIBADIKAN DAN MENGURANGKAN SISA

LESLIE KONG HAO ONG

Project ini merupakan salah satu keperluan untuk
Ijazah Sarjana Muda Sains Komputer dan Teknologi Maklumat
(Pengkomputeran Rangkaian)

Fakulti Sains Komputer dan Teknologi Maklumat
UNIVERSITI MALAYSIA SARAWAK
2025

DECLARATION

I, Leslie Kong Hao Ong with the matric number of 79858, hereby solemnly declare that the report titled '**Smart Recipe Generator: A Deep Learning-Powered Mobile App for Personalized and Waste-Reducing Food**' is prepared and completed authentically by me in partial fulfilment of the requirements for the degree of Bachelor of Computer Science with Honours (Network Computing) is a record of an original work carried out by me under the guidance of Associate Professor Dr. Johari bin Abdullah. I further declare that the work reported in this project has not been previously or concurrently submitted for any other degree at UNIVERSITI MALAYSIA SARAWAK (UNIMAS) or any other institutions. I declare that I have not copied or plagiarize from any other sources except for quotations and citations which have been duly acknowledged.

LESLIE KONG HAO ONG (79858)
Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my Final Year Project supervisor, Associate Professor Dr. Johari bin Abdullah, for his invaluable guidance, continuous support, and constructive feedback throughout this project. His expertise and encouragement have been instrumental in helping me overcome challenges and achieve my objectives.

I also wish to extend my heartfelt thanks to my examiner, Dr. Suhaila Binti Saee, for her insightful comments and suggestions, which greatly contributed to improving the quality of my work.

Additionally, I am grateful to my family, friends, and everyone who supported me during this journey, whether through their encouragement, advice, or understanding. This project would not have been possible without their unwavering support.

Thank you all for your contributions to this milestone in my academic journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADB	Android Debug Bridge
AI	Artificial intelligence
API	Application Programming Interface
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DFD	Data Flow Diagram
GB	Gigabyte
HTML	HyperText Markup Language
IDE	Integrated Development Environment
OS	Operating System
PHP	Hypertext Preprocessor
RAM	Random-access Memory
SDK	Software Development Kit
SQL	Structure Query Language
TPU	Tensor Processing Unit
UI	User Interface

ABSTRACT

The"Smart Recipe Generator" is a mobile application powered by deep learning solutions to change meal preparation to one that solves food waste and offers personalized nutrition needs. This project employs artificial intelligence with state-of-the-art image recognition technologies to recognize vegetables and fruits from user-uploaded images. It creates customized recipes via an Gemini API based on available ingredients as well as diets, restrictions, and preferences. Using TensorFlow and an Agile methodology, it ensures development will be iterative and user-centered. People should not throw away food they haven't used very much, rather, they can identify recipes for ingredients about to expire. The system also encompasses various dietary considerations, e.g. vegan, gluten-free, or low-carb diets. This Android solution has sustainability and health dimensions that intend to make life easier for consumers, moving them toward more informed, environmentally friendly food choices. Full application in all its functionalities as well as real-time identification with recipe modifications based on ingredients would add up to resource savings with individualized nutrition control. The application has received a lot of positive reviews during UAT. 80% of testers rated the application very easy to use (registration, login, and built-in function navigation), while 20% found it easy. Regarding performance, 12 of the 15 respondents gave both detection and recipe generation speeds very fast ratings. Functionally speaking, 100% of the users indicated the system did save recipes and pantry ingredients, and 93% said it did generate recipes correctly based on selected preferences. Another 87% strongly agreed on the app going a long way towards lessening food waste and better planning of meals. One of the most critical areas for future work is building on this model to allow for the detection of a wider range of food ingredients beyond the current 120 so as to cater to more diverse meals and cultural food varieties.

ABSTRAK

"Penjana Resipi Pintar" ialah aplikasi mudah alih yang dikuasakan oleh penyelesaian pembelajaran mendalam untuk menukar penyediaan makanan kepada yang menyelesaikan sisa makanan dan menawarkan keperluan pemakanan yang diperibadikan. Projek ini menggunakan kecerdasan buatan dengan teknologi pengecaman imej terkini untuk mengenali sayur-sayuran dan buah-buahan daripada imej yang dimuat naik pengguna. Ia mencipta resipi tersuai melalui API Gemini berdasarkan bahan-bahan yang tersedia serta diet, sekatan dan keutamaan. Menggunakan TensorFlow dan metodologi Agile, ia memastikan pembangunan akan berulang dan berpusatkan pengguna. Orang ramai tidak boleh membuang makanan yang tidak digunakan banyak, sebaliknya, mereka boleh mengenal pasti resipi untuk bahan-bahan yang akan tamat tempoh. Sistem ini juga merangkumi pelbagai pertimbangan pemakanan, cth. vegan, bebas gluten atau diet rendah karbohidrat. Penyelesaian Android ini mempunyai dimensi kemampanan dan kesihatan yang berhasrat untuk menjadikan kehidupan lebih mudah bagi pengguna, menggerakkan mereka ke arah pilihan makanan yang lebih termaklum dan mesra alam. Aplikasi penuh dalam semua fungsinya serta pengenalan masa nyata dengan pengubahsuaian resipi berdasarkan ramuan akan menambah penjimatan sumber dengan kawalan pemakanan individu. Permohonan ini telah menerima banyak ulasan positif semasa UAT. 80% penguji menilai aplikasi itu sangat mudah digunakan (pendaftaran, log masuk dan navigasi fungsi terbina dalam), manakala 20% mendapati ia mudah. Mengenai prestasi, 12 daripada 15 responden memberikan kedua-dua kelajuan pengesanan dan penjanaan resipi penilaian yang sangat pantas. Dari segi fungsional, 100% pengguna menunjukkan sistem itu menyimpan resipi dan bahan pantri, dan 93% mengatakan ia menjana resipi dengan betul berdasarkan pilihan yang dipilih. 87% lagi sangat bersetuju bahawa aplikasi itu akan membantu mengurangkan sisa makanan dan perancangan makanan yang lebih baik. Salah satu bidang yang paling kritikal untuk kerja masa depan ialah membina model ini untuk membolehkan pengesanan pelbagai bahan makanan yang lebih luas melebihi 120 semasa untuk memenuhi makanan yang lebih pelbagai dan varieti makanan budaya.

CHAPTER 1: INTRODUCTION

1.1 Project Title

Smart Recipe Generator: Mobile App for Personalized and Waste-Reducing Solutions

1.2 Introduction/Background

With the increasing use of mobile applications in daily life, there is a growing demand for intelligent systems that can help users in meal preparation and healthy eating. In 2023, the global mobile application market was estimated to be worth USD 252.89 billion which is around RM 1,000 billion. It is anticipated to expand at a compound annual growth rate (CAGR) of 14.3% from 2024 to 2030 (Grand View Research, 2023).

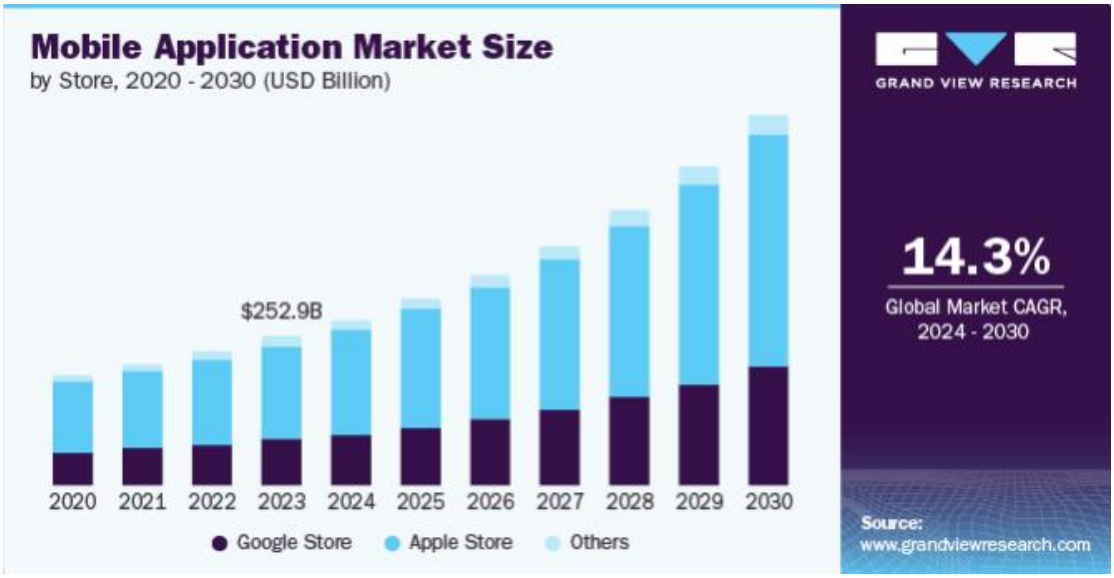


Figure 1.0 Estimated Growth of Mobile Application Market Size

Additionally, with rising awareness around sustainability, food waste management has become a pressing global problem. The Food and Agriculture Organization (FAO) estimates that approximately one-third of the food produced for human consumption is lost or wasted worldwide, amounting to about 1.3 billion tons yearly (FAO, 2021). The substantial quantity of food waste profoundly affects both the economy and the environment, highlighting the pressing necessity for remedies to

alleviate this issue. Intelligent mobile applications that assist users in managing their resources more effectively by providing personalised recipes depending on available products or those approaching expiration can significantly contribute to minimising food waste.

Individuals with diabetes must frequently monitor their carbohydrate consumption and maintain a balanced diet to regulate blood glucose levels. Research indicates that mobile applications for dietary management and nutritional tracking have resulted in enhanced health outcomes for individuals with diabetes, including improved blood glucose levels and overall well-being (Lee & Kim, 2024). These trends have led to the creation of AI-driven applications that identify food items, assess nutritional content, and suggest recipes customized for individuals with diabetes.

Moreover, dietary restrictions and preferences such as veganism, gluten-free, or low-carb diets have been increasingly common, prompting the need for flexible recipe options (Non-celiacs Drive Gluten-free Market Growth, 2014). This transition has generated a desire for tailored solutions that address specific dietary requirements, prompting the creation of sophisticated applications that provide customisable and adaptable recipe suggestions.

Ingredient recognition, recipe generation via the Gemini API, dietary restriction adaptations, and food waste minimisation through recipe recommendations that are based on available or residual ingredients are among the primary functionalities. The user interface will be intuitive and facilitate seamless interaction, and the project will be developed for mobile platforms (Android). Additionally, the application will enable users to input the names of dishes or their dietary preferences to receive personalised meal recommendations. Nevertheless, the scope is restricted to the identification of

frequently used fruits and vegetables, and it does not encompass other food categories or comprehensive multilingual support.

In summary, the rising incorporation of mobile applications into everyday life necessitates the development of intelligent systems that aid users in meal preparation and promoting healthy eating habits. In 2023, the worldwide mobile app industry was valued at USD 252.89 billion and is anticipated to have substantial growth during the forthcoming decade (Grand View Research, 2023). Food waste constitutes a significant global challenge, with around one-third of food produced for consumption discarded each year (FAO, 2021). Intelligent applications can improve users' management of available resources and provide customised meals, addressing sustainability and health issues, especially for persons with dietary limitations like diabetes. Integrating AI with food recognition technologies can aid in minimising waste and fostering healthy dietary practices.

1.3 Problem Statement

Current mobile apps for recipe ideas and meal planning often rely on manual input and provide limited personalization. Users face challenges when trying to decide what to cook with available items, especially those nearing expiration, leading to significant food waste. Furthermore, apps rarely adapt recipes to dietary preferences or restrictions, leaving users with restricted and inflexible options.

Research indicates that food waste is an escalating global concern. Approximately 931 million tons of food are wasted each year, with homes accounting for the majority of this waste (UNEP, 2021). Although meal planning applications seek to reduce waste, the majority are unable to identify components approaching expiration,

leading to lost opportunities for optimizing food utilization and minimizing waste. Moreover, studies indicate that customizing food and meal suggestions according to individual requirements might enhance user experience and encourage healthy dietary practices (Fitzgerald et al., 2020).

Despite advancements in AI and image recognition, few solutions integrate these technologies to offer real-time, personalized recipe ideas. The lack of intelligent systems capable of recognizing ingredients from images and generating tailored recipes based on dietary needs and available ingredients leaves a gap that can be filled by more innovative apps.

1.4 Project Objective

The primary objective of this project is to create a mobile application that implement deep learning and AI to offer users personalised meal recommendations that are calculated according to the availability of ingredients, dietary restrictions, and the necessity of reducing food waste.

Other objectives include:

1. To correctly identify fruits and veggies from pictures that users take by using deep learning algorithms.
2. To generate new and interesting recipes based on user input and known items by using the Gemini API.
3. Enable users to input dietary restrictions and adapt the recipe suggestions accordingly.

1.5 Brief Methodology

The application of methodology will be an Agile approach which provides a degree of flexibility and room for correction during development. The project is to be broken down into smaller units of work and will be completed in a series of increments. This also augurs well with the participation of the users since feedback is always welcome and even necessary for adjustments during the process The application of methodology will be an Agile approach which provides a degree of flexibility and room for correction during development. The project is to be broken down into smaller units of work and will be completed in a series of increments. This also augurs well with the participation of the users since feedback is always welcome and even necessary for adjustments during the process.

Figure 1.1 Agile Methodology

The first stage of the methodology will focus on requirement gathering and analysis. This includes collecting feedback from potential users to identify key features, such as ingredient recognition, recipe generation, and dietary customization. Additionally, research on existing solutions will be conducted to pinpoint areas where

the app can provide unique benefits, particularly in minimizing food waste and personalizing meal suggestions. Based on this information, a prioritized backlog will be created to guide the development process.

During the design and development stages, iterative cycles (sprints) will be followed, where one feature will be developed and refined at a time. The initial sprint will concentrate on designing the app's user interface to ensure user-friendliness. Next, deep learning models, such as CNN or MobileNet, will be implemented for ingredient recognition. Subsequent sprints will integrate the Gemini API for recipe generation and incorporate features like dietary restriction adaptations and food waste minimization. At the end of each sprint, features will undergo testing to ensure they meet the required functionality, with adjustments made based on feedback.

In the testing and deployment phases, continuous testing will ensure that new features integrate smoothly with existing components. As the project nears completion, efforts will focus on preparing the app for deployment, ensuring its stability and readiness for mobile platforms like Android. Post-deployment, user input will be gathered to facilitate ongoing improvements through additional sprints, enhancing the app's functionality over time.

1.6 Project Scope

The objective of this project is to create a mobile application that utilises deep learning and artificial intelligence to identify vegetables and fruits in images and then generate customised culinary recipes. Ingredient recognition, recipe generation via the Gemini API, dietary restriction adaptations, and food waste minimisation through recipe recommendations that are based on available or residual ingredients are among the

primary functionalities. The user interface will be intuitive and facilitate seamless interaction, and the project will be developed for mobile platforms (Android). Additionally, the application will enable users to input the names of dishes or their dietary preferences to receive personalised meal recommendations. Nevertheless, the scope is restricted to the identification of frequently used fruits and vegetables, and it does not encompass other food categories or comprehensive multilingual support.

1.7 Significance of Project

This project addresses multiple contemporary problems, such as food waste reduction, personalized meal planning, and dietary adaptation. By utilizing AI and deep learning technologies, the app brings an innovative solution to help users manage their ingredients easily and make healthier, more personalized food choices. The app's ability to suggest recipes based on real-time image recognition, dietary restrictions, and leftover ingredients encourages users to make informed decisions, possibly reducing food waste and adding to sustainable food consumption habits. It also adds convenience by automating recipe generation, which can save users time and effort in food planning.

1.8 Project Schedule

The project is expected to be completed within two semesters following the course Final Year Project I and Final Year Project II. Figure 1.2 will only show the current progress, and the complete Gantt will be shown in Appendix A.

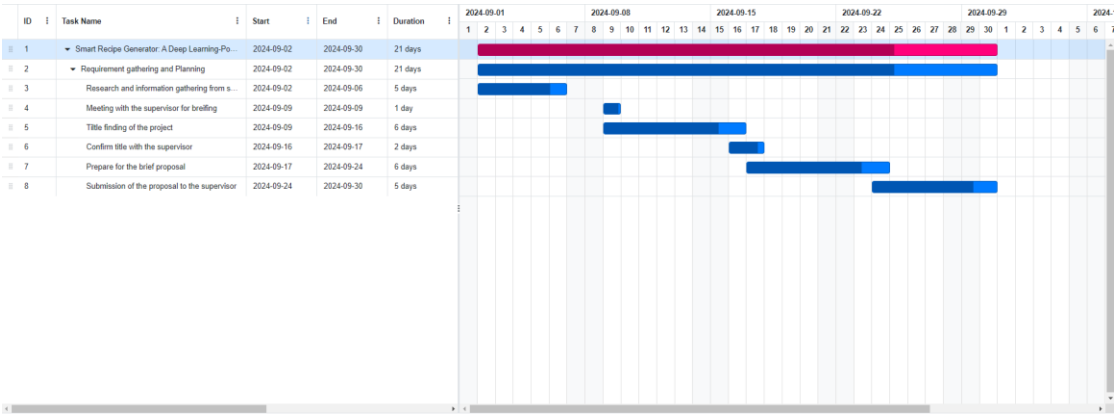


Figure 1.2: Current progress of Gantt Chart for Final Year Project

1.9 Expected Outcome

The planned output of this project is a fully functional mobile application that can properly recognize fruits and vegetables from photos and provide personalized recipes through an AI-powered system. The app will offer a user-friendly interface where users can either snap photographs of available ingredients input dish names to retrieve suitable recipes. It will adjust recipes to match certain dietary constraints (e.g., vegan, gluten-free), and it will assist reduce food waste by presenting ideas based on leftover or expiring components. The software will be tested and launched on Android smartphones, with the potential for further extension based on user feedback.

1.10 Required Skills

Android Studio, HTML, CSS, MySQL, JavaScript, PHP, OpenCV / YOLO

1.11 Project Outline

This paper consists of five chapters detailing the development and implementation of the proposed system that leverages AI and deep learning to provide real-time food recognition and personalized recipe suggestions. The system aims to address global issues such as food waste and dietary management.

Chapter 1: Introduction details an overview of the project including, its project title, background, problem statement, objectives, methodology, scope, significance, time schedule, expected results, and required skills. The background identifies the widespread integration of AI in mobile app the increasing concern of wastage of food thereby providing a background to the project. The problem statement indicates the existing meal-planning applications and their limitations, especially the lack of providing a unique intelligent recipe recommendation based on the chosen ingredients and the user's dietary requirements. The chapter highlights the prompt details of the project such as the built-in architecture of the system which includes the ability to recognize ingredients and create recipes. Methodology describes the stepwise system development, while scope provides the limitation of the system which is applicable to the basic and common vegetables and fruits. The significance segment highlights the benefits of the system with regards to food wastage and promoting dietary compliance to various health conditions. The expected outcome outlines the advantages of having AI used on the system that facilitates food differentiation in real time and the outline presents the main points covered in the paper.

Chapter 2: Literature Review takes a closer look at the available digital solutions and applications, which assist in making food and meal planning. It contains discussion of the key components integrating such applications, such as image recognition technologies and recipe generation techniques and addresses the

shortcomings of such applications in personalized meal suggestions. The chapter summarizes the important published articles, journals and industrial reports related to the AI and Machine Learning technologies implemented for the purposes of recognizing food ingredients and providing customized food recommendations. The review finishes with the description of the existing systems and their shortcomings that will be rectified in the proposed system that will integrate real time recognition of ingredients and the autonomous generation of recipes with the Gemini API.

Chapter 3: Requirement Analysis and Design outlines the process for gathering system requirements, involving stakeholders such as dietitians, software developers, and potential users. It discusses the system and user requirements for building a mobile application capable of recognizing ingredients and generating recipe suggestions. The chapter includes detailed design elements, such as the architectural design, flowcharts, data flow diagrams, and user interface prototypes, offering a visual representation of how the system will function. The design justifies the use of deep learning for image recognition and the Gemini API for recipe generation.

Chapter 4: Implementation and Testing details the process of implementing the proposed system, including the selection of development tools like TensorFlow for ingredient recognition and Gemini for recipe generation. The initial system prototype undergoes rigorous testing to assess its accuracy in recognizing ingredients and suggesting appropriate recipes. Feedback from user testing is used to refine the system, ensuring that it meets the project's objectives. The chapter also outlines the testing environment and procedures, evaluating the system's performance under different conditions, such as varying lighting for image recognition or handling dietary restrictions.

Chapter 5 contains the summary and possible directions for future research and innovation. It focuses on the objectives of the project and explains how AI was used for personal dietary management and nutrition planning. The chapter emphasizes the achievements as well as the drawbacks faced during the development, for instance, the project being limited to specific fruits and vegetables as well as lack of multilingual support. It also looks at the future work involving widening the recognition capability by including more food items and deeper dietary preferences. The conclusion outlines the direction of future research and development so that the system will be able to adapt to the new felt needs both in terms of diet and sustainability that are likely to be posed in the years to come.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

This chapter includes a comprehensive review of existing mobile applications with functionalities relevant to the proposed system. The focus lies on finding important features, procedures, and technology employed in similar solutions. Several comparable systems are studied to evaluate their strengths, weaknesses, and distinctive qualities, highlighting how they address difficulties like food management and personalized dietary tracking. Through this comparative analysis, the similarities and differences between existing applications and the proposed solution will be illustrated in an organised table.

Additionally, the chapter examines tools and technologies involved in the construction of the proposed system, including AI-driven ingredient recognition and recipe generation approaches. By merging the most effective aspects from existing solutions, the suggested system attempts to overcome current gaps such as restricted customisation and inadequate food waste management. A summary of the insights collected via this evaluation will provide a clear understanding of how the proposed application matches with and improves upon existing systems.

2.2 Review on Similar Existing Systems

This section provides a review and analysis of four mobile applications—ChefApp, DishGen, MyPantry, and Delicio that exhibit similarities to the proposed system. The innovative features and relevance to personalised meal planning, ingredient recognition, and food management of these applications have been selected.

Firstly, ChefApp - AI Recipe Creator stands out as an AI recipe creator by scanning the fridge or pantry and instantly make a recipe as well as other function instead of scanning (ACApplications, 2023). Besides, DishGen, which is the AI-Powered Kitchen Companion makes it easy to reduce waste and save time by generating custom-tailored recipes based on any idea you have or list of ingredients you need to use (Olson, 2023).

Similarly, MyPantry is a personalised cookery assistant that utilises the most advanced generative AI technologies to transform your ingredients into culinary masterpieces (Eclipse Apps LLC, 2023). Delicio, on the other hand, an app which is user-friendly and easy to navigate, making cooking a breeze (Salix Dijital Pazarlama Anonim Sirketi, 2023).

Although these applications encompass numerous critical functions, they fail to offer a comprehensive solution that seamlessly integrates ingredient recognition, waste reduction, and personalised meal suggestions within a single platform. The proposed system endeavours to resolve these constraints by integrating adaptive recipe recommendations, food administration solutions, and image-based ingredient recognition. This integration will offer a distinctive advantage over the reviewed applications by providing users with an all-in-one intelligent meal planning experience.

This chapter identifies the strengths to incorporate and the gaps to address by analysing the features and shortcomings of these extant systems, thereby ensuring that the proposed solution is in alignment with user needs and technological advancements.

2.2.1 ChefApp

Figure 2.1: Logo ChefApp

According to ACApplications (2023), ChefApp - AI Recipe Creator utilizes artificial intelligence to assist users in creating customized meal plans. Besides, it also includes function of creating recipe through the recognition of the ingredients in the users' pantry by implementing the machine learning to recognise the ingredients. The app also accommodates dietary preferences, providing vegan, gluten-free, low-calorie and other options for a personalized experience. A team of specialists from ACApplications is responsible for the innovative ChefApp application. Their mission is to help everyone to become a chef that can decide and design their own recipe without the need of professional chef around them. All they need is just a phone and key the ingredients as well as take a snap of photo of their existed ingredients (ACApplications, 2023).

ChefApp - AI Recipe Creator was first introduced and formally released in 2023. It is free to download in Apple App Store which is app store in Apple's IOS platform. For the functionality, there are some built in-app purchases through the subscription so that can remove Ads and unlimited recipes can be created. In other words, the free users are given 20 credits per day which means that free users can only create 20 recipes per day. The price of the subscription is allocated into 2 parts which are weekly subscription (RM19.90/week) and monthly subscription (RM59.90/month).

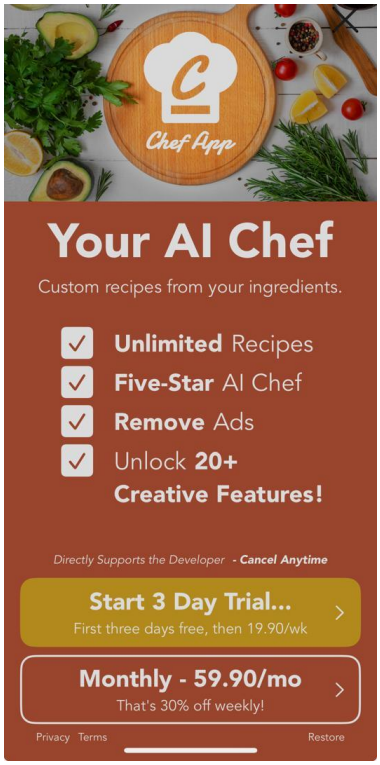


Figure 2.2: Subscription page of ChefApp

ChefApp is a mobile application that is fuelled by advanced AI and is intended to improve the cooking experience by providing personalised recipe recommendations, ingredient management, and meal planning tools. The Table 2.1 below show the features of the app.

Table 2.1: Features and Explanation of ChefApp

Feature	Explanation
AI-Generated Recipes	• Creates personalized recipes based on ingredients and preferences
Ingredient Recognition	• Identifies ingredients through photo scanning
Dietary Preferences	• Filters recipes to match dietary needs
Nutritional and time consumed Information	• Provides detailed calories and total time consumed analysis for recipes

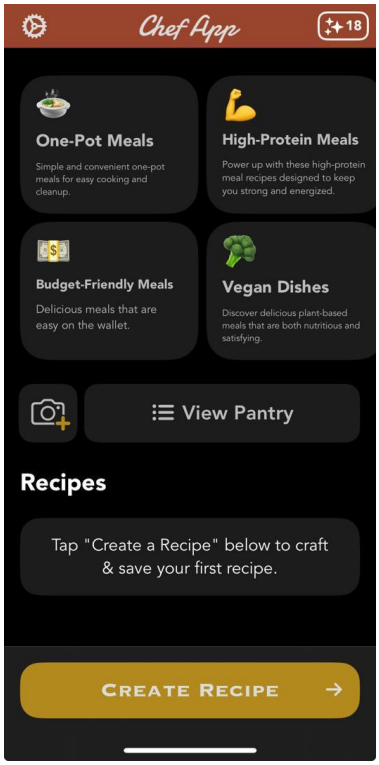


Figure 2.3: ChefApp Homepage

Initially, users can initiate ChefApp on their mobile devices by clicking the app icon after installing it from the App Store. It is not necessary for new users to register a new account. Upon launching, it will show the homepage of ChefApp. The left-top side is the setting button of the app while the right-top is the credit per user can be used to generate a recipe per day. For example, according to the Figure 2.2, the credit shown is 18, which is mean that user still can generate 18 recipes through this app by today. When turn to next day, the credit will be refreshed to 20 of credit. Next, the column below showed “One-Pot Meals”, “Budget-Friendly Meals” and other options are the feature of dietary preferences. User can choose the recipe according to their preferences like vegan dishes will only generate the recipe which is included vegetables. The camera icon leads the user to take a photo of the existed ingredients. The user can enable the crop option so that the only ingredients that exist in the frame can be recognised

which is shown in the Figure 2.3. Besides, the user is allowed to open the flashlight to increase the clarity of the photo taken or user can also use the photo in the gallery.

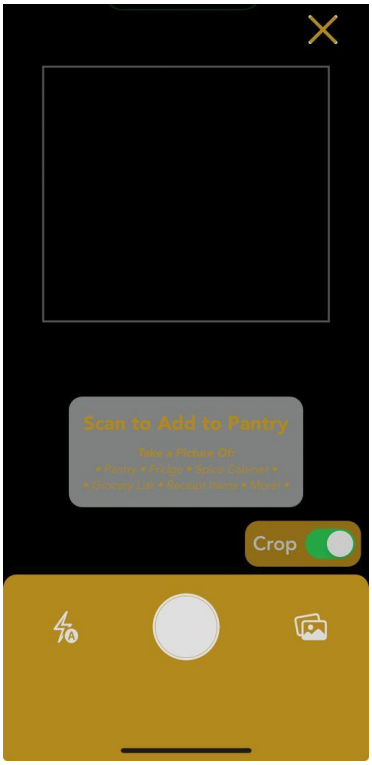


Figure 2.4: Feature Camera of ChefApp

Besides the camera icon button, is the button of viewing the pantry, it leads to show the ingredients from the photo taken by the user. All the ingredients recognised is sort into specific categories as shown in the Figure 2.4. Besides, if some of the ingredients are not recognised, user is allowed to add in manually. Of course, user can edit the ingredients by deleting the ingredients which are recognised wrongly.

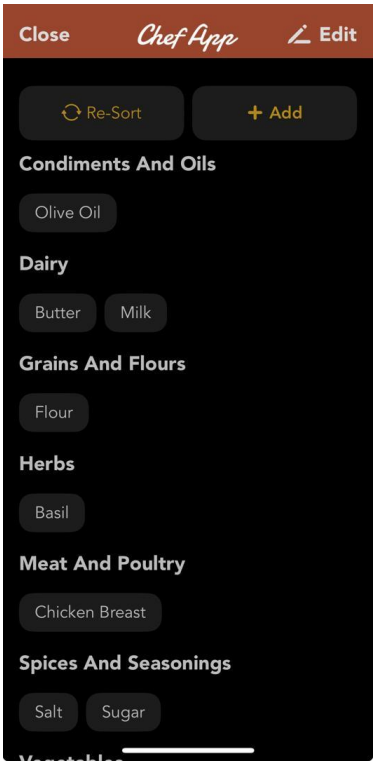


Figure 2.5 Feature View Pantry of ChefApp

After confirming the ingredients, user can generate the recipe by taping the button of “Create Recipe” below. Inside this feature, user can choose the type of meal wanted as well as some advanced options are opened for user to enable like “use all ingredients” to ensure AI does not omit any selected ingredients and “boost creativity” to let AI add ingredients to boost tastiness. Next, user can select the ingredients added just now in the feature of “view pantry”.

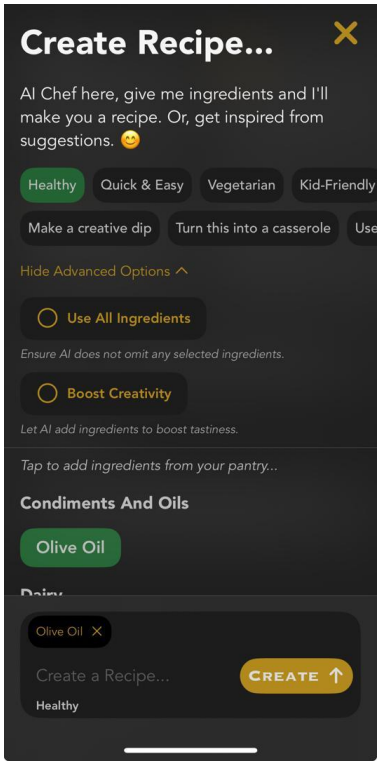


Figure 2.6 Feature Create Recipe of ChefApp

Lastly, tap the create button, two recipes will be created and shown in a list. More recipes can be unlocked through the subscription of the app as shown in the Figure 2.6. Inside the Figure 2.7, in a specific of the recipe, it will show the calories and total time consumed of this specific dish. Specific ingredients and the steps are also shown in the recipe at the same time, user can follow those steps to cook their own dish through the help of AI. Besides, user can save the ingredients by tapping the save button at the top-right corner of the page of recipe. Hence, credit will not be wasted by searching again the same ingredients to look for the same recipe.

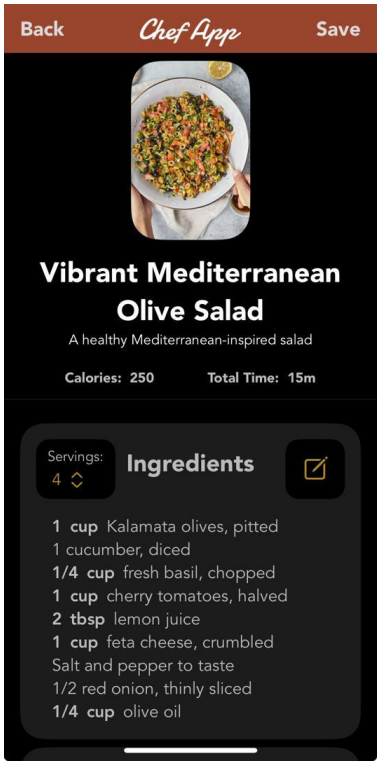


Figure 2.7 Recipe Generated

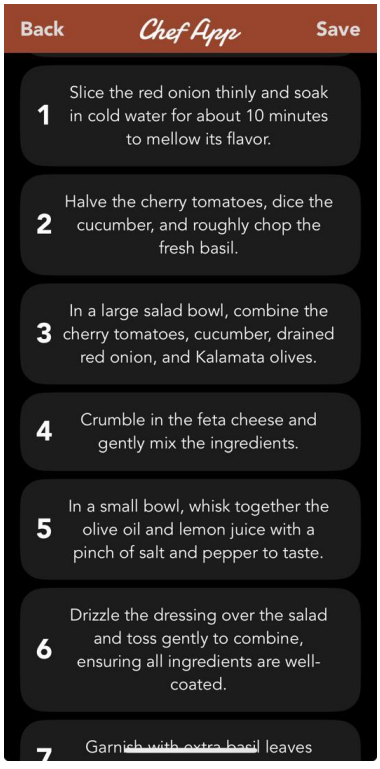


Figure 2.8 Step provided of the specific recipe

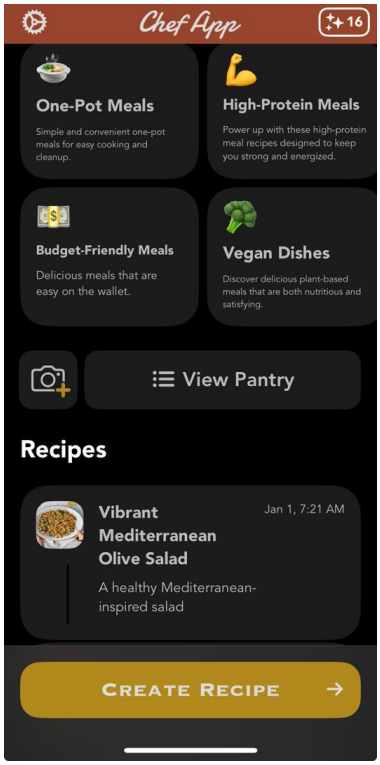


Figure 2.9 Saved Recipe

ChefApp provides users with a convenient method of generating recipes that are tailored to their dietary preferences and available ingredients, thereby facilitating meal planning and promoting mindful eating. The application offers an engaging user experience through its intuitive interface, camera-based ingredient recognition, and advanced AI features that provide personalised meal suggestions. However, despite its benefits, there are certain drawbacks that must be addressed, including the necessity of a subscription to access additional recipes and the constraints imposed by the daily credit system. The app's efficacy and user satisfaction will be improved by addressing these challenges.

Benefits of ChefApp are listed below in Table 2.2:

Table 2.2: Benefits of ChefApp

Benefits	Explanation
User-Friendly Interface	• No account registration is needed, providing quick access to users from the start. • The simple homepage design ensures easy navigation.
Dietary Preference Filtering	• Options like "One-Pot Meals" and "Budget-Friendly Meals" help users discover recipes that align with personal dietary needs (e.g., vegan).
Ingredient Recognition with Camera	• Users can scan their ingredients using the app's camera, reducing manual input and encouraging food waste minimization.
Advanced Recipe Generation Options	• Features like "use all ingredients" and "boost creativity" allow users to create highly personalized recipes with better taste outcomes.
Ingredient Management	• Users can edit, delete, or manually add ingredients to the

	pantry for better control over recipe inputs.
Health-Focused Features	• The inclusion of calories and cooking time in recipe details promotes mindful eating and time management.
Save Recipes for Future Use	• Users can save favorite recipes to avoid spending credits repeatedly on the same meals.

Drawbacks of ChefApp are listed below in Table 2.3:

Table 2.3: Drawbacks of ChefApp

Drawbacks	Explanation
Credit Limitation	• Non-subscribers are restricted by daily credits, limiting their ability to explore more than 20 recipes per day.
Subscription Requirement	• Access to a broader selection of recipes requires a paid subscription, which may discourage users seeking free solutions.
Limited Ingredient Recognition Accuracy	• Users might need to manually correct or input ingredients if the

	recognition system fails or produces inaccurate results.

ChefApp streamlines meal planning by offering customers a tailored recipe creation experience based on available ingredients and dietary preferences. With no registration necessary, customers can rapidly use its capabilities, such as AI-powered recipe creation and camera-based ingredient recognition, which allows consumers to scan ingredients or select from their pantry. The app ensures versatility through settings like “use all ingredients” or “boost creativity” for tastier outcomes. However, the daily credit limit, occasional mistakes in ingredient identification, and dependency on subscriptions for more recipes offer issues. Despite these shortcomings, the app's ease of use and customization make it a valuable tool for consumers seeking quick and innovative meal options.

2.2.2 DishGen

Figure 2.10: Logo DishGen

DishGen: AI Recipes is an intelligent kitchen assistant that eliminates the complexities of meal planning and preparation. This application utilises AI to swiftly develop customised meals, accommodating random components, dietary requirements, or the pursuit of culinary inspiration. DishGen prioritises the reduction of food waste while providing customised recipe recommendations, making it not merely a recipe generator but a resource for innovative and sustainable culinary practices (Olson, 2023).

DishGen: AI Recipes was first introduced and formally released in 2023. It is free to download in Google Play Store which is app store in Android platform and Apple App Store which is app store in Apple's IOS platform. Similarly, there are some built in-app purchases through the subscription so that can unlock other features like unlimited usage, more powerful idea generator and less time consumed to generate a recipe. However, the free users are only given 15 credits per month which means that free users can only create 15 recipes per month until next month the credit will be refreshed. The price of the subscription is allocated into 2 parts which are monthly subscription (RM39.90/month) and annual subscription (RM399.90/year). But, for those who subscribe either of these plans are given 7-day free trial which means that if the users are not satisfied with the app, they can unsubscribe the plan.

Figure 2.11: Subscription page of DishGen

The application provides a straightforward interface featuring recipe personalisation, clear instructions, and access to an extensive database of AI-generated meals. It also allows users keep track of their culinary adventure by bookmarking favourite recipes and browsing their cooking history. DishGen empowers anybody, from newbie cooks to culinary pros, to cook with confidence and creativity while decreasing food waste.

Table 2.4: Features and Explanation of DishGen (Olson, 2023)

Feature	Explanation
AI Recipe Creation	• Instantly generate unique recipes based on an idea or list of ingredients
Recipe Modification	• Tailor recipes to your taste with easy adjustments

Idea Generator	• Get creative suggestions for any dietary need or ingredients
Reduce Food Waste	• Make the most of your ingredients and minimize waste
Recipe Database	• Access thousands of AI-crafted recipes
User-Friendly Recipe Cards	• Easy to follow, perfect for cooking
Bookmarks & History	• Save your favourite recipes and track your culinary journey

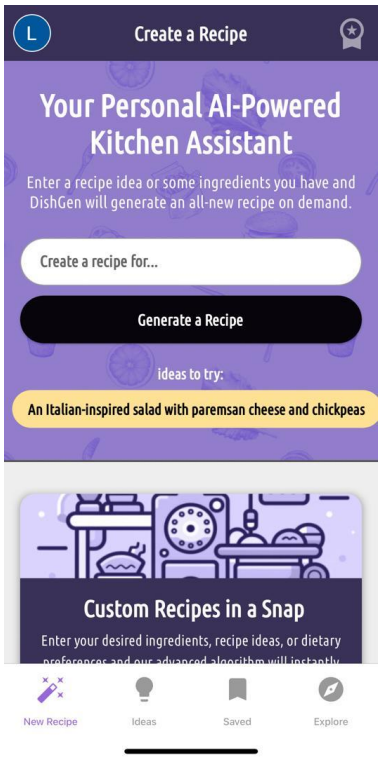


Figure 2.12: Homepage DishGen

After downloading the app either from Google Play Store or Apple App Store, users can easily access the app by just tapping the app icon which is shown in Figure

2.10. New users are told to register a new account before start using the app. From the left-top corner, an option of user profile can be chosen, then it will lead to users to their own user profile page which is shown in Figure 2.13. While the top-right corner which is a medal likely icon will lead the users to the subscription page.

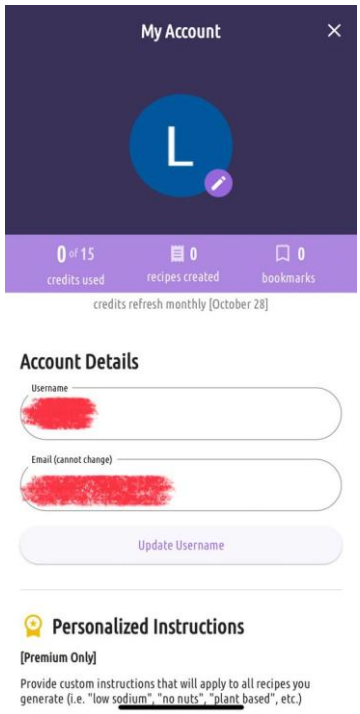


Figure 2.13: User Profile

Inside the user profile page, there are some records shown in the page like credit used, recipes created and bookmarks. Besides, the date of the credit to be refresh will be shown in the same page too. Next, turn to the account details, users are allowed to update their username and profile picture for sure at any time. Back to the home page, an obvious search bar is set for the users to enter a recipe idea, or some ingredients have so that the app can generate an all-new recipe on demand. Some ideas are given below the button “Generate a Recipe” so that users can direct choose them to generate a recipe without consider their own ideas. Then, there are some introductions of the app below it. Plus, there are some featured AI recipes with ingredients and steps provided for the

user without deducting the credit. Of course, the featured AI recipes can also be bookmarked so that the users will not lose the recipe.

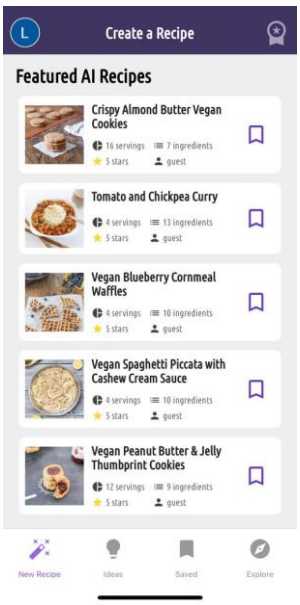


Figure 2.14: Featured AI Recipes

After tapping the button of “Generate a Recipe”, a recipe with the cuisine introduction and number of servings is generated and the usage of the credit will be shown too as shown in the Figure 2.15. Those recipes generated can be bookmarked so that users no need to waste their credit to search for the same recipe.

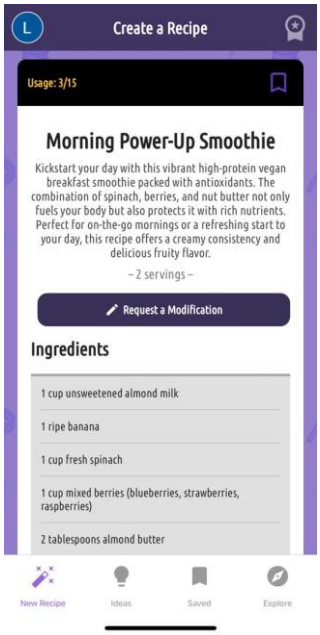


Figure 2.15: Recipe generated

Moreover, users are allowed to request for modification of the recipe. The ingredients and detail steps of the recipe will be shown in the same page too which is shown in the Figure 2.16. However, there is a note below every recipe which reminds the users DishGen will not take any responsibility for the accuracy or safety of the recipe due to recipe generated by the AI. So, every recipe should use their own best judgment when making AI-generated dishes.

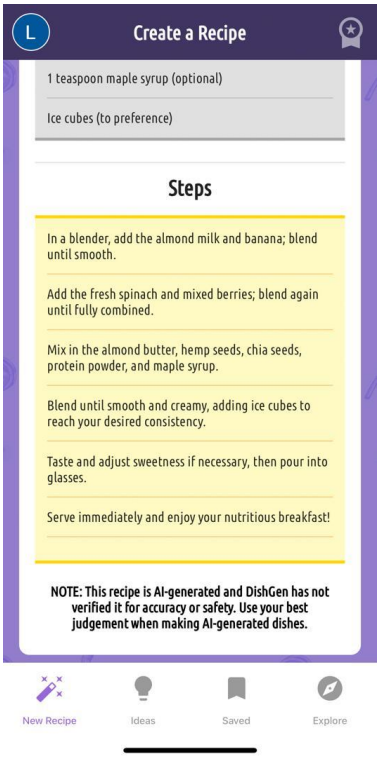


Figure 2.16: Ingredients and Steps of the recipe generated

Next, move to the next navigation which is “Ideas”, inside the “Ideas” is for the premium users. Users can use generate 7 creative recipe ideas in just one click. Food waste will be reduced by getting a bunch of tasty recipe ideas for users’ leftovers to save money.

Figure 2.17: Navigation "Ideas"

Other than that, users can refer to their recipe before through bookmarks or history in the navigation of "Saved".

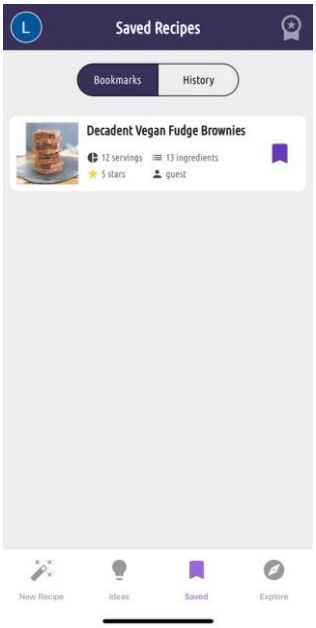


Figure 2.18: Bookmarks

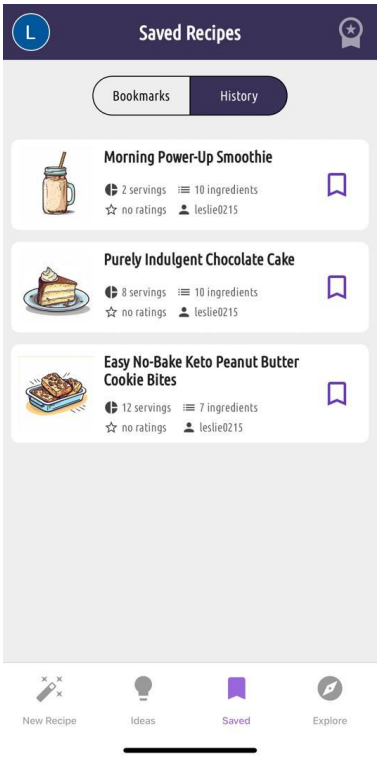


Figure 2.19: History

Lastly, for the navigation “Explore” users can search for the trending or most saved recipes. Similarly, it provides those recipes’ ingredients and steps needed. Users are allowed to search for those recipes by using tags for example like spicy, easy, healthy etc.

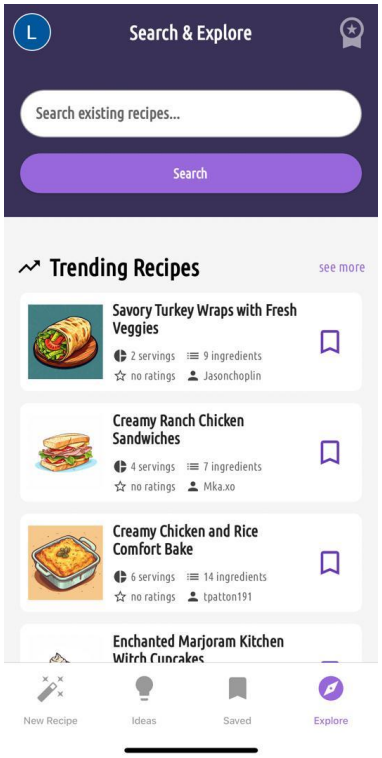


Figure 2.20: Trending Recipes

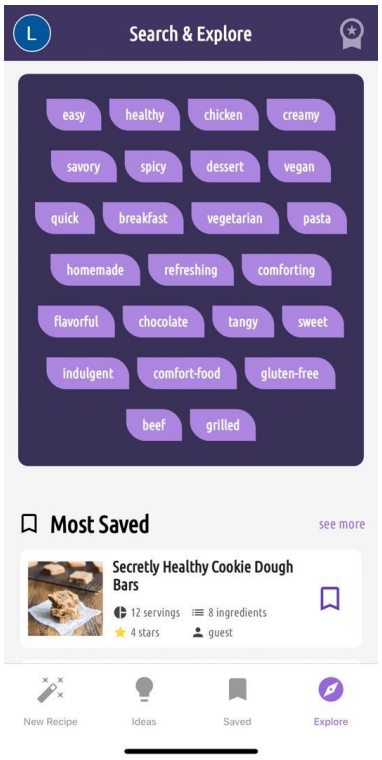


Figure 2.21: Tags and Most Saved Recipes

While DishGen has various benefits, such as personalized AI-generated recipes, waste reduction, and an easy UI, it also has its drawbacks. The app's limited credits for free users and the need on in-app subscriptions might restrict access to important functions, making it less desirable to some. Additionally, AI-generated recipes may sometimes lack precision or consistency, forcing users to adjust them manually. Despite these restrictions, DishGen remains an important tool for people seeking inventive and ecological culinary options.

Benefits of DishGen are listed below in Table 2.2:

Table 2.5: Benefits of DishGen

Benefits	Explanation
AI-Generated Recipes	• The app quickly creates customized recipes based on available ingredients or meal ideas, helping users avoid decision fatigue and get creative in the kitchen.
Reduces Food Waste	• By generating recipes from leftover ingredients, DishGen encourages sustainable cooking and minimizes wastage, saving money and resources.
Recipe Personalization	• Users can modify generated recipes according to their preferences, dietary needs, or

	ingredient availability, ensuring tailored results.
User-Friendly Interface	• The simple and intuitive layout makes navigation easy for both beginners and experienced cooks, enhancing the user experience.
Recipe Bookmarking and History	• Users can save favorite recipes and revisit previous creations, allowing them to track and improve their culinary skills over time.
Free Trial and Subscription Flexibility	• The 7-day free trial allows users to explore premium features before committing, while both monthly and annual subscription options offer flexibility.
Access to Featured Recipes Without Credit Usage	• Users can try featured AI recipes without spending credits, providing value even for free users.

Drawbacks of DishGen are listed below in Table 2.3:

Table 2.6: Drawbacks of DishGen

Drawbacks	Explanation
Limited Credits for Free Users	• The 15 monthly credits limit can be restrictive for users who frequently need recipes, potentially reducing app utility for non-subscribers
In-App Purchases Required for Full Access	• Some key features, like unlimited recipe generation and the "Ideas" section, are locked behind a paywall, which could deter some users.
Subscription Cost	• The subscription plans (RM39.90/month or RM399.90/year) may be considered expensive by some users, limiting accessibility to the premium features.
Dependency on AI Accuracy	• As recipes are AI-generated, there may be inconsistencies in taste or preparation, with a disclaimer that DishGen is not

	responsible for recipe accuracy or safety.

DishGen's user-friendly interface provides easy recipe editing, bookmarking, and access to featured AI recipes without credit usage. However, drawbacks include the credit restriction for free users, need on in-app purchases for full access, and occasional AI mistakes. The software also requires a reliable internet connection, and its monthly fees may deter some users. Despite these obstacles, DishGen presents a practical and creative solution for anybody wishing to enhance their culinary experience while avoiding waste and managing supplies properly.

2.2.3 MyPantry

Figure 2.22 Logo MyPantry

MyPantry redefines home cooking by converting everyday ingredients into personalized culinary experiences. Leveraging advanced generative AI, the app tailors recipes to suit users' dietary preferences, cuisines, and accessible ingredients. Whether you're planning a specific meal or playing with what's in your pantry, MyPantry offers seamless solutions to encourage creativity in the kitchen.

With features like ingredient inventory management, recipe exploration, and a guided Cook Mode, MyPantry ensures that cooking becomes more efficient and pleasurable. Users can also bookmark and organize their favorite recipes, making meal planning straightforward and hassle-free. MyPantry is the perfect partner for anybody wishing to simplify cooking while making the most of their ingredients.

According to Eclipse Apps LLC (2023), MyPantry: Generate Recipes was introduced and officially released in 2023. It is free to download from the Apple App Store, which is an app store for Apple's iOS platform. For functionality, there are some built-in in-app purchases available via subscription, allowing you to earn infinite recipe generating tokens and access to special features. In other words, free users receive one credit per day, which implies they can only make one dish each day. The PRO subscription is RM29.90 per month.

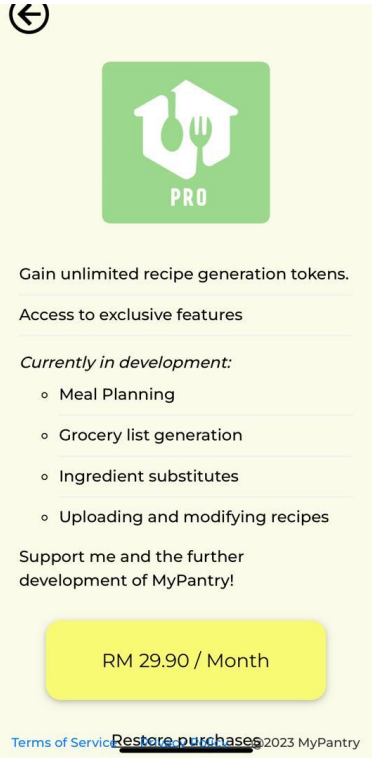


Figure 2.23: Subscription page of MyPantry

MyPantry uses innovative AI technology to simplify cooking and ingredient management. The software delivers tailored recipe suggestions, flawless ingredient detection, and an orderly cooking procedure, making it a multipurpose tool for home chefs. The Table 2.7 below show the features of the app.

Table 2.7: Features and Explanation of MyPantry

Feature	Explanation
Tailored Recipe Generation	• Create customized recipes based on dietary preferences, cuisines, or a list of ingredients. Generate multiple dish options for experimentation.

Ingredient Recognition & Management	• Identify ingredients by taking photos of items, receipts, or grocery lists. Manually select from predefined lists or add custom items.
Explore Recipes	• Browse and discover community-shared recipes filtered by the ingredients available in your pantry.
Cook Mode	• Navigate through recipes with a step-by-step interface, providing a clear and organized cooking experience.
Save and Organize Recipes	• Bookmark your favorite recipes and categorize them into folders for easy meal planning.

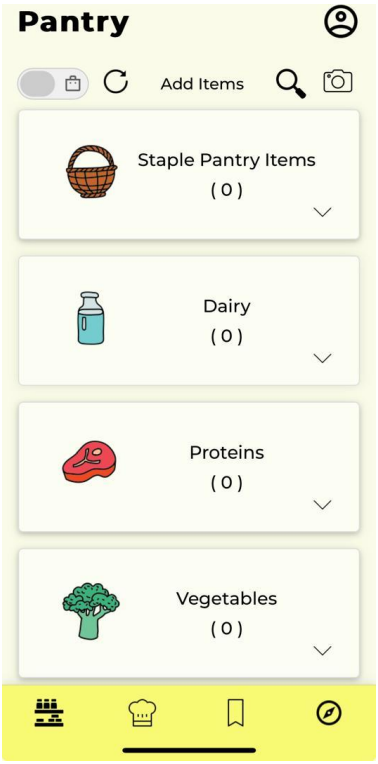


Figure 2.24: Homepage of MyPantry

Once the users tapping on the icon of MyPantry which is shown in the Figure 2.22, it will lead the users to the homepage as the Figure 2.24. Start from the top-right corner of the app, the profile icon will lead the users to their profile page. But, of course, users need to create or register an account first before entering the profile page. Inside the user profile page, it will show the credit left at the top-right corner, the status of your account which is PRO version or not, the email address registered by the users and logout button as well as the delete account button as shown in Figure 2.25.

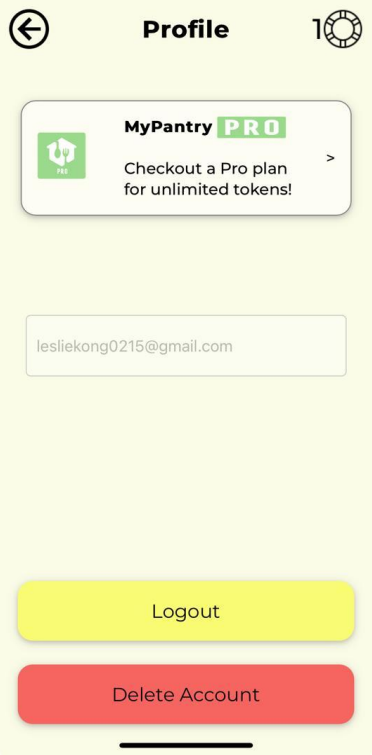


Figure 2.25: User Profile

There is a switch at the top-left corner which allows the users to add items and view items added. Besides the switch is the refresh icon which will let the users to clear all the items added in the pantry. Next, the search icon beside is to allow the users to add anything they want by searching it. Lastly, the users can recognize the ingredients in their pantry by taking the photos through the tapping of the camera icon. Not only than that, but it can also recognize ingredients through the photos of grocery list or receipt. But, due to the high accuracy of recognizing ingredients, it will take a longer time to wait for the AI to recognize and may list out all the possibilities. Users can decide to keep or delete the recognized ingredients. For example, a photo of tomatoes is taken, and it recognizes the tomatoes and categorize it. If fail to recognize, users can just directly add in and import them to your pantry. After importing the ingredients, they will be shown in the specific category at the homepage as shown in Figure 2.27.

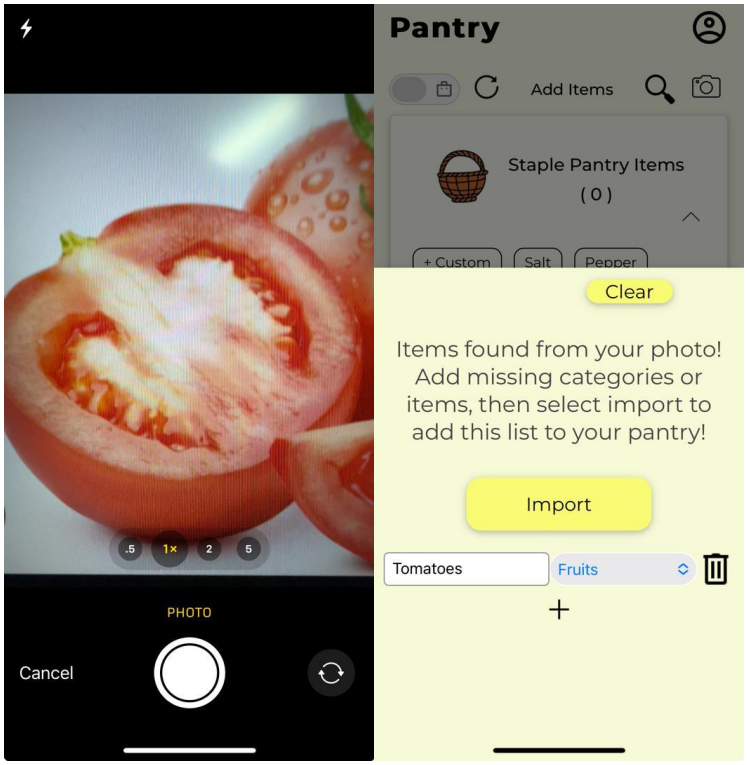


Figure 2.26: Camera Recognition

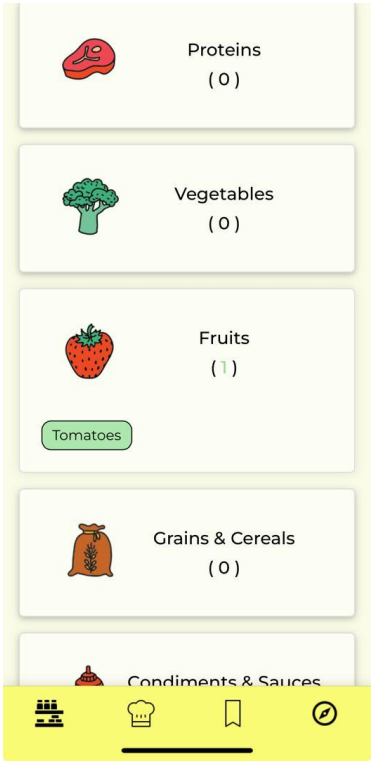


Figure 2.27: Import ingredients

Next, move to the next navigation which is creating recipe. So, in default all the ingredients existed in pantry will be selected to generate recipe. Users are allowed to remove the unwanted ingredients to be selected.

Figure 2.28: Selected Ingredients

After selecting all the ingredients needed, users can generate or create recipe by taping the “Start Creating” button which is below the page. Then, users can select their key ingredients which means the ingredients that definitely want to include in the recipe. The next step is let the users to select what are they looking for create like main course, side dish, dessert etc. Besides, users can select the specific cuisine for their incoming dish like Italian, Mexican, Indian, Japanese etc. Moreover, the dietary preferences can be selected by the users like vegan, gluten-free etc. Not only than that, even the vibe of the recipe generated can also be selected (Quick and Easy, Complex, Minimal Prep, Advanced Skills Needed etc). Lastly, users just review and submit all the options made and decide the number of people.

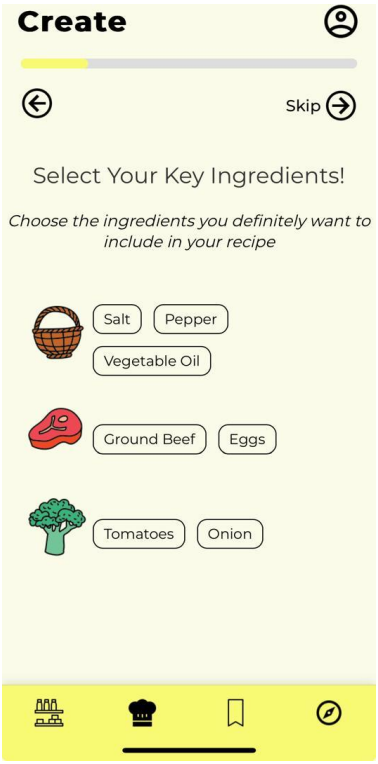


Figure 2.29: Select Key Ingredients

Figure 2.30: Select Type Dish

Figure 2.31: Select Specific Cuisine

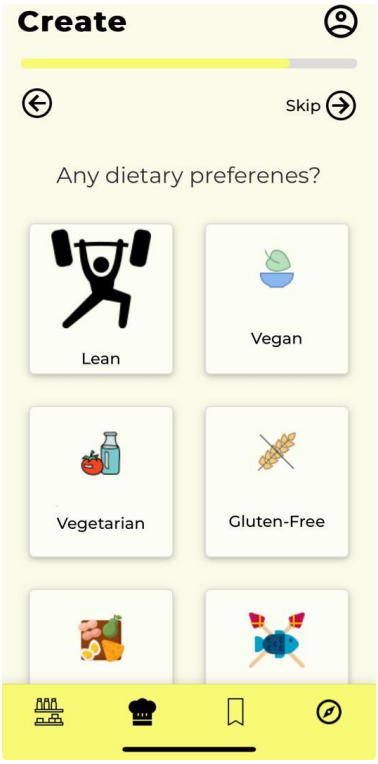


Figure 2.32: Select Preferences

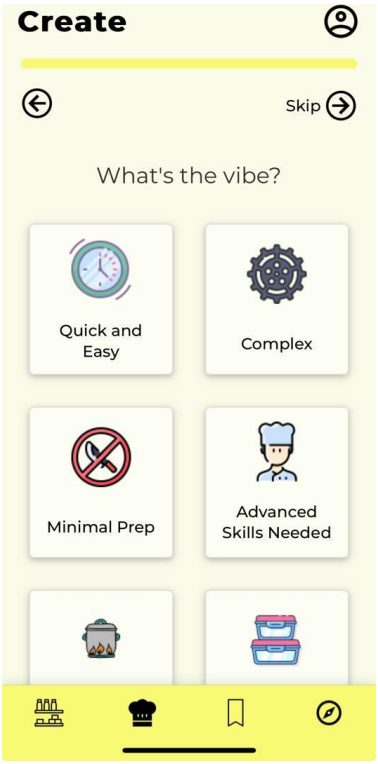


Figure 2.33: Select Vibe

Figure 2.34: Review and Submit

After users submit all the options, an advertisement will be shown in the duration of generating recipe as shown in Figure 2.35. Lastly, a recipe with intro and picture of the dish will be generated. Besides, other info like time consumed and number of serves as well as calories (with protein, carbohydrates and fats). Of course, users are given ingredients and instructions to be followed to cook the existing dish. Users also can save the recipe generated by tapping the bookmark icon at the top-right corner of the generated recipe page. Figure 2.37 shows that all the saved recipes are in the bookmark navigation.

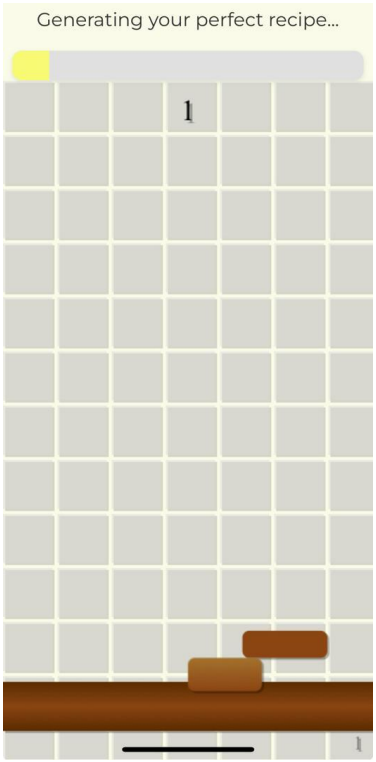


Figure 2.35: Advertisement

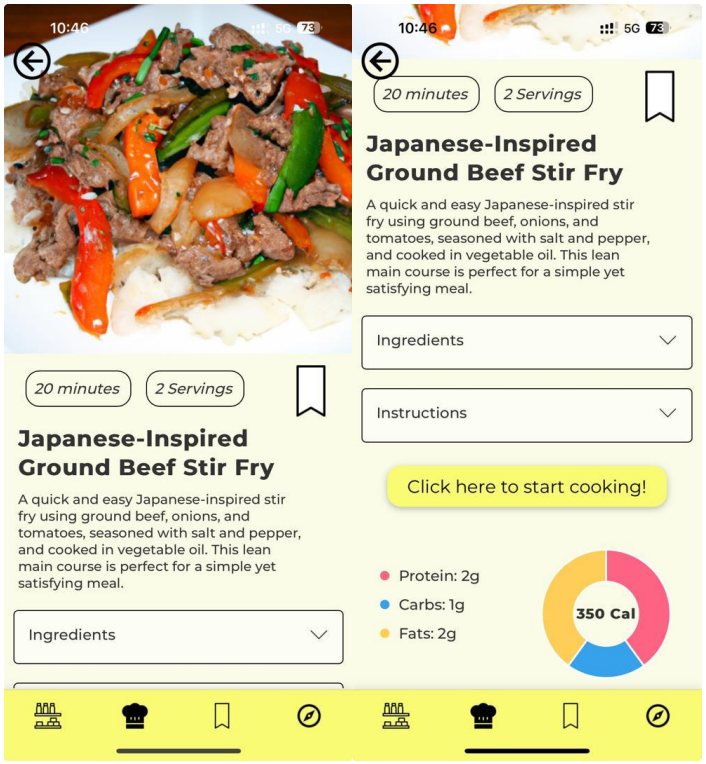


Figure 2.36: Recipe Generated

Figure 2.37: Saved Recipe

Moreover, if users are out of credit, users can go to the explore page (Figure 2.38) to explore more random recipes. Same as recipe generated, all the info will be given and all of it can be saved too so that users can review again without the waste of credit.

Figure 2.38: Explore Recipe

Although MyPantry offers various benefits, such as personalized recipe production, efficient ingredient management, and a guided cooking experience, it also has certain drawbacks. The app's free edition limits users to one recipe each day, driving many toward a paid membership. Additionally, the great precision of ingredient recognition might result in sluggish processing times, which may disappoint users wanting speedy answers. Advertisement interruptions during recipe generation significantly disturb the user experience. While MyPantry improved cooking for many, these challenges may decrease overall user happiness.

Benefits of MyPantry are listed below in Table 2.8:

Table 2.8: Benefits of MyPantry

Benefits	Explanation
Personalized Recipe Suggestions	• The app uses AI to generate recipes tailored to users' dietary preferences, selected ingredients, and desired cuisine types, ensuring that meals align with personal tastes.
Reduces Food Waste	• By generating recipes based on available pantry items, the app encourages efficient ingredient use, helping users avoid expired or unused food.
Ingredient Recognition via Photos	• Users can quickly identify and add items to their pantry by taking photos of ingredients, grocery lists, or receipts, saving time compared to manual entry.
Community-Shared Recipes	• Access to recipes contributed by other users provides inspiration and allows users to explore new cooking ideas.

Guided Cooking Process (Cook Mode)	• The step-by-step interface simplifies cooking, especially for beginners, ensuring clear instructions and a more enjoyable experience.
Convenient Recipe Management	• Users can bookmark favorite recipes and organize them into folders, making future meal planning more efficient and accessible.

Drawbacks of MyPantry are listed below in Table 2.9:

Table 2.9: Drawbacks of MyPantry

Drawbacks	Explanation
Limited Free Access	• Free users are restricted to generating only one recipe per day, which may be insufficient for users who cook multiple meals.
Subscription Costs	• Access to unlimited recipe generation and special features requires a PRO subscription, which might be expensive for

	some users at RM29.90 per month.
Long Ingredient Recognition Time	• Due to the high accuracy of AI-based ingredient recognition, processing can be slow, causing delays for users who need quick results.
Advertisement Interruptions	• Ads are shown during recipe generation, which can disrupt user experience and feel intrusive.

MyPantry is a modern kitchen aid that uses advanced generative AI to make meal preparation easier. It customizes meals based on user preferences, available supplies, and dietary requirements, making meal planning more personal and efficient. The app includes features such as photo-based ingredient detection, pantry management, and a guided Cook Mode to help you stay organized while cooking. Users can browse a wide range of community-shared recipes, save favorites, and categorize them for easy access. However, the free version of the program only allows users to create one recipe each day, with extra capabilities only available with a premium membership. While MyPantry makes cooking more convenient, delayed ingredient detection and adverts during recipe generating might be inconvenient. Despite these disadvantages, MyPantry stands out as a useful tool for home chefs looking to reduce food waste and expand their culinary creativity.

2.3 Features Comparison of the Reviewed Systems

Table 2.10: Comparison of Features between Existing Systems and Proposed System:

Features	Existing Systems			Proposed System
	ChefApp	*DishGen*	*MyPantry*	
Subscription Fees (In-app Purchase)	✓	✓	✓	✗
Deep Learning (AI)	✓	✓	✓	✓
Ingredient Recognition (Photo-based)	✓	✗	✓	✓
Dietary Preferences	✓	✓	✓	✓
Food Waste Reduction	✗	✓	✓	✓
Save and Review Past Recipes	✓	✓	✓	✓
Source of Input	Camera, Manual	Manual	Camera, Manual	Camera, Manual
Assessment Modes	Ingredient Recognition,	Ingredient Recognition,	Ingredient Recognition,	Ingredient Recognition,

	Recipe Generation	Recipe Generation	Recipe Generation	Recipe Generation
Limitations	Free with daily credit limits; occasional ingredient inaccuracies	Free with daily credit limits; occasional ingredient inaccuracies	Free with daily credit limits; occasional ingredient inaccuracies	No limitation

According to the table 2.10, a detailed comparison of features among three existing AI-powered recipe apps—ChefApp, DishGen, and MyPantry as well as the proposed system have been listed clearly. The capabilities and limitations of each application are comprehensively assessed by evaluating each system across a variety of factors, such as subscription fees, the use of AI (specifically deep learning), input sources, and distinct functionalities.

Beginning with subscription fees, all three current systems—ChefApp, DishGen, and MyPantry—demand in-app purchases or offer subscription plans for full feature access, whereas the proposed system is meant to be accessible without any such restriction, so potentially more user-friendly and cost-effective for regular use. A characteristic common across all applications, each system uses deep learning (AI) to improve recipe generating and user experience.

The ingredient recognition feature, which is a critical component of food-related AI applications, is integrated into both ChefApp and MyPantry. This feature enables

users to identify ingredients through photo-based inputs. DishGen, on the other hand, is devoid of this functionality and is entirely dependent on manual inputs. In addition, all current systems offer dietary preference filtration, which allows users to personalise recipes according to their dietary restrictions or preferences. DishGen and MyPantry are the only applications that offer the ability to reduce food spoilage by recommending recipes that are based on the available ingredients. ChefApp does not have this feature.

A crucial user-friendly feature which is saving and reviewing past recipes is given by all three existing applications, providing easy access to preferred or frequently used recipes. In terms of input sources, ChefApp and MyPantry offer both camera and manual inputs, giving users with multiple options for entering items. DishGen, on the other hand, limits input to manual entries alone.

The assessment modes in all three systems include ingredient detection and recipe production, which aligns with the apps' primary goal of simplifying cooking and meal planning. However, constraints are there in the form of daily or monthly credit limits for free users in each existing system, as well as occasional ingredient detection errors. In contrast, the proposed system offers to provide unlimited access with no constraints, distinguishing it from the reviewed applications.

Overall, this table provides a clear, side-by-side analysis of the strengths and drawbacks of each system. The proposed system, with no subscription requirement and unlimited access, intends to address some of the limitations observed in the existing systems, making it a potentially more accessible and versatile choice for users seeking AI-powered recipe solutions.

2.4 Review on Tools and Technology

This section provides an overview of the tools and technology required for developing the proposed system. It encompasses both software and hardware components, outlining their background, functions, and importance to the system's aims. Serving as a "justification" section, this chapter outlines the logic behind selecting particular tools and technologies over other alternatives, ensuring that each option matches with the system's unique requirements and goals.

2.4.1 Software

The software used in the proposed system will be review in this section clearly.

2.4.1.1 Deep Learning

Numerous facets of modern life are driven by machine-learning technology, encompassing web searches, content curation on social networks, and suggestions on e-commerce platforms. It is becoming used in consumer items such as smartphones and cameras. Machine-learning algorithms are utilised to detect objects in photos, transcribe speech into text, align news items, posts, or products with users' interests, and choose pertinent search results. These applications are progressively utilising a set of methodologies referred to as deep learning (LeCun et al., 2015).

Neural networks, inspired by biological neurons, are built with interconnected layers of artificial neurons or nodes, each processing incoming data through weighted connections and activation functions. Gibson and Patterson (2017) underline that training neural networks requires modifying the weights to reduce prediction errors, a

process crucial for the network's ability to "learn" over time. Backpropagation, a technique for updating weights across layers, plays a critical role in this training process, allowing neural networks to systematically reduce mistakes and improve performance (Patterson & Gibson, 2017).

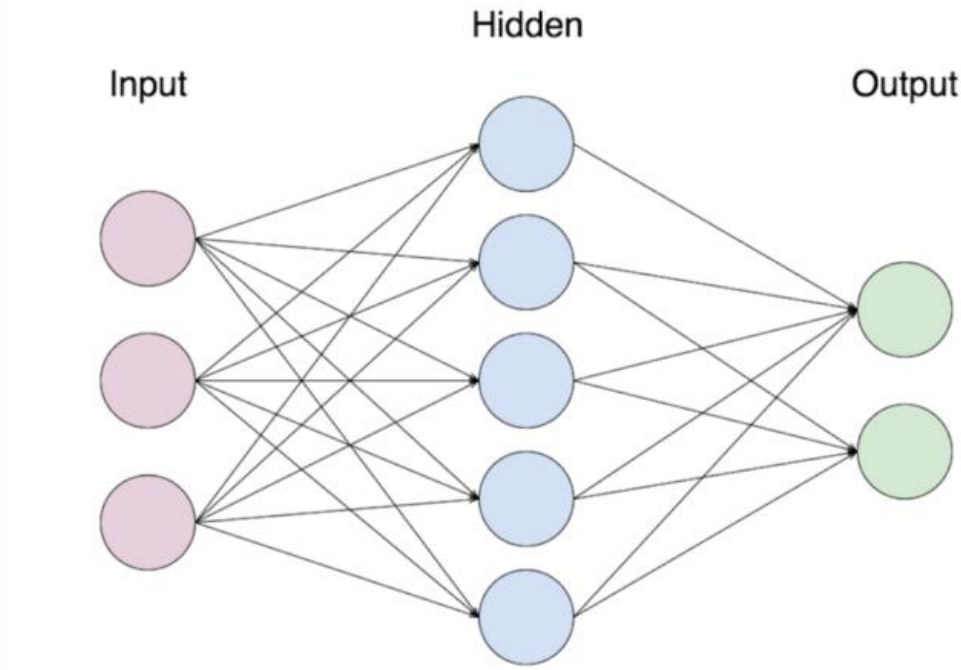


Figure 2.39: "Hidden" equals Deep Learning Node

Source: https://orbograph.com/understanding-ai-what-is-a-deep-learning-node/ (Orbograph, 2020)

A node is a location where processing occurs, loosely analogous to a neurone in the human brain, which activates upon receiving sufficient stimulation. A node integrates input data with a series of coefficients, or weights, that either enhance or attenuate that input, so attributing significance to inputs in relation to the objective the algorithm aims to achieve, such as identifying which input is most effective for accurate data classification. The input-weighted products are aggregated and subsequently processed through a node's activation function to determine the degree to which the signal should propagate through the network, ultimately influencing the final outcome,

such as a classification action. If the signals are transmitted, the neurone has been "activated" (Orbograph, 2020).

Various network designs are applied in deep learning to respond to diverse data kinds and applications. Feedforward networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and autoencoders each fulfil distinct tasks. CNNs, for instance, are very successful in spotting patterns in image data, whereas RNNs excel in processing sequential data such as text and time series (Patterson & Gibson, 2017). The adaptability of these designs is further highlighted by generative models like autoencoders and generative adversarial networks (GANs), which are effective for creating fresh data in unsupervised tasks.

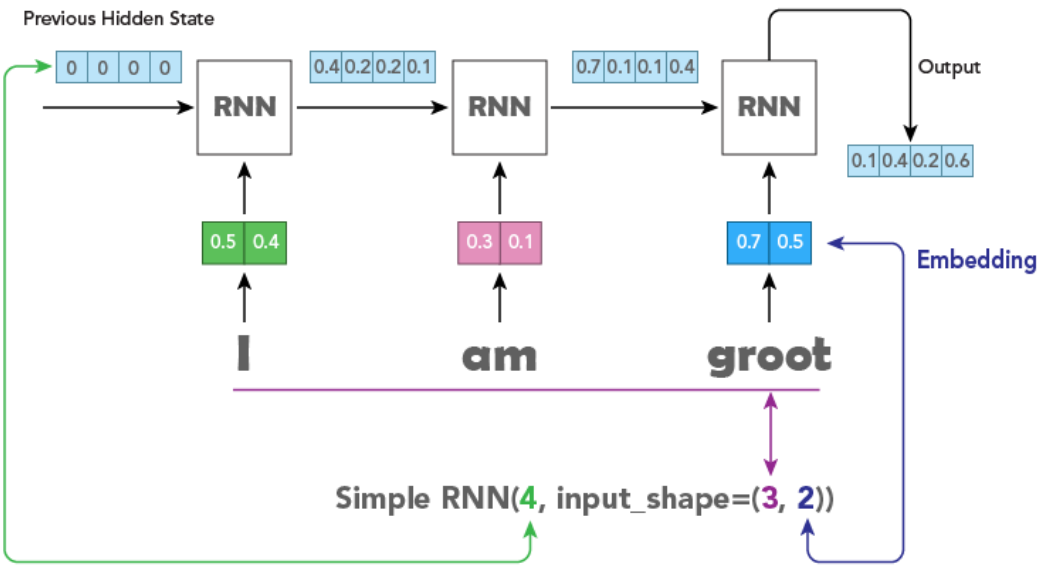


Figure 2.40: RNN Architecture

Source: https://www.softwebsolutions.com/resources/difference-between-cnn-rnn-ann.html (Solanki, 2024)

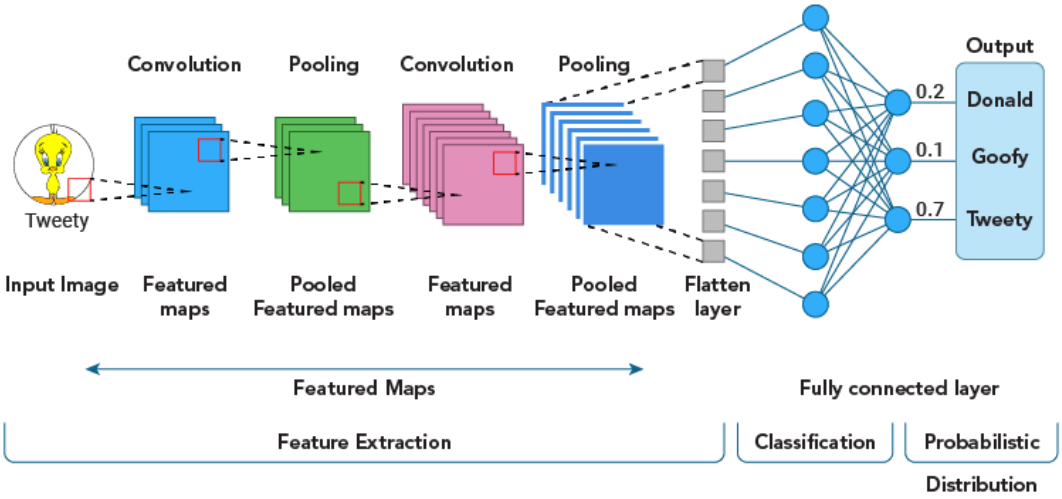


Figure 2.41: CNN Architecture

Source: https://www.softwebsolutions.com/resources/difference-between-cnn-rnn-ann.html (Solanki, 2024)

Table 2.11: Difference Between RNN and CNN

Source: https://www.softwebsolutions.com/resources/difference-between-cnn-rnn-ann.html (Solanki, 2024)

Features	CNN	RNN
Data Type	Image	Sequential
Parameter Sharing	Yes	Yes
Fixed Length Input	No	Yes
Recurrent Connections	Yes	No
Gradient Issues	Susceptible to vanishing and exploding gradients	Susceptible to vanishing and exploding gradients
Spatial Relationships	Maintains spatial relationships	No spatial relationships

Performance	Generally more effective in handling images	Less suited for feature-rich data compared to CNN
Applications	Facial recognition, text digitization, NLP	Text-to-speech, time series analysis
Advantages	High accuracy for image recognition, efficient weight sharing	Good memory retention for sequential data, suitable for time series
Disadvantages	Requires large datasets, lacks positional encoding	Prone to gradient vanishing and exploding issues

Training and improving deep networks require complex approaches. As Patterson and Gibson (2017) emphasise, optimization approaches like gradient descent are critical for modifying weights to decrease errors effectively. Additionally, the adjustment of hyperparameters—including learning rate, batch size, and regularization methods—has a substantial impact on the model's overall performance and its ability to generalize to new data.

Implementing deep learning models in practice involves employing specialized libraries and distributed computing systems. Deeplearning4j (DL4J), a deep learning library for Java, is recognised by Patterson and Gibson (2017) as a useful tool for constructing deep networks in enterprise applications. Additionally, Apache Spark's distributed computing capabilities are vital for handling large-scale datasets, enabling fast training across numerous computers. Vectorization techniques are also critical for preparing varied data types, ensuring that they are efficiently arranged for input into deep networks (Patterson & Gibson, 2017). Through these implementations, deep

learning models are increasingly scalable and adaptable to many real-world applications.

2.4.1.2 Dataset for Deep Learning

In machine learning, a dataset is a collection of data instances used to train, validate, and test models. It often comprises labeled examples for supervised learning tasks, where each instance is paired with a label or goal value. According to Yu et al. (2015), big datasets are necessary for deep learning because they enable the adjustment of various model parameters, contributing to higher accuracy in complicated tasks such as image and object recognition (Yu et al., 2015).

For example, medical datasets, such as those used in colorectal cancer detection, are highly valuable yet hard due to restricted availability and the necessity for exact categorization. Durdov et al. (2024) examine the value of medical datasets in machine learning, noting that their availability is constrained, often causing researchers to rely on small, internal datasets. They demonstrate how data merging, standardization, and pruning procedures can assist maximize dataset utility while addressing the paucity of data in medical applications (Durdov et al., 2024).

Dataset construction and maintenance need balancing size and quality. As model complexity and data demands increase, datasets must grow in size and density to prevent overfitting and ensure generalizability. Techniques like crowd-sourced labeling and automated categorisation help manage this data demand by leveraging human input and deep learning models repeatedly to identify enormous volumes of data with high precision, as demonstrated in the LSUN dataset development (Yu et al., 2015).

Figure 2.42: Dataset of Food Ingredients

Source: https://universe.roboflow.com/seongmin/food-ingredients-pnisb/dataset/1/images (*Food Ingredients Computer Vision Dataset by Seongmin*, n.d.)

In this proposed system, a dataset of food ingredients will be constructed. The dataset is collected from all the vegies and fruits as well as other ingredients in our daily life to propose a recipe for the users. Figure 2.12 shows the example of dataset of food ingredients which consists of thousands of ingredients pictures.

2.4.1.3 Tensor Flow

TensorFlow is one of the most comprehensive open-source tools used for building and running modern machine learning systems. It is an open-source software library for machine learning that utilizes data flow graphs and has a wide range of users - data scientists, software engineers, educators as well. In this case, mathematical computations are represented by nodes in the graph and multidimensional arrays of data (tensors) that are moved between them are the graph edges. Such an architecture makes

it possible to describe machine learning practices as a set of interrelated actions. These can be trained and executed on different processors, such as a chip, a CPU, or a TPU, and on different devices, from handheld devices to desktops to powerful servers without altering the code. This means that highly diverse classes of programmers can use the same tools and work together, which significantly increases their productivity. The technology itself was developed by the Google Brain Team to conduct experiments with machine learning and deep neural networks (DNNs), but the design is flexible enough for use in a many other fields as well (What Is TensorFlow? n.d.).

Furthermore, TensorFlow is a software library complex that has been built, especially for numerical computation, through the use of dataflow graphs. It allows easy programming, training, and production deployment of neural networks and other machine learning models. The core technologies are coded in highly efficient C++ and in the CUDA programming model, developed by NVIDIA, which allows code parallel execution on the GPU visual engines. TensorFlow supports APIs in multiple programming languages, with the Python API being the most comprehensive and stable. Other officially supported languages are JavaScript, C++, Java, Go, and Swift, while some other third-party packages have been created to support languages such as C# and Ruby (Pang, B et al., 2019).

The TensorFlow workflow consists of three main components: data preparation, model design, and model evaluation for predicting outputs. The framework accepts data in the form of multidimensional arrays which it refers to as tensors and operates in two main fashions. One of the major approaches is the building of a computational graph which is used in laying out a dataflow for the purposes of training the model. The other, which is often the more intuitive one, is eager execution, which is in line with the

concepts of imperative programming and evaluates operations right away (What Is TensorFlow? n.d).

When using the TensorFlow framework, it is usual to perform training on a computer or in a data facility. In both situations, the action is enhanced by positioning the tensors on the GPU. Once trained, the models may be deployed on various devices, including desktops, smartphones, and even on the cloud (What Is TensorFlow? n.d.).

TensorFlow also offers several supporting features. For example, TensorBoard, which allows users to visually observe the training process, underlying computational graphs, and metrics for purposes of troubleshooting runs and evaluating model performance. Tensor board is the unified visualization framework for Tensorflow and Keras (What Is TensorFlow? n.d.).

Keras can be described as a high-level API that extends TensorFlow. Keras extends the capabilities of TensorFlow by providing a simplified API better suited for building standard models. The fundamental principle driving the design of the API in question is the ability to go from a thought to a concrete result as quickly as possible (What Is TensorFlow? n.d.).

2.4.1.4 Android

Android is a solid, mobile OS that is open source and comprises of OS, middleware, system applications and key applications. Android is suitable for many types of devices, as it was designed to be flexible and scalable and is managed by the Open Handset Alliance, where Google plays a major supporting role (Developers, 2011). The architecture is in layers, with the Linux Kernel forming the lowest layer which provides

functions such as security and memory and process management services, thus allowing applications to run concurrently without interfering with one another and allowing optimal usage of resources (Developers, 2011).

Figure 2.43: The Android component stack

The Android operating system nurtures an ecosystem suitable for development by making available the same framework APIs that are used by the core applications to the developers. This flexibility encourages aspects of application components being used in various applications thus enhancing development. In Gargenta's (2011), Android is defined as an operating system stack for mobile devices that allows phone manufacturers to implement their own applications, if necessary, without affecting the basic distribution. Such an approach facilitates the adoption of the given platform wide arrays of libraries and tools as in the case when there is a need graphics and data storage as well as communication (Gargenta, 2011).

The Android Software Development Kit (SDK) enhances the Android development environment by providing basic tools, libraries and emulators that enable

application testing without the need of physical devices (Gargenta, 2011). The Dalvik virtual machine (VM) used in android is particularly important to the mobile environment, as it is designed to run very efficiently and is memory hungry. The element allows different applications to run in their individual processes, improving the system's robustness and security. Because of its architecture and the flexibility it brings, Android is ideal for building various applications ranging from simple tools to advanced AI solutions.

Below is the table to show the reason that Android is chosen to be considered in this proposed system.

Table 2.12: Reason and Explanation of choosing Android

Reason	Explanation
Open-source flexibility	• Allows customization and easy integration of AI libraries, making it versatile for unique projects.
Extensive developer resources	• Rich libraries and tools in Android SDK, which support image processing and machine learning models.
Wide device compatibility	• Supports a broad range of hardware, ensuring the application reaches a wider audience.

Optimized for mobile devices	• Efficient memory management and resource use, crucial for real-time AI tasks on mobile platforms.
Growing market share	• Large user base increases the potential impact of the application, especially for food enthusiasts.
Community and support	• Extensive developer community and documentation aid in troubleshooting and optimization.

2.4.1.5 Android Studio

The official integrated development environment (IDE) for developing Android applications is Android Studio. The software is built upon incorporating development, debugging tools and code editors from IntelliJ IDE, which is a software development kit primarily for Java programming. The application was initially revealed during the Google I/O event held in May 2013, and in December 2014 the first stable version of the software was made available. Android Studio is compatible with macOS, Windows, and Linux operating systems. The latter has instead UI Design Center of the Eclipse Android Development Tools (ADT). Android Studio as well as the SDK can be gotten from Google (Contributor, 2023).

Android Debug Bridge, mostly known as ADB, is a command-line tool that assists mobile app developers for Android phones to perform a range of operations on a certain machine, one of them being app installation or debugging. Moreover, it encompasses the Android Virtual Device (AVD) which is one more software for developers, allowing them to create and work with Android Virtual Devices. The latter refers to connotative devices that can be used to test the application on various android devices without the need of the physical device (Drexler, 2022).

Figure 2.44: Emulator in Android Studio

Source: https://medium.com/@wijebahuwmpwdgb.20/developers-guide-to-android-emulator-setup-and-secrets-tips-501c44cf25a2 (Wijebahu, 2024)

It is well known that Android Studio has become the de facto Integrated Development Environment (IDE) when it comes to the development of Android applications due to its extensive features aimed at easing the development process. For a project designed around the themes of food ingredient recognition and recipe formulation using artificial intelligence, Android studio has proven to be the best development environment owing to the rich tools, testing options, and cut-throat

collaboration in app development. The table below presents the outlined benefits of using Android in this project (Drexler, 2022).

Table 2.13: Reason and Explanation of choosing Android Studio

Reason	Explanation
Faster Coding	• Real-time live rendering enables quick adjustments, speeding up the development and UI fine-tuning.
Fast Emulator	• Allows fast testing on multiple virtual devices, ensuring compatibility and performance checks.
Comprehensive Testing	• Provides a wide array of testing tools to simulate different devices, ensuring reliable app performance.
Firebase Integration	• Simplifies adding push notifications and real-time data updates, beneficial for user interaction and data sharing.
Feature-Rich Environment	• Includes APK Analyzer, Vector Asset Studio, and Translation

	Editor, all useful for app optimization and localization.
Collaboration Tools	• Facilitates teamwork, enabling code sharing and merging without extra communication hassle.
Native Android Support	• Designed specifically for Android, offering a seamless environment tailored for high-quality Android apps.
Project Templates	• Speeds up initial setup with templates, helping to start projects quickly and with standardized layouts.

2.4.1.6 SQLite

SQLite is regarded as the simplest yet the most efficient database management tool. It has several advancements, features, and functionalities over the other relational databases. Many big MNCs like Adobe incorporate SQLite in their product Adobe's Photoshop Lightroom as the application file format. Airbus, a Europe-based multinational aviation company also employs SQLite in the flight software of its A350 XWB aircraft family. You will learn various concepts and also get to practice them hands-on in this SQLite tutorial (S, 2024).

SQLite is known as a database engine. This software helps users in working with a relational database. In SQLite, unlike other database engines, a whole database is kept in a single file. This enables a high level of portability – duplicating roes does not require any special skills than just photocopying the file that contains the data, and even distributing the database could just be emailing the attachment (Codecademy, 2024).

There is no server installation necessary for the functioning of SQLite. SQLite database gets incorporated within the application accessing it. The application uses the SQLite database through direct storage of data in the forms of database files on the disk and retrieval of the data for use by the application. The following diagram shows the server-less architecture of SQLite (*What Is SQLite?* 2024):

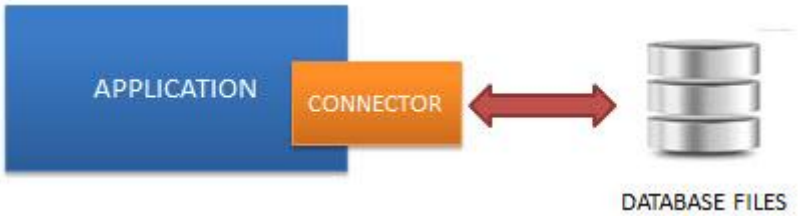


Figure 2.45: Serverless architecture of SQLite

Source: https://www.sqlitetutorial.net/what-is-sqlite/ (What Is SQLite? 2024)

SQLite is a frequently used database management system that is easy to use and implement. Therefore, it is suitable for mobile and embedded applications. Its uniqueness such as being standalone in nature and its ability to work on any Operating Environment (OS) makes it ideal for projects that need lightweight and easily installed storage options. The table below summarizes the pros of using SQLite in managing project databases.

Table 2.14: Reason and Explanation of choosing SQLite

Reason	Explanation
Open Source	• Free to use without any licensing requirements, making it a cost-effective choice.
Serverless	• Operates without a separate server process, simplifying deployment and reducing resource requirements.
Multi-Database Flexibility	• Supports working with multiple databases in the same session, enhancing flexibility and convenience.
Cross-Platform Compatibility	• Runs on various platforms, including macOS, Windows, and Linux, ensuring wide accessibility.
No Configuration Needed	• Requires no setup or administrative tasks, allowing for quick integration and ease of use.

2.4.2 Hardware

The hardware used in the proposed system will be reviewed in this section clearly.

2.4.2.1 ASUS ROG Strix G (G531GT)

Figure 2.46: ASUS ROG Strix G (G531GT)

The ASUS ROG Strix G (G531GT) is a well-furnished gaming laptop that comes with several features capable of supporting heavy workloads such as tasks involving AI and machine learning. To be more specific, one of the most crucial components is the 9th Generation Intel Core i5-9300H CPU, which is a four-core processor with novel turbo boost technology of up to 2.40GHz, thus aiding in supporting multi-processing and heavy calculations. Furthermore, the laptop's primary graphics card is NVIDIA GeForce GTX 1650, which offers adequate graphics even for gaming purposes and AI applications like visual processing or image recognition.

This particular model has a choice of 8 GB to 16 GB of DDR4 RAM, which can support up to 32 GB (currently the RAM of the laptop has been expanded to 32 GB), allowing for better performance when working with larger data and complex

processes. To provide the best possible performance, a 512 GB SSD storage unit has been fitted to the device as well. Furthermore, the ROG Strix G comes equipped with a 15.6-inch Full HD screen with a high refresh rate, making it suitable for detailed graphical work as well as playing videos. This laptop shall also be able to perform for long periods of time comfortably, and even more so for intensive usage scenarios thanks to the ROG Intelligent Cooling thermal solution inside.

The table below shows the reason on the choice of this laptop:

Table 2.15: Reason and Explanation of choosing ASUS ROG Strix G (G531GT)

Feature	Explanation
High-Performance Processor	• The Intel Core i5-9300H with four cores allows efficient handling of AI computations and multitasking.
Dedicated GPU	• The NVIDIA GTX 1650 supports graphic-intensive tasks, including AI model training and image processing.
Ample RAM (Expandable)	• 8 GB to 16 GB RAM, expandable up to 32 GB, enables smooth processing of large datasets and model execution.
Fast SSD Storage	• The 512 GB SSD ensures quick data access, reducing wait times

	when handling large files or datasets.
High-Refresh-Rate Display	• The 15.6" Full HD display with a high refresh rate provides clear visuals, helpful for detailed work and testing.
Efficient Cooling System	• ROG Intelligent Cooling allows for extended, stable performance during intensive tasks.
Portability	• Lightweight and portable, making it convenient for on-the-go development and testing.

2.4.2.2 Huawei Nova 5T

Huawei had announced the release of its latest mid-range smartphone, the Huawei Nova 5T, on the 27th of August 2019 in Malaysia. The Nova 5T by Huawei is priced at RM1,599 in Malaysia and buyers can choose from three available colours: Midsummer Purple, Crush Blue and Black (Lee, 2019).

Figure 2.47: Three Colors of Huawei Nova 5T

Source: https://www.gadgetmatch.com/huawei-nova-5t-specs-price-availability/

(Buduan, 2019)

The most appealing part of the Huawei Nova 5T is probably its spec sheet which, for its price, features really nice internals. The smartphone packs Huawei's flagship Kirin 980 processor with 8GB RAM and 128GB internal storage. Too bad, the internal storage cannot be expanded using the microSD slot, though 128GB is still a pretty good point to begin with.

At the front, the Nova 5T has a 6.26″ LCS screen featuring a small punch hole at the top left corner, which is where the 32MP selfie camera resides. Powering the device is a 3,750 mAh battery, which supports Huawei's 22.5W SuperCharge fast-charging technology. Furthermore, the phone has a side-mounted fingerprint scanner and a quad camera set up at the back. The principal camera assembly includes a 48MP primary camera; featuring a wide-angle lens with an f/1.8 aperture, a 16MP ultra-wide camera, a 2MP macro lens, and a depth sensor of 2MP.

Here's a table outlining its features and why it is suitable for this project:

Table 2.16: Reason and Explanation of choosing Huawei Nova 5T

Feature	Explanation
Powerful Processor	• Equipped with the Kirin 980 processor, the Nova 5T can handle AI tasks and process images efficiently.
Quad-Camera Setup	• Features a 48 MP primary camera with additional lenses for wide, macro, and depth shots, ideal for capturing ingredient images.
Ample RAM	• 8 GB of RAM allows smooth multitasking and better handling of AI-powered applications.
AI Photography Capabilities	• Built-in AI scene recognition aids in identifying food items, optimizing image quality for analysis.
Fast Storage	• 128 GB of internal storage provides sufficient space for storing captured images and app data.

Portable and Lightweight	• Compact and easy to carry, making it ideal for mobile data collection and testing.
High-Quality Display	• The 6.26-inch Full HD+ display is useful for reviewing image details and interacting with AI outputs.

2.5 Summary

Briefly, Chapter 2 reviews the three existing mobile apps on AI recipe generation. All of the benefits and drawbacks have been listed in every section of each review of the mobile app. Table 2.10 has compared each of the mobile apps including the proposed system in every aspect. Moreover, Chapter 2 has an overview of the tools and technologies that will be used in this proposed system. The tools and technologies are separated into 2 categories which are hardware and software. Of course, reasons and explanations are provided as evidence of the suitability of those tools and technologies used

.

CHAPTER 3: REQUIREMENT ANALYSIS AND DESIGN

3.1 Introduction

This chapter describes the methodology employed in preparing the Smart Recipe Generator mobile application. Research methodology refers to the organized and scientific manner to have data collected, analyzed, interpreted, or used for answers to research questions or tests of hypotheses in either quantitative or qualitative manner (Sreekumar & Sreekumar, 2024). Such application of the Agile methodology for this project becomes essential because of its stress on flexibility, iterative development, and incorporating user feedback. Agile makes it simple to break a development of a project into small, manageable increments so that they become adaptable to users' needs within the project lifecycle. The Agile methodology goes in line with this project as it allows continuous improvement and quick changes from feedback from users. The process begins with requirement gathering and analysis continuing with a series of iterative sprints that focus on the development and refining of functionalities like ingredient recognition, recipe generation, and dietary customization. Each sprint involves rigorous testing to ensure that the new added functionality does not cause a problem in overall features of the application. The Agile framework, therefore, allows for ongoing improvements after deployment because the application will now be aligned to the changes brought about by user settings. This chapter gives a breakdown of every stage adopted under Agile in this project to ensure the efficient and user-centered development of the proposed application.

3.2 Agile

Agile software development constitutes a wide selection of software development methodologies based upon iterative development and it involves requirements and solutions being developed through the collaboration of self-organizing and cross-functional teams (Cprime Inc., 2023). Contrary to earlier approaches like the Waterfall model, which had a linear sequence of phases, Agile focuses on making things smaller and more manageable units known as iterations or sprints. The result of these sprints is that they deliver functional increments of the product and make possible continuous feedback and improvements in the project lifecycle (Beck et al., 2001). Agile encourages the involvement and flexibility of users in producing results with every aspect in the process of developing a program. Therefore, it is a fruitful choice for innovatively unexpected, ever-changing projects.

Agile principles are described in the Agile Manifesto, a foundational document put forward by Beck et al. (2001). Its application will imply collaboration with customers, responding to change rather than strictly following a plan, delivering a working program instead of being bound by documentation, and appreciating individuals and interactions as opposed to processes and tools. Underlying all these principles is the recognition that stakeholder involvement is a prerequisite for iterative progress and adaptability in software development. End-user satisfaction and adaptability are Agile's principal concepts to be run as the preferred framework for responding to scope project uncertainty actively and dynamically in development.

Agile comprises various frameworks such as Scrum, Kanban, and Extreme Programming (XP). Though each is peculiar in its own way, they all involve iterative development and flexible adaptability. For this project, an arbitrary Agile approach has been applied, incorporating elements of Scrum to ensure effective management of the

Smart Recipe Generator's development. This has augured well towards meeting the project objectives of integration of ingredient recognition, recipe generation, and diet customization; evaluating user feedback against flexibility.

The agile approach is initiated with the gathering and analyzing of the requirements, which is accomplished closely with the stakeholders and future users. Critical features defined in this phase consist of ingredient identification by using deep learning, recipe generation using the Gemini API, and considering dietary restrictions. The feedback obtained during this phase informs a prioritization for an organized backlog, which is a systematic collection of tasks to help in the development process. Schwaber and Beedle (2002) claimed that maximizing the priority from the user end concerning features that have greater significance can take place during the early phases of the development cycle, thereby improving user satisfaction and project efficiency.

In Agile, development is done through the sprints of different lengths wherein each lasts for two to four weeks. This gives them ample opportunity of focusing on a set of features or tasks thereby enabling them to deliver functional increments of the application. The Smart Recipe Generator, for example, has its first sprint organized around building a user-centred interface. Later, deep learning models will be implemented for the ingredient recognition during future sprints while the integration of Gemini API will give a personalized recipe generation. This process allows for systematic construction, testing, and refinement of components.

The user customer feedback is an integral part of the Agile wherein stakeholders review the functional prototype at the end of each sprint. This feedback loop enables development teams to make corrections whenever necessary and ensures alignment with the user's needs (Cockburn, 2006). Another very important Agile concept is to test

at regular intervals to check for the assiduous integration of new features with already existing components. Continuous testing during development would render minimal error while maintaining the application quality and reliability.

Agile process covers the deployment process and follows up with post-deployment iterations based on input from users to further improve the application. This is so that the Smart Recipe Generator is flexible to changing needs of users and remains functional over time. It has been analyzed that operational iterative enhancements improve all systems over user satisfaction (Highsmith, 2009).

Many advantages are associated with Agile methodology that make it suitable for the Smart Recipe Generator project. First and foremost, it is basically flexible in order to change requirements, so this application can still be developed according to changing user needs. It rapidly provides functional components in multiple iterations so that testing can commence early with user feedback in mind. It also requires greater collaboration among developers, stakeholders, and users, which enhances communication and keeps misunderstandings to a minimum. Furthermore, all components are tested and improved at every iteration, with the aim of producing a robust and user-friendly application.

Thus, agile is not without its fair set of challenges. Its inherent flexibility may create scope creep- where there is uncontrolled augmentation of the project requirements. To deal with this, backlogs are reviewed and reprioritized on a regular basis. Another challenge associated with agility is increased dependency on stakeholders- who may not be available or totally disengaged. Clear communication and regular updates ensure active participation. In addition, Agile demands optimum

resource utilization due to its iterative development process, which is again factored into cross-functional teams for maximum benefit in their working arrangements.

It must be telling that agile also has its own difficulties. Such flexibility is what may cause scope creep: uncontrolled augmentation of the project requirements. However, this selected backlog is regularly reviewed and reprioritized. The other challenge an agile project may face is the increasing dependency on stakeholders, most of whom can't be there or are totally disengaged. Participation, therefore, has to be ensured through clear communication and regular updates. Furthermore, Agile demands optimum resource utilization due to iterative development, which is again factored in cross-functional teams for maximum benefit in their organization.

For the Smart Recipe Generator, Agile methodology makes the project relatively flexible without compromising the user-centered development within it. With flexibility, iterative development, and stakeholder input, the application will be able to deliver solutions addressing issues such as reduction in food waste as well as personal meal planning. Besides, the organized yet flexible approach makes sure that the project meets its intended purposes successfully.

3.3 Requirement

Requirement gathering is a step that is extremely critical in Agile methodology, the rest are building blocks towards the development of the Smart Recipe Generator mobile application. In this step, the major issues are first identified in the present system, the essential objectives defined, and the requirements for the proposed system documented. The issues to be addressed by the Smart Recipe Generator include innovative solutions like real-time ingredient recognition as well as personalized recipe generation and dietary customization while minimizing food waste.

Requirement phase involves a set of activities under which detailed requirements could be gathered concerning what the user needs and what the project needs to deliver. Activities here are typically user interviews, stakeholder meeting, and brainstorming sessions towards an agreement between both clients and developers. Connectivity between all parties needs to take place throughout effective requirement planning to avoid misunderstanding and to ensure that the proposed application meets the actual user needs (Chien, 2020). Requirements have been collected for the project through structured surveys and questionnaires that were distributed among the prospective users. The surveys concentrated on studying user preferences regarding existing recipe or meal-planning applications along with their problems and expectations for the proposed new system.

3.3.1 Analysis on Similar Existing Systems

The existing apps for recipe creation and meal planning are many but with much salient gaps that prove the necessity of a better solution such as the Smart Recipe Generator. A lot of these applications comprise interesting elements like ingredient recognition, personal dietary recommendations, and preventing food waste. However, they do not tend to bring all these features into a single platform.

Another spectacular example is their chef application, an AI invention that allows users to create personalized recipes from available ingredients. It includes ingredient recognition through photo scanning and dietary preference settings. However, it is not void of an undeniable limitation, such as inconsistent inaccuracies in detection for the ingredients and limited usage of credits per day for non-subscribers (ACApplications, 2023). Like DishGen, DishGen offers innovative recipe ideas to prepare an environmentally conscious dish with minimal wastes. However, because it requires complete manual input, with some rather low and challenging credit restrictions for free users, practicality in daily use is limited significantly (Olson, 2023).

MyPantry is a futuristic application that will actually allow identification of ingredients by taking photos and help create a virtual pantry from which one can generate recipes according to the ingredients available. Community shared recipes and guided cooking through a step-style interface can also be found. However, slow ingredient recognition and very high accuracy along with importunate advertisements interrupting recipe generation (Eclipse Apps LLC, 2023). However, since the incorporation of community-shared recipes with guided cooking through a step-style interface, MyPantry has also suffered disadvantages like slow ingredient recognition due to its level of accuracy and very importunate advertisement flooding the recipe generation.

One of the disadvantages of these systems is their dependence on subscription fees or in-app purchases to unlock advanced features. Such a monetization model has restricted access for any user seeking a free or cost-effective solution. Moreover, there are no bespoke features that holistically cover the broad issues of food waste and dietary customization. For instance, filtering recipes based on dietary preference can be done through some applications, but not integrated into real-time ingredient recognition and personalized recipe generation all under one platform.

In addition to this, the applications being considered yield variable levels of efficiency related to ingredient recognition and the production of recipes. Some use AIs to actually detect the ingredients of pictures, but the accuracy level is inconsistent so that many times manual corrections are needed. Such efforts impede users from conveniently becoming engaged, taking more time to come up with recipes. They are thus observed to lack any resources directed towards educating the users on improving food utilization strategies or raising awareness of sustainable cooking practices.

These insufficiencies point out the apparent requirement for a more efficient application of high precision ingredient detection, dietary preference modification and recipe formulation; all organs focusing on food waste problems. The Smart Recipe Generator is designed to eradicate the above hindrances by providing an extremely accurate and efficient ingredient identification function through the deep learning technique. It is also connected to the Gemini API to provided highly personalized recipes of available ingredients and user's preferred inputs. Unlike other software applications, the proposed application will be user-friendly, low-cost, and multifunctional while keeping the priorities of access and convenience ahead.

3.3.2 Analysis on Proposed System

The innovative system of the Smart Recipe Generator is designed to cater to the identified limitations associated with existing applications after a comprehensive study of their features and drawbacks. This advanced solution provides a whole new dimension concerning user experience through an advanced technology application-in deep learning and AI-in ingredient recognition, personalized recipe generation, and diet personalization. Thus, through the application of these technologies, the proposed system aims to deal with some issues like food wastage, limited customization, and a seamless event integration.

With a smartphone camera, ingredient recognition and real-time recipe generation are integrated within the mobile application. It's also possible for recipes to be adjusted according to specific dietary restrictions. Unlike the other systems in existence, this is an all-in-one platform with no subscription fees or in-app purchases, making it more accessible to the wider public. Users can take a picture of the ingredients or enter them manually into the app so that personalized recipes can be generated using the Gemini API. This personalization leads not only to the adequate customization of meal suggestions according to the consumer's preferences but also to better use of all available ingredients to minimize wastage.

The foremost component of the proposed system is the incorporation of deep learning, such as CNNs or Mobilenets, for ingredient-specific recognition. The application uses these models to identify fruits and vegetables through photographs, making it easier for users to generate recipes by classifying these images. The new model is aimed at solving existing inaccuracies and inefficiencies existing in other applications and providing users with real-time interactive feedback and experiences in the application.

Technical prowess aside, the Smart Recipe Generator also emphasizes sustainability and health. The fact that it minimizes food waste through effective use of ingredients means that it too helps support cooking practices that are environmentally conscious. Users may enter information about dietary restrictions or preferences-in terms of example ingredients such as vegan, gluten-free, or low-carb diets, among other things-and have the system return recipes that match those items. Thus, this level of customization provides application services to different users with varying needs to promote a healthier eating habit.

The suggested system would also feature an uncomplicated and intuitive user interface through which the user can access all functionalities. For instance, the user can initiate the process of generating recipes either by selecting ingredients from a virtual pantry or simply taking a picture. The system then produces detailed recipes that entail every step in the cooking process, nutritional as well as estimated preparation times. These results are also available to be saved for future use, thus providing users with a convenient way of keeping track of their kitchen activities.

The Smart Recipe Generator shall be made available on the android mobile platforms, which the user could easily download from app stores to ensure accessibility and usability. This application is very effective even on devices with the most basic AI features. Therefore, this application is into the hands of many. The new system will not provide a barrier in terms of subscription fees or even high-end hardware requirements, unlike other existing solutions available today.

Detailed features and functionalities of the proposed mobile application include:

1. Ingredient Recognition:

- Snap pictures of the ingredients that are found with the camera of the device.
- Add ingredients manually for further flexibility.

2. Personalized Recipe Generation:
    - Creating recipes with taste settings will give personalized culinary preferences like vegan or gluten-free diets.
    - Advanced options include meal type selection, for example breakfast, lunch, or dinner, and cuisine preferences.
3. Food Waste Minimization:
    - Add recipe ideas which can include ingredients reaching the end of their best before date.
    - Also advise ways to best save or incorporate leftover ingredients into new dishes.
4. Dietary Customization:
    - Allow users to set dietary restrictions and adapt recipes accordingly.
    - Include features for calorie tracking and portion control.
5. Recipe Management:
    - Save generated recipes for future use.
    - Categorize recipes based on preferences or meal types.

The Smart Recipe Generator effectively brings almost all extraordinary revolution over the already visible systems, which attract newer and fantastic

technologies directly into user-centric design. The system comprises an innovative solution that is sustainable, personalized, and even user-friendly for complex issues such as meal planning or cutting waste in the kitchen, becoming an important element of a modern household's landscape.

3.3.3 Questionnaire

A questionnaire is an instrument mostly used in data collection, consisting of a systematically organized question set for gathering information, opinions, and feedback from individuals or groups. This project utilizes a questionnaire as a tool in the requirement elicitation process for gathering the essential data that will aid in the formulation of the Smart Recipe Generator mobile application. The questionnaire is created to acquire user needs, preferences, and expectations regarding features such as ingredient recognition, personalized recipe generation, and dietary customization, allowing the system proposed to be made suitable to user demands' expectations.

Thus, Google Forms were made used in disseminating the questionnaire for accessibility and simplicity of the respondents. There were 50 respondents to this survey as part of their input. All respondents were potential users who are interested in applications for meal planning and recipe generation. The questionnaire was developed to obtain data that seek to capture diverse perspectives and its sections were divided into four parts that included open-ended and closed-ended questions. The sections included:

1. **Basic Information** – Collecting demographic data and cooking habits.
2. **Current Practices and Challenges** – Exploring existing meal planning methods and pain points.

3. **Desired Features** – Understanding preferred functionalities in a recipe app.
4. **Feedback and Preferences** – Gathering suggestions and insights for feature enhancements.

The survey questions targeted various issues such as the need for customization of diet, difficulties in reducing food waste, and platforms preferred for using applications. Data sourced from such questions was qualitative as well as quantitative which means having a comprehensive analysis of the entire study. The responses are represented visually using pie charts and bar graphs. These visual presentations are very clear on some of the trends and insights necessary to develop the functional requirements of the system.

3.3.3.1 Demographics

The survey involved the age-wise demographic analysis of the 50 respondents. The selection of the respondents incorporates important considerations because it ensures representation of the entire spectrum of age groups for feedback in developing the envisioned system Smarter Recipe Generator.

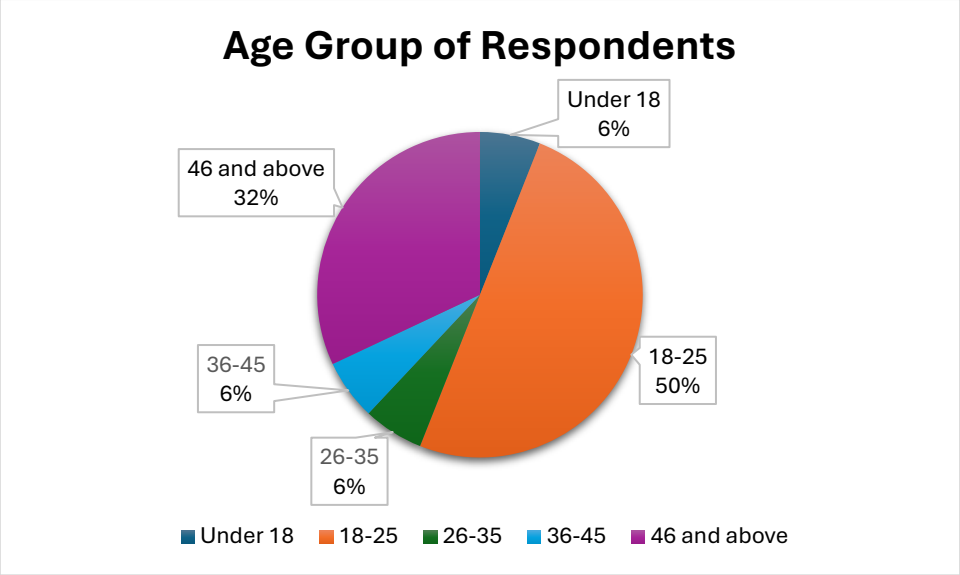


Figure 3.1: Age Group of Respondents

Figure 3.1 shows the age distribution of the respondents. Of the 50 respondents, the largest group consisted of people belonging to the age group 18-25 years, which took the number of respondents to 25 (50%). This was followed by respondents aged 46 and above, which accounted for 16 respondents (32%). Smaller clusters included those under the age of 18, comprising 3 respondents (6%), as well as the groups aged 26-35 years and 36-45 years, each with 3 respondents (6%).

The system appeals to younger adults and older individuals, reflecting its potential applicability to various age demographics. Inclusion of respondents under 18 emphasizes the importance of further features to suit users' needs, personalized dietary options, food waste reduction for households, and so much more.

From Figure 3.1, it very clearly comes out that age groups have been defined visually concerning the survey respondents, further helping to project the composition of the survey population. From this diverse group, feedback ensure that the Smart Recipe Generator is designed to be approachable and easy to operate and effective to a wider range of audience.

3.3.3.2 Cooking Frequency and Use of Meal Planning Apps

To adapt the Smart Recipe Generator to the needs of the respondents, it is important to know their cooking habits and app usage patterns. In the survey, two important questions were asked to know how often the participants cooked and how often they used mobile applications for meal-planning or recipe searching.

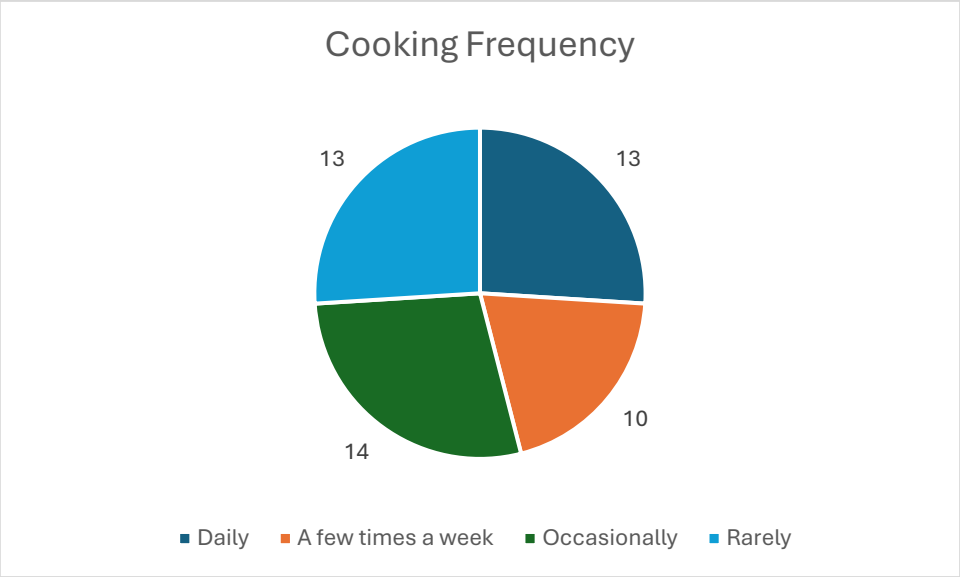


Figure 3.2: Cooking Frequency of Respondents

The query regarding the frequency of cooking has produced a rather evenly distributed response from respondents, as seen in Figure 3.2. Among the 50 participants, 13 of them or 26% indicated that they would cook every day. Thus, the feature is required for organizing the meal preparations. Another 10 respondents, making up 20%, indicated they would occasionally cook a few times a week and could appreciate suggestions for recipes now and then. The biggest group includes the 14, or 28%, who identify themselves as cooking sometimes, while the same number, 13 or 26%, say they rarely cook.

This is how the target audience differs in terms of the cooking habits one would have. People who cook every day or on most days are likely to be interested in functions

such as quick recipes and ingredient searching, as they would help in making their work easier. In contrast, those who cook occasionally or seldom might appreciate creativity or easy-to-follow recipes that inspire users of little experience: that is, the smart recipe generator here should somehow be flexible enough to accommodate this diversity and customize more for users with different cooking experiences.

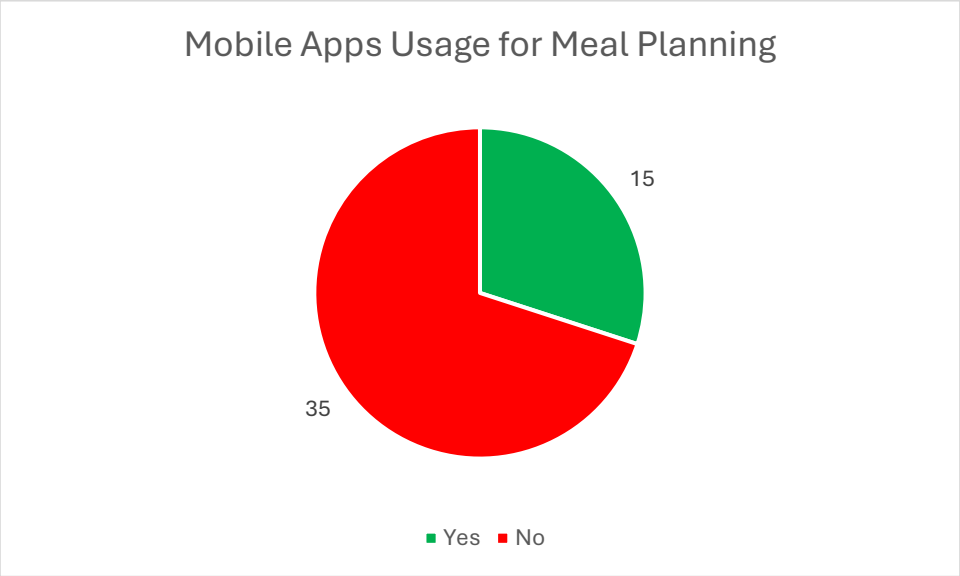


Figure 3.3: Mobile Apps Usage for Meal Planning

When asked if they currently use mobile applications for meal planning or recipe ideas, the majority of respondents, 35 out of 50 (70%), indicated they do not use such apps. Only 15 participants (30%) reported using these applications. This finding underscores a significant gap in the adoption of meal planning tools among the surveyed population, despite the increasing availability of similar apps in the market.

This is a chance for the Smart Recipe Generator to reach out to an audience still largely unreached. Of the respondents, 70% do not use such applications. This group is typically put off by the complicated nature of existing applications, a subscription fee, and a lack of relevant features. The proposed system has the potential to attract non-users into becoming users by concerned areas such as having a friendly interface,

offering free access, as well as personalized recipe suggestions. Meanwhile, users, comprising about 30% of the respondents already using similar applications, will find advanced features and easy integration of ingredient recognition and dietary customization provided by the Smart Recipe Generator.

This wisdom also entails hence creating an all-inclusive design that attracts habitual and occasional use to new users in the meal-planning app. The proposed system will help minimize the gaps: an intuitive, accessible, and comprehensive solution to answer heterogeneous user demands.

3.3.3.3 Meal Planning Habits, Challenges, and Food Waste

Understanding the habits and difficulties of heat respondents around meal planning will be the basis for developing an intervention that addresses their needs effectively. The following three issues were examined with the aim of gathering information on how respondents decide what to cook, the different challenges they face, and their experiences with food waste.

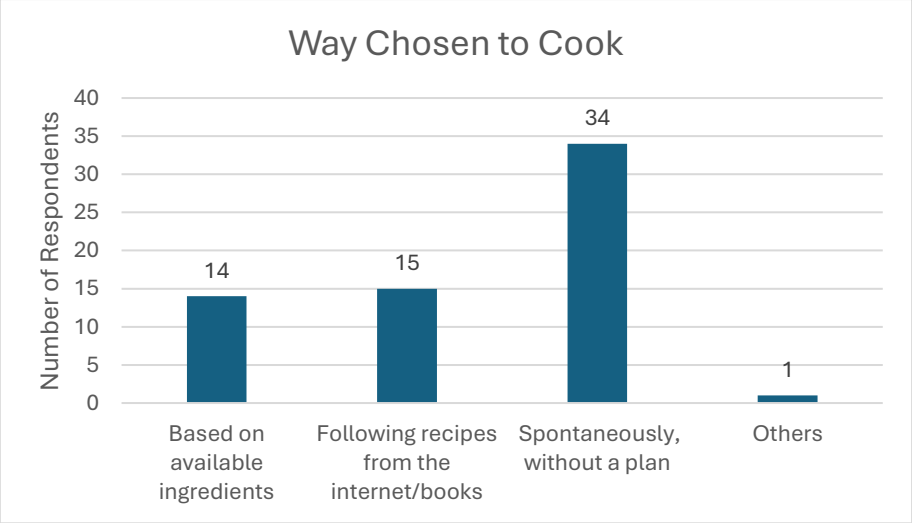


Figure 3.4: Way Chosen to Cook

The survey responses revealed different strategies among participants regarding meal planning. Most respondents, 34 out of 50 (68%), indicated that they decide what to cook at the spur of the moment-so it will look like dependency on-the-minute decision making, oftentimes resulting in ineffectiveness and missed possibilities to optimize utilization of their ingredients. Meanwhile, 15 respondents (30%) would follow through with recipes on the internet and cookbooks. This is a strong indication that these participants are indeed very structured. Another 14 (28%) respondents say they come up with meal ideas according to what ingredients they have left over, which is considered a very resourceful way of thinking in terms of meal planning. Only one other respondent (2%) selected "others," which suggests a different, unspecified mode.

These results underpin the argument for establishing a system which would accommodate spontaneous decision-makers and support the organized planning method as well. The features of ingredient-based recipe suggestions and quick meal ideas will benefit both Hot and Messy users, thus motivating well-organized meal preparation and evoking efficiencies.

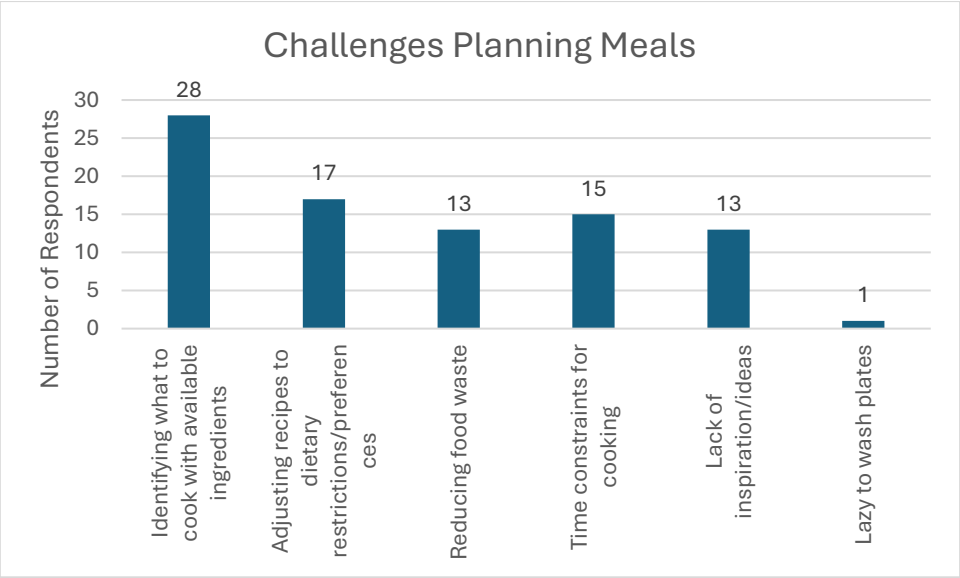


Figure 3.5: Challenges Planning Meals

As claimed by the survey participants, meal planning will have many challenges that it comes with. In Figure 3.5, the most frequently perceived obstacle, discovered by 28 (56%) respondents, was figuring out what to cook using the ingredients they already have on hand. This reinforces the need for a function such as ingredient recognition in the Smart Recipe Generator. Further adding to these 17 respondents (34%) who reported having to readjust recipes to comply with dietary restrictions or preferences, underscoring the need for individualized recipe customization.

15 respondents (30%) noted time constraints as one of the obstacles to effective meal planning. Another 13 (26%) were concerned equally with waste reduction in food. The same 13 said they lacked inspiration or ideas around meals or meal preparation. One (2%) respondent noted "laziness to wash plates" as one factor affecting meal planning.

Such insights strengthened the need to have a tool which besides giving quick and accessible meal ideas could also cater to individual variations in diet and encourage sustainable cooking. The end goal behind such an elaborate framework is the Smart Recipe Generator, generating recipes from available ingredients, dietary preferences, and cooking time.

Figure 3.6: Discards of Food Waste

Through the observation from Figure 3.6, the most mentioned issue among the respondents was food waste, where 28 respondents. This was almost more than half (56%), stating that they sometimes throw food out due to expiration. Another 14 respondents (28%) stated they rarely throw out expired food. Four respondents (8%) stated that they very often face this particular problem. Surprisingly, the remaining 4 respondents (8%) stated that they never throw food away just because it has expired.

It is critical to have good meal planning and ingredient management, as indicated in the results. It includes features, like recipe suggestions based on the ingredients that are nearing their expiration date, reminders on how to consume the food, that can dramatically help users to cut down on wastage. The Smart Recipe Generator, however, aims at sustainable practices and will, therefore, tackle this issue by promoting environmentally friendly and cost-effective cooking habits.

3.3.3.4 Desired Features and Importance of Dietary Restriction Support

This part scrutinizes the preferences of respondents concerning the features of a recipe app and their opinions on the significance of accommodating dietary restrictions. The results will serve as relevant guidance for the prioritization of features to be included in the Smart Recipe Generator.

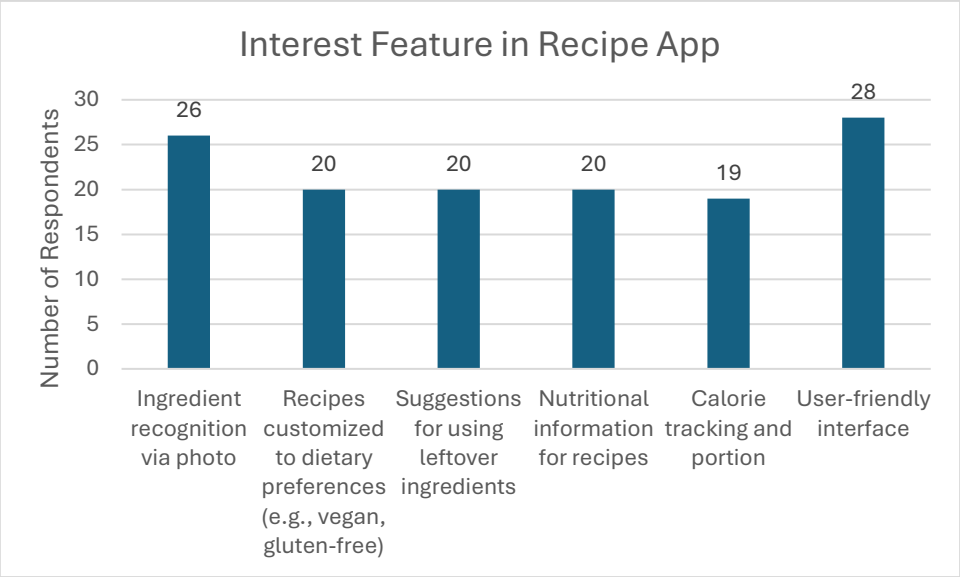


Figure 3.7: Interest Feature in Recipe App

The responses to the survey which is shown in Figure 3.7 have shown a strong inclination toward several important features of a recipe app, with recognition through a photo of the ingredient being the most sought-after among these. This survey has received responses from 26-evidence (52%) participants that value it to be highly useful, thus underscoring the type of application that is likely to simplify meal planning by recognizing what ingredients are available. Recipes tailored to various diets, leftovers ingredient use suggestions, and nutritional information pertaining to the recipes were equally supported, with 20 respondents each (40%) of the total sample selecting those features. This is apparently the result of the increased consciousness

people are associating with health, sustainability, and personalization while choosing meal plans.

Tracking calories and portion control are also noted as important by 19 respondents (38%). This application is meant for people who are following a regime for health management or weight control. This user-friendly function has also been highlighted by 18 respondents (36%), which shows how simple and accessible an app should be in its interface design.

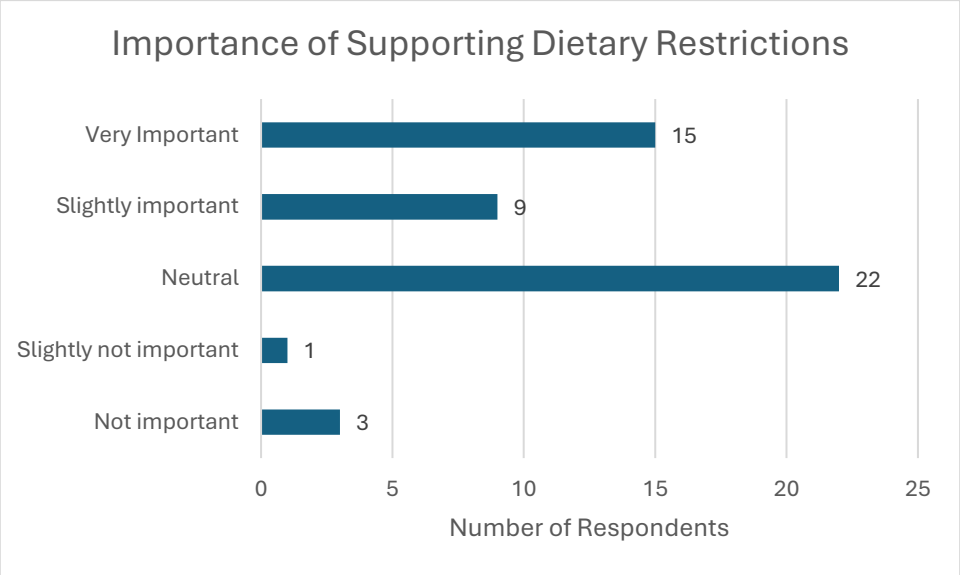


Figure 3.8: Importance of Supporting Dietary Restriction

Respondents had different views on how crucial this function would be in the app, and only 3 respondents (6%) deemed it not important, and 1 respondent (2%) thought it somewhat not important. The rest, however, appeared neutral to positively inclined towards it.

While 22 respondents (44%) provided a neutral response, this would indicate that not every respondent places priority on dietary restriction support; however, it could still remain an important consideration for those who do. Nine respondents (18%)

rated this feature as slightly important while 15 respondents (30%) voted for its very importance; thus, a relatively important number of users find value in such function.

Such findings compel a need for integrating a dietary restriction provision in the Smart Recipe Generator so that its users can access the software concerning dietary requirements or preferences. Including features for vegan, gluten-free, or low-carb recipes gives the app an edge to stay inclusive and entice health-conscious users.

It can thus fulfill the requirements of its diverse target population by concentrating its attention on defined features and diet customization, making it a complete user-centered solution in recipe apps.

3.3.3.5 Platform Preferences, Willingness to Pay, and Additional Suggestions

This study investigates the platform preference of respondents, their willingness to pay for premium features, and suggestions for improvement with the Smart Recipe Generator. This will help improve the development of the app to fit the user with a better touch.

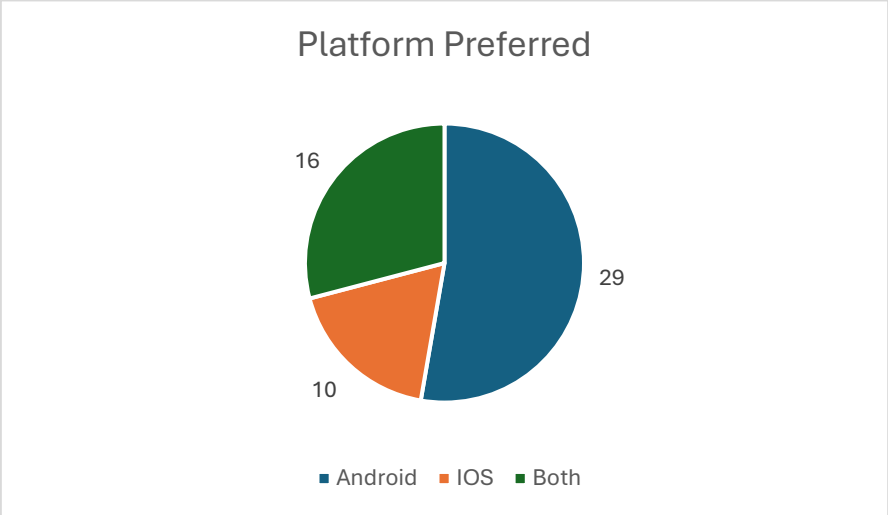


Figure 3.9: Platform Preferred

Survey results in Figure 3.9 shown that Android is the favoured platform for Smart Recipe Generator as preferred by 29 out of 50 respondents (58%). Following this, 16 respondents (32%) voiced their desire for the app to be made available on both platforms while only 5 (10%) preferred it to be exclusively on iOS.

Thus, these findings show that Android should be prioritized in the first phase of development with later additions of cross-platform features to target users of both platforms. Distribution for multiple platforms will facilitate easier access and, thus, increased acceptance of diverse user types.

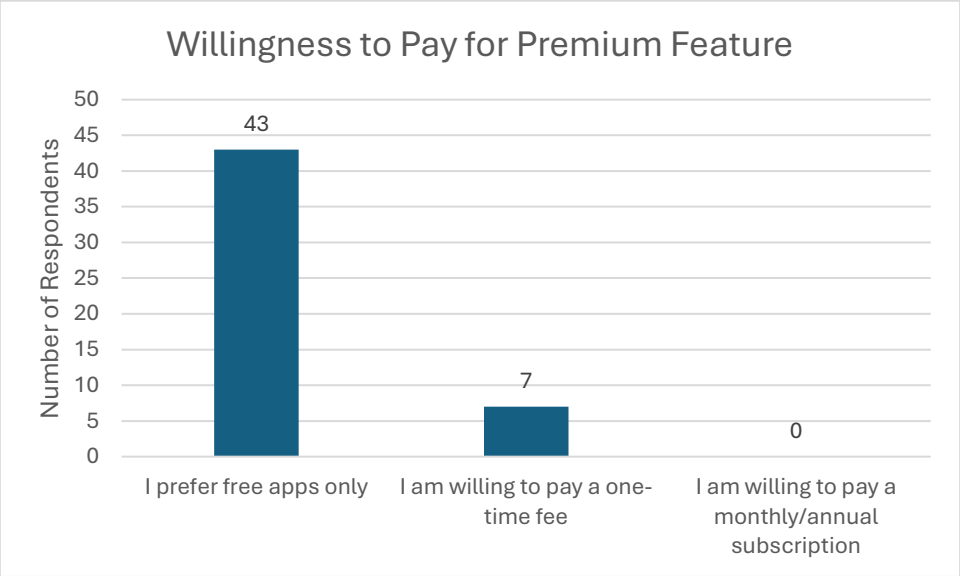


Figure 3.10: Willingness to Pay for Premium Feature

In response to questions designed to know whether the user would be willing to pay for features from advanced recipe generation to ad-free usage, 43 out of 50 respondents, or 86%, declared that free apps would suffice for them. This sends clear indications that there may be some reluctance on the part of the majority of users to be monetized through in-app purchases or subscriptions.

Only 7 respondents (14%) were favourable to the idea of a one-time fee, while none of them (0%) were interested in a subscription model that involved monthly or annual payments. This clearly supports making a value-driven, free application using optional monetization routes such as a one-time payment for additional features for a small group of users.

The last question on the survey asked the respondents to open their minds and express their thoughts and ideas on how to improve the Smart Recipe Generator. Many did not have anything to add, while some gave incredible suggestions that would enhance the app's usability and function. Table 3.1 summarizes the most relevant suggestions and their possible integration into the app:

Table 3.1: Respondents' New Features Suggestions with Concluded Features

Respondents' Suggestions	Concluded Features
"A virtual (AI) cooking assistant that gives step-by-step instructions and handy tips."	Virtual Cooking Assistant with Step-by-Step Guidance
"An interactive shopping feature that checks ingredient availability at local stores."	Ingredient Availability Checker and Shopping Suggestions
"The app should be secured and safe to use."	Enhanced Security Features
"A log system to help users keep track of previous uses."	Recipe and Usage Log System
"Give ideas for using up leftover ingredients."	Leftover Ingredient Utilization Suggestions

"A guide on each step that can be stored for later reference."	Step-by-Step Cooking Guide with Save Feature

Such recommendations tend to exhibit the heterogeneous needs of users- from entry-level features such as a cooking assistant powered by artificial intelligence to extremely advanced functions like a shopping checker and a logging system. These findings yield very useful light for future improvements that will always keep the app fresh and user-oriented.

3.3.4 Software Requirement

For the development of a Smart Recipe Generator mobile application, a variety of software tools and technologies are employed in order to ensure efficiency in its implementation, functionality, as well as user experience. Each software tool is assigned a specific role during the development process, ranging from the designing and training deep learning models to the actual building and deployment of the mobile application. Selected software tools as well as their descriptions are detailed below:

Table 3.2: Software Requirements for the Proposed Application

Software	Description
Deep Learning	Enables ingredient recognition through advanced image processing for accurate identification.
Dataset for Deep Learning	Provides annotated images of fruits, vegetables, and ingredients for training, validating, and testing models.
TensorFlow	A robust framework for building and training neural networks; TensorFlow Lite is used for mobile optimization.
Android	Open-source mobile operating system for developing and deploying the application, ensuring device compatibility.

Android Studio	IDE for building the mobile application, featuring tools for code editing, debugging, and an emulator for testing.
SQLite	Embedded database engine for local storage, enabling offline access to saved recipes and user data.

3.3.5 Hardware Requirement

The whole project designing of the Smart Recipe Generator includes certain hardware requirements for both the proposed Android application and the back-end system. The choice of hardware components is such that their use would facilitate development, testing, and eventual deployment. The following Table 3.3 below provides a concise overview of the hardware components to be employed in the development of the back-end system.

Table 3.3: Hardware Requirements for the Back-end System

Hardware	Description
Intel Core i5-9300H Processor	A quad-core processor with a base speed of 2.4 GHz, capable of handling complex computations in ASUS ROG Strix G (G531GT).
NVIDIA GTX 1650 Graphics Card	Ensures smooth performance for tasks requiring graphical rendering, such as UI previews.

32GB DDR4 RAM	Provides ample memory for efficient deep learning model training and multitasking.
512GB SSD	Offers fast storage for datasets, TensorFlow models, and project files.
64-bit Operating System	Supports the development environment and compatibility with required software tools.

Table 3.4 below provides a concise overview of the hardware components to be employed in the development of the proposed Android application.

Table 3.4: Hardware Requirements for the Proposed Android Application

Hardware	Description
Kirin 980 Processor	Ensures efficient operation of the application, including TensorFlow Lite computations in Huawei Nova 5T.
48 MP Primary Camera	Facilitates high-quality image capture for ingredient recognition testing.
8GB RAM	Allows smooth operation of the application, ensuring a lag-free user experience.
128GB Internal Storage	Provides sufficient space to store application data, including local SQLite databases.

3.4 User Design

User design is an important phase in the development of the Smart Recipe Generator, during which the visual structure and functionality of the system are conceptualized and refined iteratively. This phase links requirements gathering with implementation, to ensure that the product is really what users expect from it. According to Chien (2020), the user design phase comprises the creation of prototypes and mock-ups that clearly represent the architecture, components, and user interface (UI) of the system for all people involved. Such designs serve as a roadmap for the development team and help to better communicate and collaborate among the stakeholders.

The design process of the Smart Recipe Generator is Agile in itself that emphasizes continuous prototyping and active engagement from the users. User feedback is very vital, as pointed out by Egeonu (2022), in this part, hence allowing development for improvement of designs based on real-world insights. Such a method of practice encompasses much of its iterative action where the development cycle identifies any missed gaps in understanding as well as discrepancies in function (Makadia, 2023).

3.4.1 Overall System Architectural Design

Indeed, system architecture refers to the basic framework in a complex system, providing how its components will work together toward the overall goal (Paganini, 2019). It will also serve as the foundation on which a system will be designed and developed, being the conceptual schema from the outset for clarity, efficiency, and scalability. According to Kang (2023), some considerations in system architecture are given to the internal structure of the system, as well as how external entities build

interactions with the system, aiding understanding of the full gamut of components and their interconnections within the system.

The advanced technologies such as deep learning and TensorFlow Lite are embedded within the system architecture of the Smart Recipe Generator for features like ingredient identification and personalized recipe generation. The proposed system can be categorized into two major components, namely, the back end which deals with the model training and data management and the Android application for end-user interaction as well as real-time functionality.

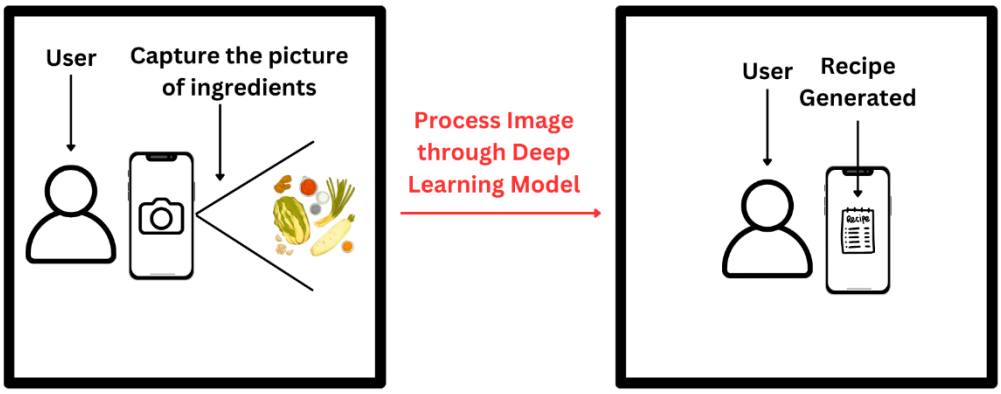


Figure 3.11: Overall System Architectural Design

The data flow of the architecture and its efficient processing-consumed steps are as follows:

1. Data Capture and Input:

    - Users capture images of ingredients using the mobile device's camera, or manually input the data into the system. The image becomes the

input for the system that will undergo preprocessing before any further processing.

2. Model Integration and Processing:

    - Temporal data is processed most recently by the pre-trained TensorFlow model that recognizes ingredients. Formatted into TensorFlow Lite for mobile deployment and maintained high performance-low latency on Android devices.

3. Recipe Generation:

    - Recognized ingredients are fed into the Gemini API to generate recipes according to users' preferences and dietary restrictions. Eventually, the generated recipe is sent back to the user in a structured manner.

4. Data Storage and Access:

    - SQLite acts as the local database that stores user preferences, saved recipes, and even activities done in recent times. Hence, the application stays functional even without an active internet connection.

A proposed system follows a number of sequential steps all the way through from starting up the Android application up until generation of the recipe is complete:

1. Manual entry or capture of image for the ingredients.
2. Preprocessing and feeding the data into the TensorFlow Lite model.
3. Processing by the deep learning model to recognize ingredients.
4. Sending recognized ingredients to the Gemini API for recipe generation.
5. Projecting the recipe along with nutritional values.

6. Saving user usage and recipe data into SQLite for reference later.

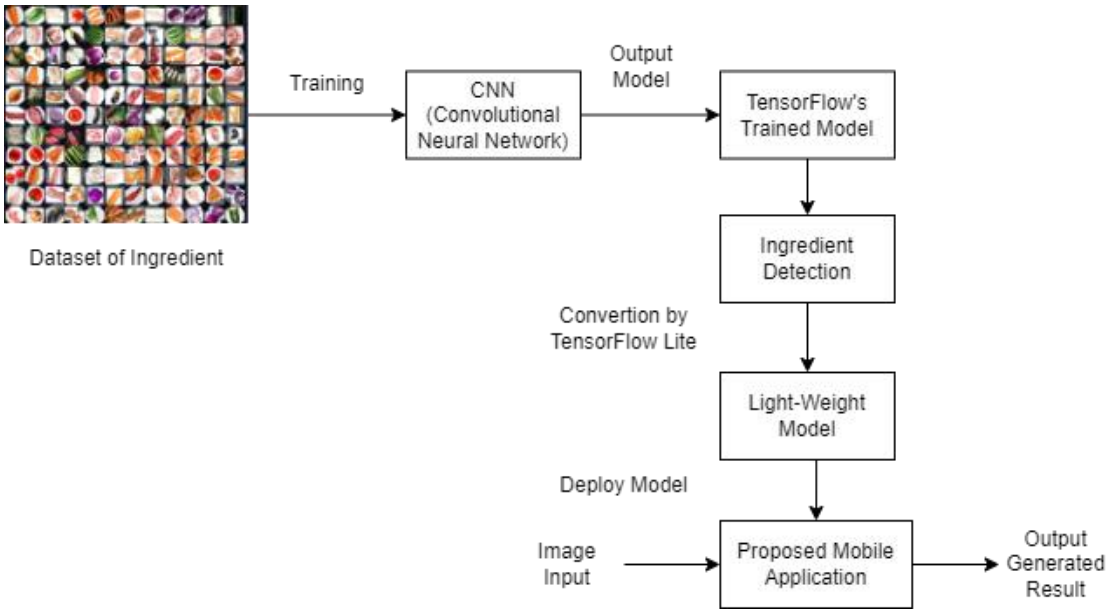


Figure 3.12: General Flow of TensorFlow's Training Phase and its Output

A deep learning training code is built into the back-end system for identifying ingredients with higher accuracy. The image in Figure 3.12 represents the overall flow for such a training phase:

1. Feeding annotated ingredient datasets into the TensorFlow framework.
2. Using a Convolutional Neural Network (CNN) for feature extraction and model training.
3. Evaluating the model's performance and optimizing it for TensorFlow Lite deployment.
4. Converting the model to a lightweight format and integrating it into the Android application.

The continual learning and optimizing of models amply provide with an expression of great accuracy while ensuring efficiency to be perfectly suitable for real-time mobile applications.

This system architecture is meant to redound to the comfort of the user while taking care of very complex back-end operations. With such an application of very modern technologies, for example, deep learning and lightweight mobile frameworks, the Smart Recipe Generator can accomplish many remarkable tasks including even ingredient recognition and individualized recipe production with storage on a local database. Along with diagrams showing the flow of the system, training phase, and data processing pipeline, this gives further clarity of understanding and acts as visual guidance in development.

3.4.2 Scenario Illustration

The scenario illustration pertains, in the view of functionality and use cases, to that of the Smart Recipe Generator. It shows how users interact with the system and gives a better understanding of how that technology can apply to real user needs to guide and develop features. This system will include broadly two major scenarios of user interaction-also ingredient recognition and recipe generation.

I. Scenario 1

In this scenario, the user uses mobile applications to recognize ingredients by clicking an image with a smartphone camera. The procedure follows the steps below:

1. The user sets the camera to receive clear images from the ingredient's point of view.

2. The system pre-processes the captured image and runs it through a TensorFlow Lite model to identify many fruits, vegetables, and other available ingredients.

3. The identified ingredients will be shown on the interface to the user along with some recommendations if the recognition is not sure.

This scenario showcases the function of the system in making the cooking process easier through the provision of the inventory of available ingredients to the users. It emphasizes the relevance of precision image processing and user-friendly feedback mechanisms.

II. Scenario 2

The second scenario describes how the application would generate recipes based on discovered key ingredients and user preferences:

1. After the ingredients are finalized, a user selects preferences such as cuisine type, dietary restrictions, or meal category (like breakfast, lunch, or dinner).

2. The application sends the ingredient and preference information to the Gemini API.

3. The API generates a recipe based on the received ingredients and returns it to the mobile application with all the cook instructions, prep time, and nutritional values.

This scenario illustrates the system's purpose of addressing the different culinary needs while bringing about sustainability in the use of available ingredients.

3.4.3 Web Application and Software Architecture

The Smart Recipe Generator application consists of two main parts: the front end, which serves user interaction and the back end, which does the handling of operations and processing the data concerned. Together, these components form the foundation of the system's architecture, thus ensuring that all functionalities work correctly and users feel at home while using the application. The interaction between these layers could be depicted through a web response-request cycle, which Figure 3.13 shows in detail how the front-end communicates with the back-end to provide services (S, 2024).

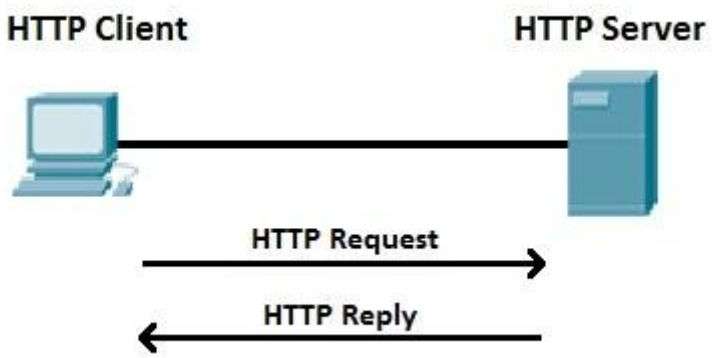


Figure 3.13: Web Response-Request Cycle

The front end of the system refers to the user-facing side, and it provides a user-friendly interface where users can make use of the application. All the necessary things when dealing with the front end deal with designing and implementing graphical user interfaces that would allow the users to navigate in such a way that they can interact naturally with the application. The main interface of this mobile application, known as the Smart Recipe Generator, was developed using Android Studio for building the Android platform. The interface is designed to allow users to capture ingredient images for recognition, input dietary preferences and restrictions and view personalized recipes generated by the application. The front-end development utilizes a combination of

languages and tools which are Java/Kotlin for Android application development, XML to design and structure the app's user interface and SQLite Integration to provide offline functionality for storing user preferences and recipe history locally

The back end is the part of the system which handles the main operations such as processes involving data, models, and mainly their communication through APIs. The back-end notes that Legierski (2021) describes this process as comprising all the developments pertaining to the server-side logic and databases allowing the frontend to function. The Smart Recipe Generator's back end comprises TensorFlow Lite to optimize for mobile devices, this lightweight deep learning framework handles ingredient recognition based on user-provided images, Gemini API Integration to facilitate the generation of personalized recipes based on recognized ingredients and user preferences and SQLite Database to store user data, including saved recipes and ingredient history, enabling offline access. In this way, the communication with the front end will be through HTTP requests on the back end without restricting any data flow. This is the architecture behind the real-time user input processing, executing the recognition model, and returning results instantly.

3.4.4 Flow Chart Diagram

The flow chart is a diagram representation of the work or process, which prudently details the sorts of steps that should be followed in an algorithm or operation of the system (GeeksforGeeks, 2023). For example, Pradiptamu (2023) mentions that flowcharts are much simpler visual frameworks that help for comprehension reasons of processes at the back. The flowchart of Smart Recipe Generator shows the general flow of the system from the interaction of users with a mobile app to the personalized results of recipes.

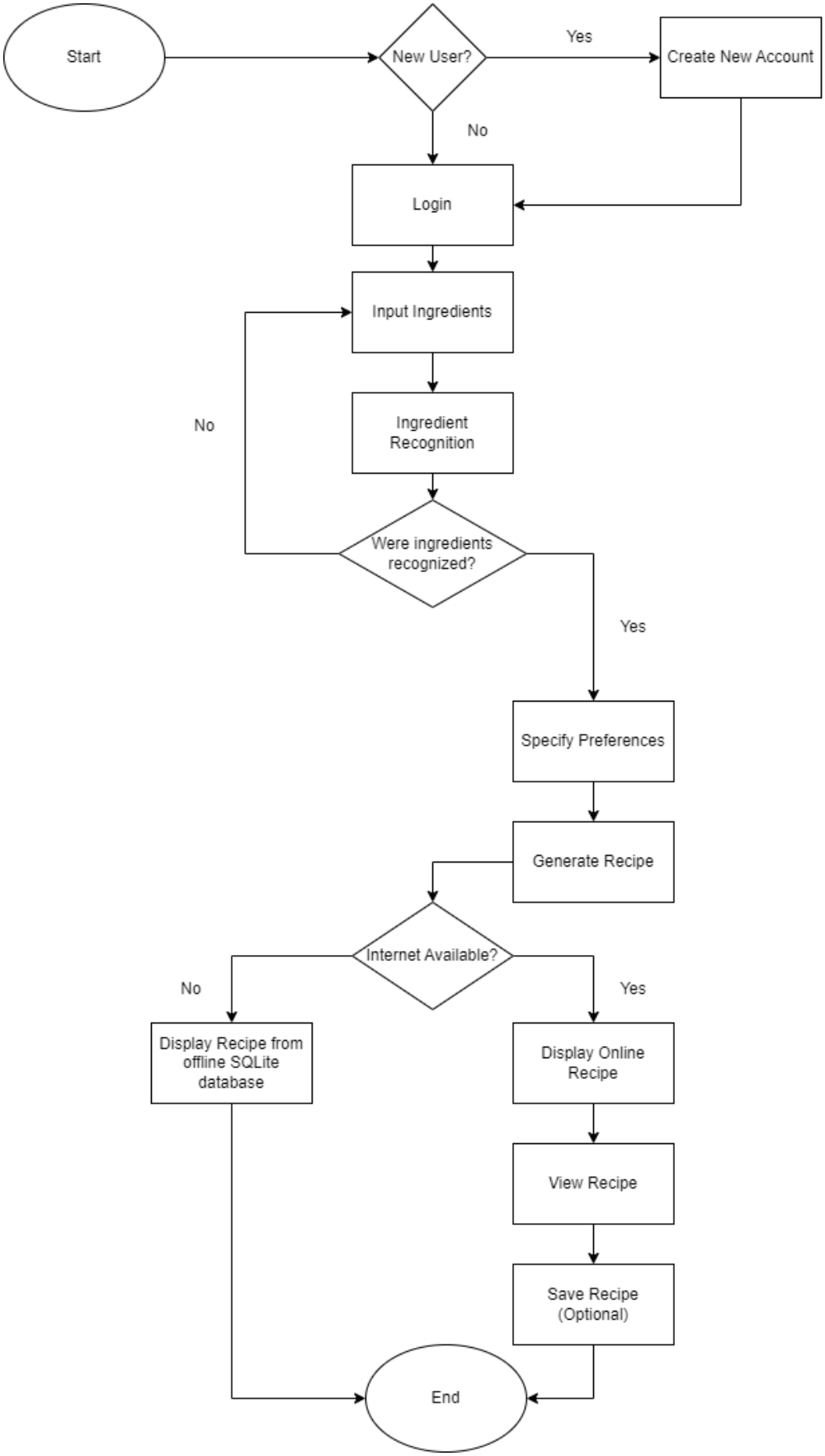


Figure 3.14: Flow Chart

1. Starting the Application

    - Users activate the smart recipe generator on their Androids, wherein it being set up for the users by initializing the system

2. New User

    - If a new user existed, registration is needed, else user can direct login and continue using

3. Input Ingredients

    - Ingredients can be input by the user in several ways

        i. Capture picture of ingredients through phone cam

        ii. Manually key in the ingredients

4. Ingredients Recognition

    - The TensorFlow Lite model takes an image input like any other and demonstrates identifying the ingredients using a pre-trained deep learning algorithm for detection and classification

5. Specify Preferences

    - Users define dietary choices or restrictions (like vegan, gluten-free) as well as the kind of meal (for example, breakfast, lunch, dinner). With all this information, recipe suggestions are created.

6. Generate Recipe

    - The system allows for communication with the Gemini API to generate personalized recipes based on provided or recognized ingredients and the user's specified preferences. The API returns recipe suggestions in structured formats.

7. View Generated Recipe

    - Users view the recipes generated in the application on the mobile interface. The recipes have cooking instructions, nutritional information and suggested portion sizes.

8. Save Recipe

    - The recipes can be saved for future reference, by users, into any locally installed SQLite database. Such a database can access saved recipes offline.

9. End Session

    - Exit the application

3.4.5 Context Diagram

The context diagram that is also referred to as a Level 0 DFD (Data Flow Diagram) provides an overview of the flow of information between the system and its external entities, and is shown diagrammatically (GeeksforGeeks, 2024). It acts as the foundation for understanding the boundaries and interactions of the entire system, Smart Recipe Generator, and that of its environment.

The diagram is about the external entities that interact with the system, the data they exchange, and the central process of the system itself. This establishes the relationships which ensure that the context is clear with respect to the functionality and scope of the system.

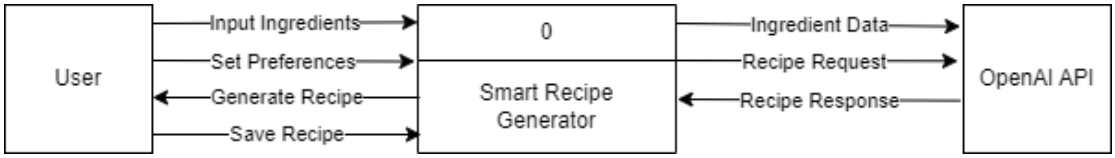


Figure 3.15: Context Diagram for Smart Recipe Generator

There are 4 data flow movements between the system and the user:

- **Input Ingredients**: Users input ingredients via the camera or manual entry, sending raw data to the system.
- **Set Preferences**: Users specify dietary restrictions or preferences.
- **View Recipe**: The system generates and returns personalized recipes.
- **Save Recipe**: Users save recipes to their local database for offline access.

There are 3 data flow movements between the Gemini API and the system:

- **Ingredient Data**: The system sends processed ingredient data to the Gemini API.

- **Recipe Request**: The system requests recipe suggestions based on user preferences.
- **Recipe Response**: The Gemini API returns personalized recipes to the system.

3.4.6 Data Flow Diagram (DFD)

DFD, or a data flow diagram, is a graphical method to express data flowing within a system. This diagram is a clear illustration of the interrelationship between the outside world, processes, and data stores, thus giving clarity in seeing the working of the system as a whole (Robinson & Nolle, 2024). DFDs can vary from overviews to very detailed and even microscopic views making them useful for spotting inefficiencies, improving processes, and enhancing performance systemically.

3.4.6.1 Data Flow Diagram Level 1

A Data Flow Diagram (DFD) Level 1 is a more detailed presentation of the primary processes in the system as an elaboration of the high-level functionality in the context diagram (Chi, 2024). Each of these processes illustrates how the data flow is modified due to the interaction of the external entities, internal processes, and data store. The Level 1 DFD provides more specific operating conditions for the system as a whole and its critical components.

Figure 3.16 describes the Data Flow Diagram Level 1 of the proposed system having eight general processes, namely, Register User, Login User, Manage User Account, Retrieve Saved Recipes, Set Preferences, Display Recipe, Send Ingredients, and Save Recipe. This system has two data stores, Account Details (D1) and Recipe Data Store (D2). Outside sources include the User and the Gemini API.

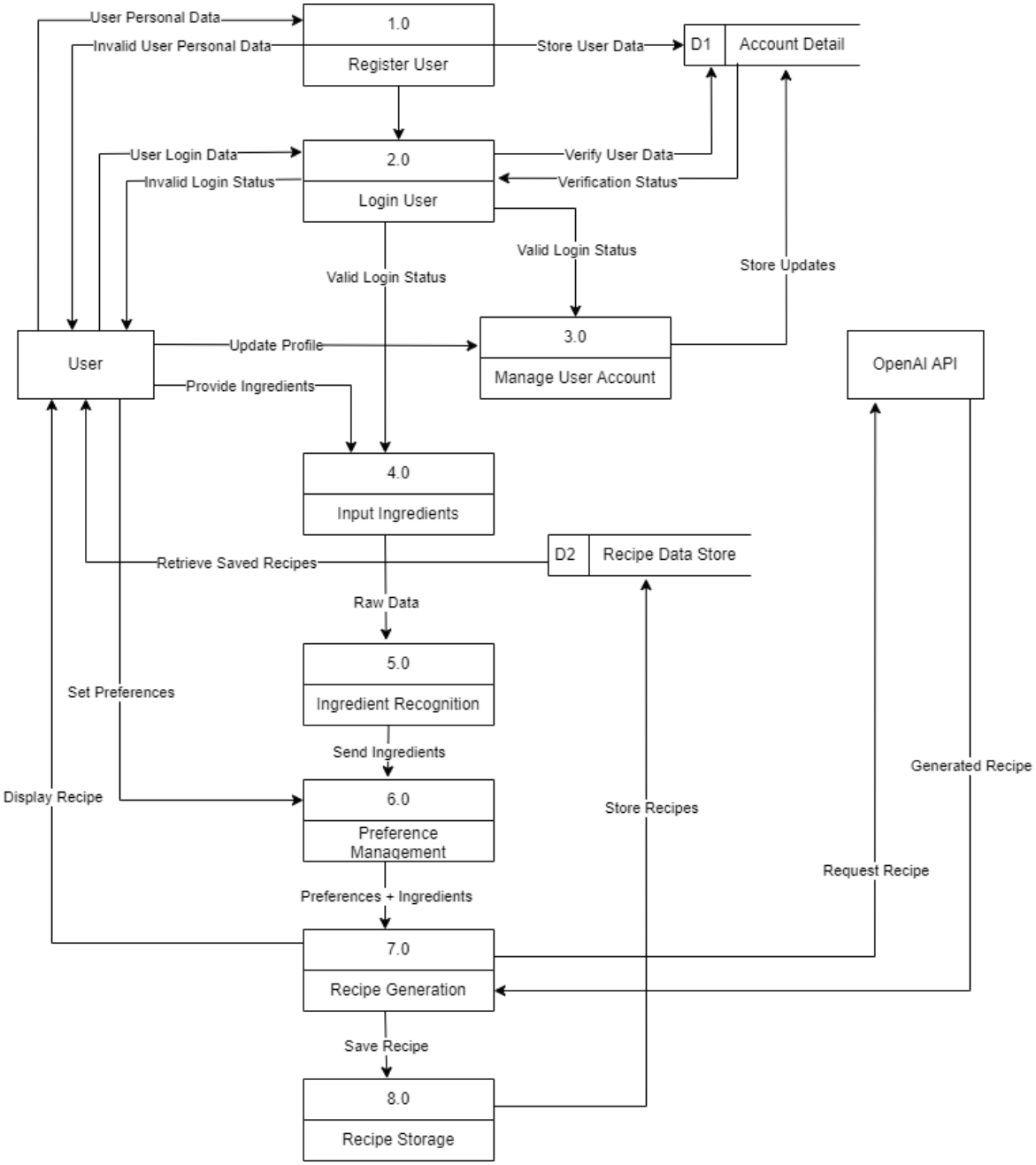


Figure 3.16: Data Flow Diagram Level 1

Process 1.0: Register User

Users create an account to gain access to the system, wherein they should provide personal data such as email, password, and other requirements. User input data are validated by the system, since all fields must also be formatted correctly and will meet

the requirements of the system. If any valid personal data fails, then the user has to re-enter the information. After verification, formatted user data is safely kept in the Account Details data store (D1).

Process 2.0: Login User

The system is accessed by users logging in with their credentials- email and password. These login credentials are compared with stored details in the Account Details data store (D1) for validation. If the credentials are incorrect or empty, the user is informed of this and requested to re-enter the required information. Once successful verification, the user is allowed entry to the system with a valid login status.

Process 3.0: Manage User Account

Users can manage their account after logging in to either change personal information or input new details, such as adding some ingredients to generate a recipe. The updated data will be validated and stored again as a part of the details in accounting detail data market (D1). Invalid updates are rejected with the relevant comments for correction.

Process 4.0: Input Ingredients

Users provide the ingredients by either snapping/ taking a photo of ingredients or manually key in the ingredients existed.

Process 5.0: Ingredients Recognition

System processes input data to recognize ingredients that provided by the users.

Process 6.0: Preference Management

Users can specify their dietary preferences, including restrictions, cuisine types, or cooking methods, which influence recipe recommendations.

Process 7.0: Recipe Generation

When users give a list of ingredients, the system sends the input information to the Gemini API as well as any other user-stored preferences to develop a customized recipe. This is generated from the API and then displayed to the user. The user may also choose to save this recipe for future reference.

Process 8.0: Recipe Storage

It allows users to save any recipes generated within Recipe Data Store (D2) for ease of retrieval later. In this way, the users will have personalized libraries of recipes built on their tastes and ingredients available.

3.4.6.2 Data Flow Diagram Level 2

Data Flow Diagram Level 2 (DFD Level 2) is the level of DFD in which processes are broken down into sub-processes, and though theoretically, DFDs may not be more than level 3, practically they do not. Level 3 DFDs usually contain a detailed description; further breakdowns appear impractical. A DFD Level 2 identifies processes to be further defied from level 1 DFD. Thus, the single process should become a unique

dedicated task. However, it should be noted that not all processes at Level 1 DFD need to proceed to further decomposition. Only those processes that seem to be more intricate processes, where further detail is needed and understanding requires decomposition into Level 2 DFDs.

3.4.6.2.1 DFD Level 2 for Process 1.0

Figure 3.17 below illustrates the Data Flow Diagram Level 2 for Process 1.0, which is the process to Register User.

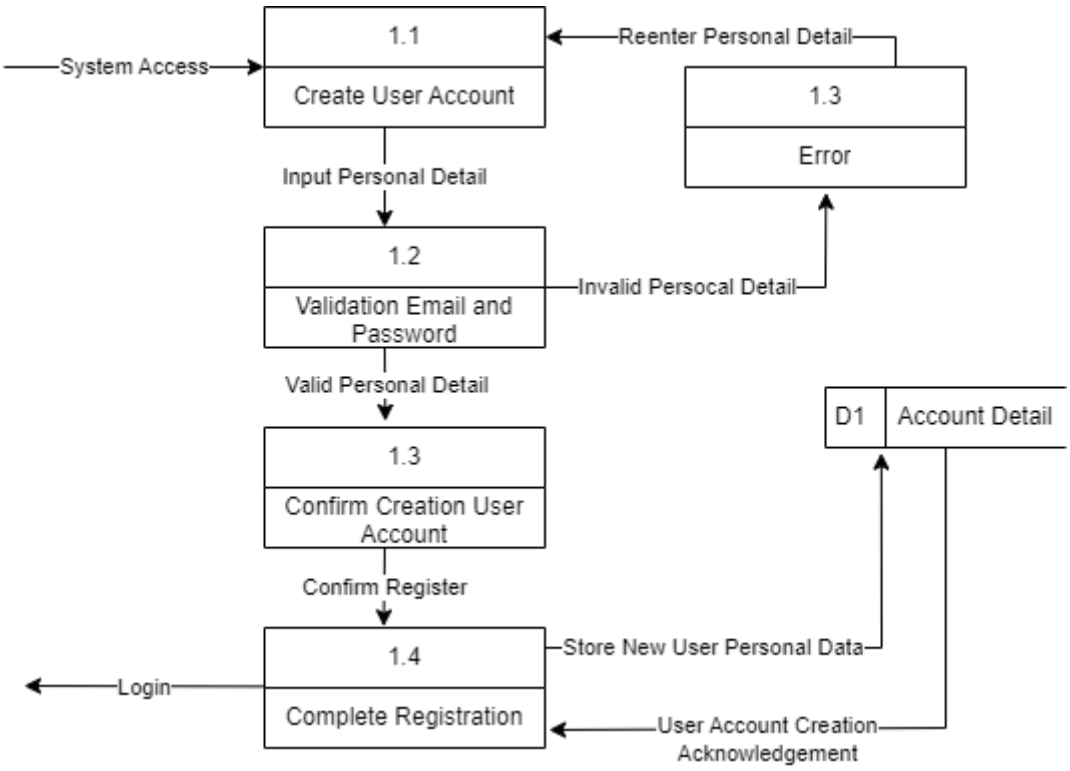


Figure 3.17: Data Flow Diagram Level 2 for Process 1.0

The Process 1.0 describes the part where the user fills a registration form with personal data such as email, password, and other required data, which initiates the account creation process. Then this input is validated to ensure it meets certain criteria, such as whether an email is valid and the password is strong. If any of the entries are invalid, an error message prompts the user for all appropriate corrections and resubmission. After getting through this validation step, the new user's data gets confirmed and securely stored in the Account Details data store (D1); after which, successful registration is acknowledged, and the user is prompted to log in to access their account.

3.4.6.2.2 DFD Level 2 for Process 2.0

Figure 3.18 below illustrates the Data Flow Diagram Level 2 for Process 2.0, which is the process to Login User.

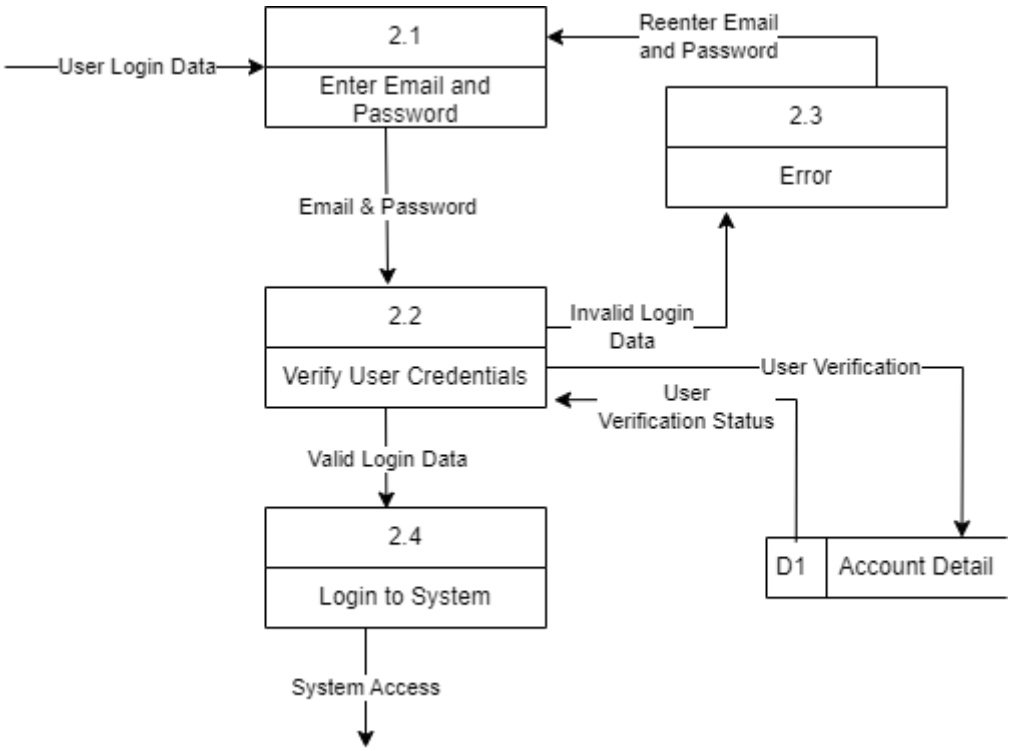


Figure 3.18: Data Flow Diagram Level 2 for Process 2.0

The process 2.0 is initiated through a flow with user login data such as email and password. The email and password issued are verified for the benefit of veracity of user login credentials. Checking algorithms match input credentials to records in the account detail. Then, the user verification status is returned from the account detail. If email and password are invalid, it raises an error message as, Invalid login data to the user. Then the user is re-prompted to enter his-email address and the password is then asked again to try logging in the system. In contrast, if the user verification status is passed, it allows the user to entry into the system, denoting successful login.

3.4.6.2.3 DFD Level 2 for Process 3.0

Figure 3.19 below illustrates the Data Flow Diagram Level 2 for Process 3.0, which is the process to Manage User Account.

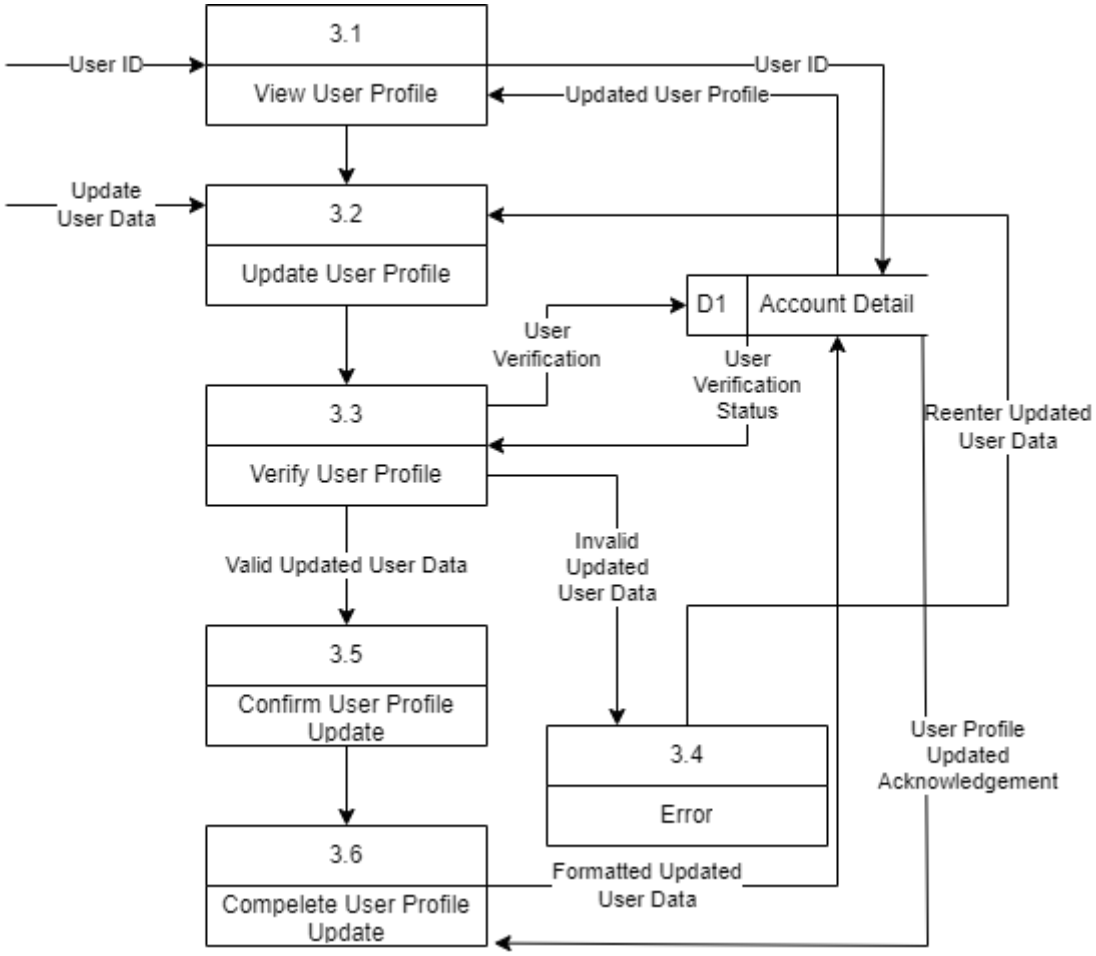


Figure 3.19: Data Flow Diagram Level 2 for Process 3.0

Process 3.0 involves managing user accounts that allow them to view and update their user profiles using which can be their unique user ID as input sources. A user can tap on her profile to get details associated with it. A user ID is used as an input source in retrieving the user profile from the account detail. A user can also edit their profile by tapping the "Edit Profile" button, filling out the remaining fields as necessary, and then clicking save. Systematically, validates input by the user that checks for compliance of the modified detailing against specifications. Spoiled information would generate an

error message and request that the data be re-entered. If the input is correct, then the system will confirm updated user data and close the process for the update of the user profile. The confirmed user data will then go into account detail. On successfully updating the user account, the system will acknowledge the user relevant and notify him or her that the account has been updated. With all these provisions, the user will manage his or her profile on the system more easily.

3.4.6.2.4 DFD Level 2 for Process 4.0

Figure 3.20 below illustrates the Data Flow Diagram Level 2 for Process 4.0, which is the process to Input Ingredients.

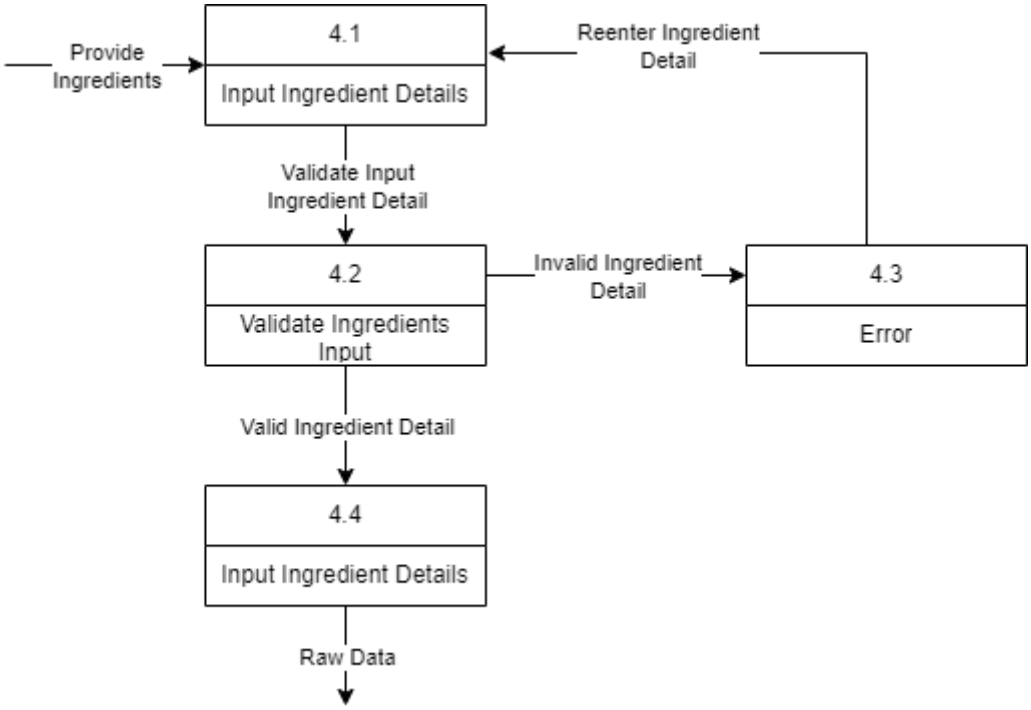


Figure 3.20: Data Flow Diagram Level 2 for Process 4.0

The DFD Level 2 concerning Process 4.0 thus presents how users are entering their ingredients into the system, with an emphasis on the validation depicted in input processing as to ensure that clean and delimitated data is the only thing that moves through.

3.4.6.2.5 DFD Level 2 for Process 5.0

Figure 3.21 below illustrates the Data Flow Diagram Level 2 for Process 5.0, which is the process to Ingredient Recognition.

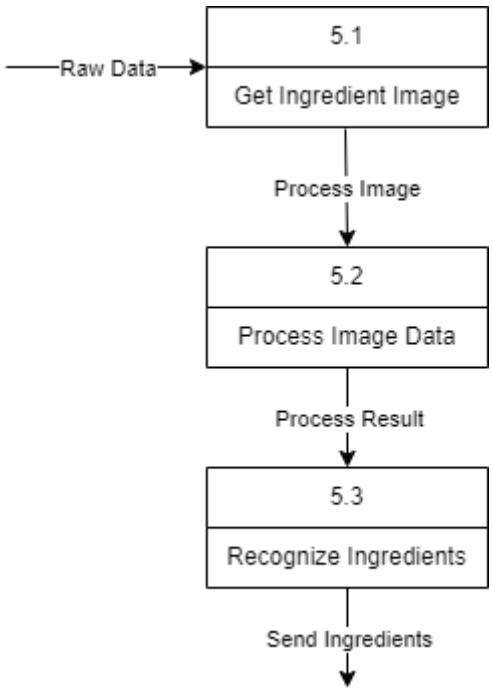


Figure 3.21: Data Flow Diagram Level 2 for Process 5.0

The DFD Level 2 diagram for the Process 5.0 shows the real process of ingredient recognition. This starts when the user uploads their raw data, the ingredient image into the system. After this, the required features are extracted so that the system can identify the ingredient.

3.4.6.2.6 DFD Level 2 for Process 7.0

Figure 3.22 below illustrates the Data Flow Diagram Level 2 for Process 7.0, which is the process to Recipe Generation.

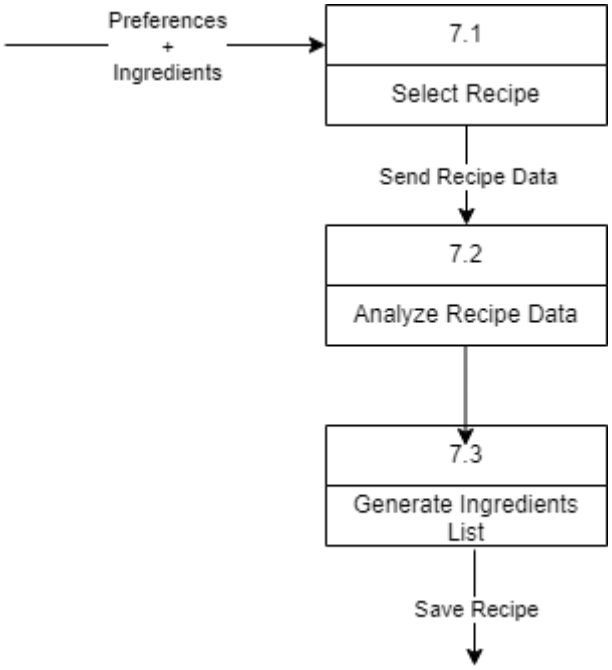


Figure 3.22: Data Flow Diagram Level 2 for Process 7.0

The DFD level 2 diagram for Process 7.0 represents how the system generates an ingredients list for each selected dish or recipe. The process begins with the user selecting one dish or recipe. The next stage is the analysis of the recipe by the system to extract all of the specific ingredients with which the recipe needs to be prepared. Finally, the system compiles the list and sends it to the user in easy-to-read format. This is how the user receives the accurate and complete ingredients list as per his specific requirements.

3.4.6.2.7 DFD Level 2 for Process 8.0

Figure 3.23 below illustrates the Data Flow Diagram Level 2 for Process 8.0, which is the process to Save Recipe.

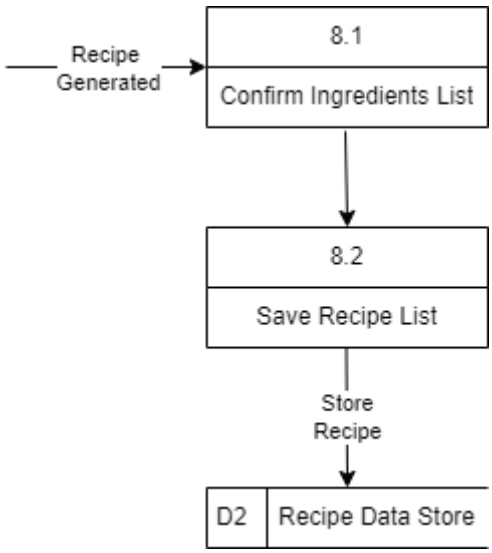


Figure 3.23: Data Flow Diagram Level 2 for Process 8.0

The DFD Level 2 for this particular activity "Save Ingredients" illustrates how the system provides a means for the user to review and save the generated list of ingredients into the Recipe Data Store (D2). The process is initiated when a user confirms the generated list, after which the system saves it securely to the specified store. At the end of the saving process, the system informs the user that he has successively accomplished the saving operation for future access and persistence of data.

3.4.7 Entity Relationship Diagram (ERD)

An ERD or Entity Relationship Diagram serves as visualization tools that represent entities in a system and relationships that these entities have, thus serving as a basis for database design and system analysis (Hanna & Biscobing, 2024). The Figure 3.25 shows relationships between entities in a Recipe Generation System proposed with Crow's Foot notation.

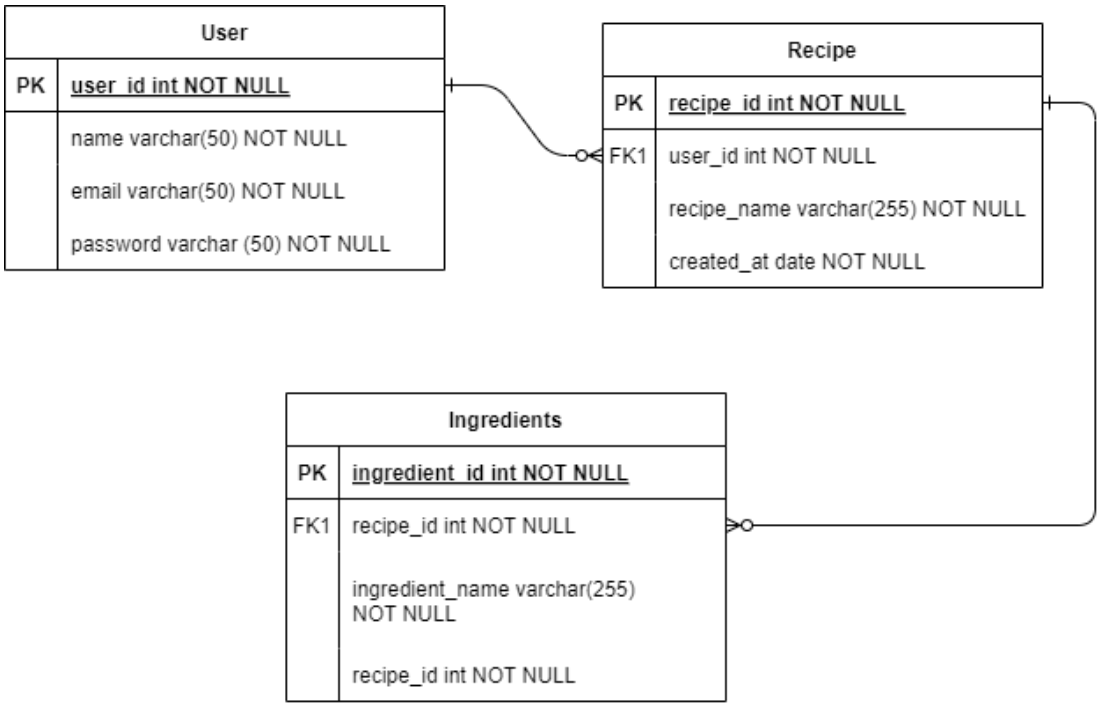


Figure 3.24: Entity Relationship Diagram (ERD)

To illustrate how entities interact within the proposed application, the ERD in Figure 3.24 includes the following key tables:

- User Table
- Ingredient Table
- Recipe Table

The relationships between these tables are described as follows:

- The relationship between the **User Table** and the **Recipe Table** is one-to-zero or many. A user can create zero or many recipes.
- The relationship between the **Recipe Table** and the **Ingredient Table** is one-to-many. Each recipe consists of multiple ingredients, but each ingredient belongs to a single recipe.

A few business rules can be considered as follows:

- A user can create zero or multiple recipes.
- Each recipe can have one or more ingredients.
- Each ingredient is associated with a single recipe.

3.4.8 Data Dictionary

According to Chia (2023), a data dictionary refers to a centralized repository that acts as a source of requisite information regarding the data used within a system, including object names, data types, sizes, and relationships with other data elements. It serves as a guide for referring to context and meaning of each data point for developing and stakeholder to reach a common understanding. The section discusses the data dictionary of the proposed system, which describes data used in detail to augment understanding and consistency in the development phase.

3.4.8.1 Data Dictionary for User Table

Table 3.5 illustrates the data objects inside the User table. The field name, field type, and constraints for the data are listed down in this table.

Table 3.5: Data dictionary for User Table

Field Name	Field Type	Constraints
user_id	Integer (11)	Primary Key, Unique, Not Null
name	Varchar (50)	Not Null
email	Varchar (50)	Unique, Not Null
password	Varchar (50)	Not Null

There are 4 fields in Table 3.5:

- **user_id**: Must be an integer with a maximum of 11 digits. It is the primary key for the User table and uniquely identifies each user.
- **name**: A string with a maximum length of 255 characters. This field cannot be null.
- **email**: A string with a maximum length of 255 characters. Must be unique and cannot be null.
- **password**: A string with a maximum length of 255 characters. This field cannot be null.

3.4.8.2 Data Dictionary for Recipe Table

Table 3.6 depicts the data objects such as field name, field type, and constraints inside the Recipe Table.

Table 3.6: Data dictionary for Recipe Table

Field Name	Field Type	Constraints
recipe_id	Integer (11)	Primary Key, Unique, Not Null
user_id	Integer (11)	Foreign Key (references user_id in User Table), Not Null
recipe_name	Varchar (255)	Not Null
created_at	DateTime	Not Null

There are 4 fields in Table 3.6:

- **recipe_id**: Must be an integer with a maximum of 11 digits. It is the primary key for the Recipe table.
- **user_id**: Must be an integer that serves as a foreign key referencing the user_id field in the User table.
- **recipe_name**: A string with a maximum length of 255 characters. This field cannot be null.
- **created_at**: Captures the date and time of recipe creation. This field cannot be null.

3.4.8.3 Data Dictionary for Ingredients Table

Table 3.7 illustrates the data objects inside the Ingredients table. The field name, field type, and constraints for the data are listed down in this table.

Table 3.7: Data dictionary for Ingredients Table

Field Name	Field Type	Constraints
ingredient_id	Integer (11)	Primary Key, Unique, Not Null
recipe_id	Integer (11)	Foreign Key (references recipe_id in Recipe Table), Not Null
ingredient_name	Varchar (255)	Not Null
quantity	Varchar (50)	Not Null

There are 4 fields in Table 3.7:

- **ingredient_id**: Must be an integer with a maximum of 11 digits. It is the primary key for the Ingredient table.
- **recipe_id**: Must be an integer that serves as a foreign key referencing the recipe_id field in the Recipe table.
- **ingredient_name**: A string with a maximum length of 255 characters. This field cannot be null.
- **quantity**: A string with a maximum length of 50 characters to specify the ingredient quantity. This field cannot be null.

3.4.9 Wireframes

Wireframes are basic graphic representations of a system's structure that are used for designing purposes and layout in planning the functionality of an application or a website (Bruton, 2022). They give an outline of the interface elements expected to exist on key pages, focusing more on the content location and pathways for user interaction. Wireframes normally help early stages of development for getting feedback from stakeholders and ensure that user needs are addressed before final implementation. The proposed system employed high-fidelity wireframes to represent an interface design of both the mobile and web applications showing the most important features and navigation flows.

3.4.9.1 Wireframe for User Registration

Figure 3.25 presents the designed interface for the registration or Sign-Up page within the proposed application.

Figure 3.25: Wireframe for User Registration

Once the user downloads the app from Google Play Store, user can access the application by just taping on the icon of the app. First time new user, the app will direct bring to the user to the sign-up page. New users need to fill in his/her own valid email address and password to complete sign up. Once the user finish filling the email and password, user can direct press the “Sign Up” button and they will be led to the login page. If the users already own their accounts, they can direct go to the login page by taping on the “Login” word.

3.4.9.2 Wireframe for User Login

Figure 3.26 presents the designed interface for the Login or Sign-In page within the proposed application.

Figure 3.26: Wireframe for User Login

After the user finish the registration or already own their accounts, all the users need to do is just entering their valid email address and password. After that, by pressing button "Sign In", users will be led to the main page of the system. Same as Sign-Up page, if users need to create a new account, there is a word "Sign Up" which can be tapped and lead the users back to the Sign-Up to register a new account.

3.4.9.3 Wireframe for Main Page of the Application

Figure 3.27 presents the designed interface for the Main Page within the proposed application.

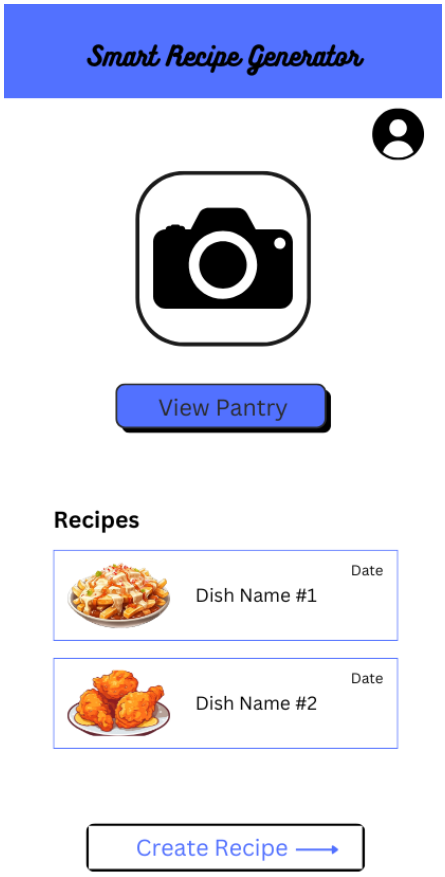


Figure 3.27: Wireframe for Main Page of the Application

Inside the main page, the icon of camera can be tapped to be led to the section of snapping photo of ingredients. Next, the "View Pantry" button will lead the users to the page that what ingredients have been included. The user icon at top right corner is allowing the users to view and edit their profile. The section of "Recipes" is the history of saved recipes by the users, users are allowed to view offline if save the recipe that interested. The "Create Recipe" button will lead the users to the section of creating the recipe.

3.4.9.4 Wireframe for User Profile

Figure 3.28 presents the designed interface for the User Profile within the proposed application.

Figure 3.28: Wireframe for User Profile

For the user profile, users can view their name and email address. By tapping the edit at the middle right of the page, users will be led to the edit user profile page. Besides, users are allowed to logout and back to the Sign-In page as well as delete account.

3.4.9.5 Wireframe for Edit User Profile

Figure 3.29 presents the designed interface for the Edit User Profile within the proposed application.

Figure 3.29: Wireframe for Edit User Profile

After users are led to the edit user profile page, users are allowed to edit their name, email address, password. Then, users can save their changes by pressing the "Save" button.

3.4.9.6 Wireframe for Snap Photo of Ingredients

Figure 3.30 presents the designed interface for the Snap Photo of Ingredients within the proposed application.

Figure 3.30: Wireframe for Snap Photo of Ingredients

By tapping the camera icon, users are allowed to take the photo of the ingredients. It will automatically recognize the ingredients through the snapping of those ingredients' photos.

3.4.9.7 Wireframe for the Pantry

Figure 3.31 presents the designed interface for the Pantry within the proposed application.

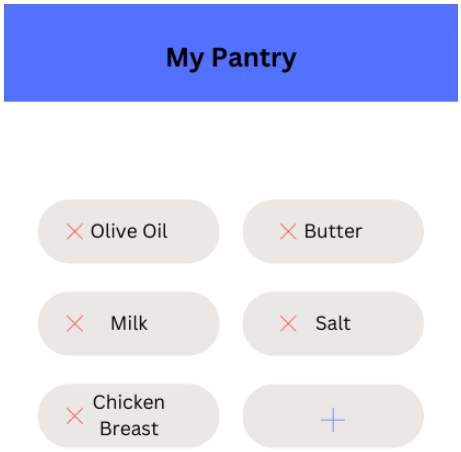


Figure 3.31: Wireframe for the Pantry

Inside “My Pantry”, users can view what ingredients have been recognized in the pantry. Besides, users can edit the Pantry delete the ingredients by tapping the “X” sign as well as users also can manually add the ingredients by tapping the “+” sign.

3.4.9.8 Wireframe for Create Recipe

Figure 3.32 presents the designed interface for Create Recipe within the proposed application.

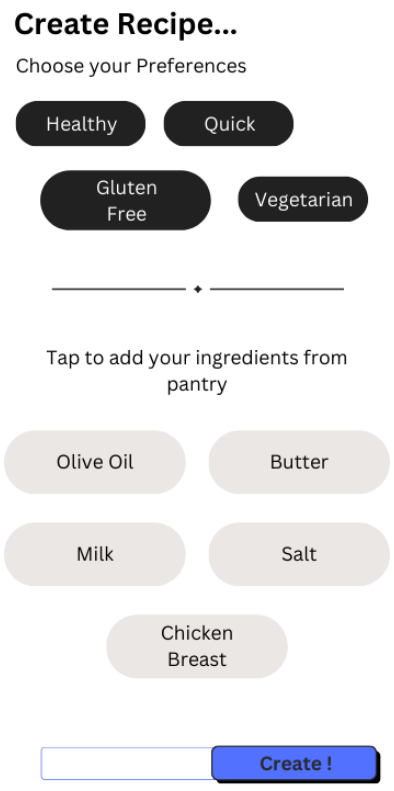


Figure 3.32: Wireframe for Create Recipe

When creating the recipe, users are allowed to manage their preferences on their dishes by selecting predefined preferences. Next, the users have to choose any ingredients from the pantry to be cooked as a dish. If anything needed to be added, users are allowed to input those requirements at the box below. Last, users can press the button “Create!” to create a recipe.

3.4.9.9 Wireframe for Recipe Generation

Figure 3.33 presents the designed interface for Recipe Generation within the proposed application.

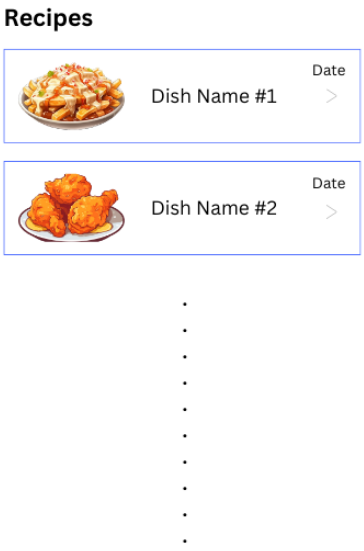


Figure 3.33: Wireframe for Recipe Generation

Some recipes based on the users' preferences will be generated in this page. The dish pic and name will be shown too. Besides, the date generated will be shown too for the users to recall back which recipe have generated before.

3.4.9.9 Wireframe for Recipe

Figure 3.34 presents the designed interface for Recipe within the proposed application.

Figure 3.34: Wireframe for Recipe

This page will show the calories and total time taken to be cooked besides of dish pic and name. Inside the ingredients, users are allowed to change the number of servings and the quantity of ingredients needed will be changed too depends on the number of servings. After that, the steps to cook the dish will be shown below. Users are allowed to save the particular recipe by pressing the "Save" button or just leave the page by pressing the "Done". Both will lead the users to the main page again, the only difference is if the users have saved the ingredients, the saved ingredients will be shown at the section of "Recipe" at the main page.

3.5 Construction

The construction phase is when the actual development and coding for the system occurs to convert the designs and prototypes into a built application (Chien, 2020). This phase primarily focuses on iteration, during which new parts and amended components may be assimilated based on feedback from the clients and end users. Iterative prototyping thus ensures that the development team meets the user issues and clients' requests in a much shorter time compared to the traditional method of building the working model.

During this process, applications will be thoroughly tested to ascertain that it meets the required functionality and expectation of users. Close coordination-oriented people, include developers working with stakeholders, as clients will bring in perspectives, make suggestions, and raise creative ideas on how emerging issues may be solved. Delivering the best possible working version of the software while still being able to respond to new needs will remain priority tasks.

3.6 Testing

The testing phase is important to make sure that application meets all its requirements and works accordingly. The details of the various levels of testing carried out on the system to verify its correctness, reliability, and usability have been compiled in a development report (Glaschenko, 2023). The four main testing phases included in the evaluation are: Unit Testing, Integration Testing, System Testing and Acceptance Testing.

Unit testing is the first stage in testing modules of a software application when all the singly written functions are introduced into the code. They are then checked for

any possible syntax errors, runtime errors, and interface inconsistencies. This whole process is often referred to as white box testing, wherein the internal structure of the system is made available to the testers to examine it closely.

Next is integration testing to check how well the various modules of the application can communicate with one another. This checks whether the user interface is well integrated with the back-end functions to ensure that the data can easily flow back and forth across the components without a hitch.

During this phase, the whole application is tested against the entire set of requirements prescribed for it. It tests whether the application can perform its overall functionality and checks for the highest standards of performance, security, and usability.

Acceptance tests are left to the end-user/stakeholder involvement. This is also called black box testing because it looks at the software from a user's perspective while being required not to know the open code inside. Acceptance testing decides whether the application is good for release based on the evaluation of its usability, user experiences, and satisfaction.

After these stages of testing are completed, a detailed summary report for the test is prepared, including results from test iterations, comments from users, and any constraints or areas for improvement. This iterative approach ensures that the application is released into the real world and optimized fully while tuning to user expectations.

3.7 Deployment

The last phase of system development is the deployment phase. This is when the ready-for-production application goes live and users start accessing it. At this point, the launched mobile application is to be available to all intended users in Google Play Store. The application moves from dummy data to real-life data and thus gives a better indication of how it will perform in the real world (Chien, 2020).

Technical and user documentation accompanies deployment. This documentation serves as a source of development for understanding, using, and handling the application by multiple stakeholders, end-users, and developers. Deployment is inevitably followed by maintenance that practically aims for periodic updates and periodic benefits to be received by the application through improved overall conditions over time.

In addition, the application might provide training for the end users to clarify any confusion regarding the application. Thus, this effort can facilitate the transition of the organizations into a real-life application, ensuring maximum productivity from the application, while minimizing problems that may be encountered in the use of the application itself. Moreover, backup takes place to back up user data in order to enhance reliability to the system.

3.8 Summary

Agile methodologies are topics of discussion in chapter 3. The requirements phase was acknowledged through the use of questionnaires which gather user input and help identify existing challenges. Architectural drafting, which included system architecture, flowcharts, DFDs, ERDs, and wireframes to visualize the system, was one of the key elements produced during the design phase. The construction was carried out through iterative development and subsequent refinement based on user feedback, testing coverage for all functionality and usability, and finally deployment changed the implementation from test into real-world use, supported by training and maintenance plans.

CHAPTER 4: IMPLEMENTATION

4.1 Introduction

In this chapter, the implementation phase of the proposed system is presented, which was developed using the analysis of requirement and design concepts provided in Chapter 3. Construction of the application using stated software tools, technologies, and hardware resources is described. The entire implementation process will follow the Agile method, where it stresses iteratively building, testing, and refining. Additionally, this chapter records front-end and back-end development and integration of core functions such as ingredient recognition and recipe generation. It acts as a complete outline of the steps that were taken in creating and preparing the system for deployment.

4.2 Installation and Configuration of Smart Recipe Generator

Various planning software applications need to be set up in order to start developing the proposed application.

4.2.1 Installation and Setup of Android Studio

Installation of Android Studio IDE

Android Studio, as the Integrated Development Environment (IDE), is very important in Android application development helping with two major functions. It can be considered a powerful code editor because it provides such support as syntax highlighting, code completion, and error-checking in real time while you write your code. On the other hand, it includes important development components such as the

Android Software Development Kit, the SDK Command-line Tools, and the SDK Build Tools that support the building, testing, and deployment of Android applications.

Android Studio can be downloaded from the official website with the link of https://developer.android.com/studio. A button of "Download Android Studio Meerkat Feature Drop" will lead users to a Term & Conditions page as shown in Figure 4.1. The download of Android Studio will be started once users agree the T&C and clicking on the download button as shown in Figure 4.2.

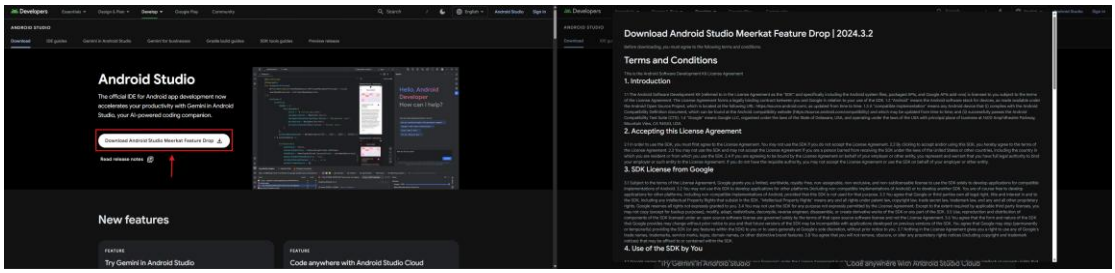


Figure 4.1: Term & Conditions Page

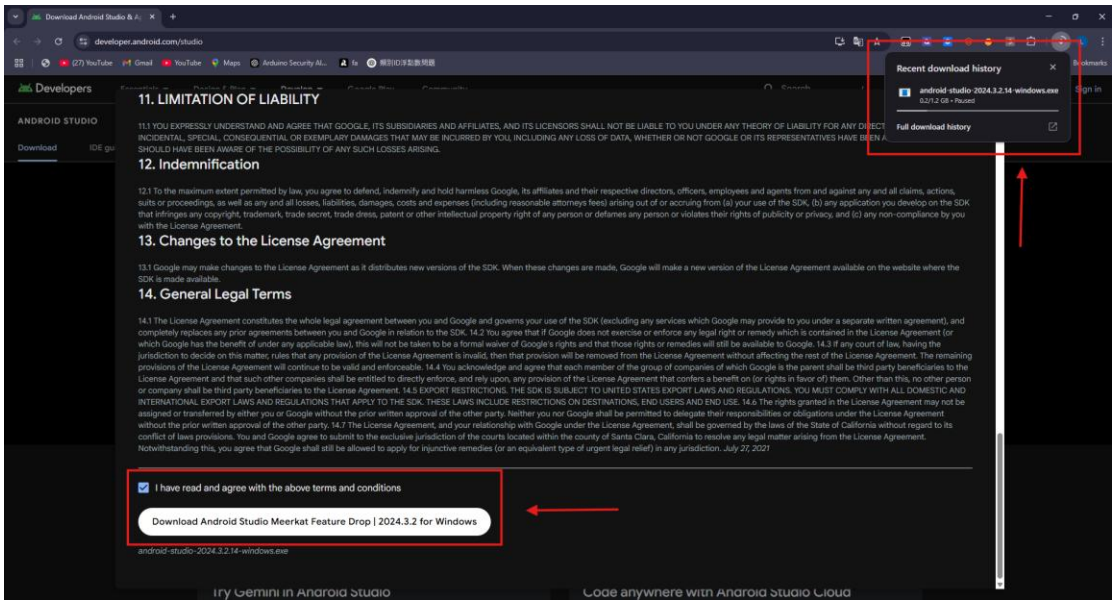


Figure 4.2: Start Downloading Android Studio

Once complete downloading, an exe setup file will lead the users to then Android Studio Setup. Users just install the programme with default setting by clicking the "Next" button to go ahead with the process as shown in Figure 4.3.

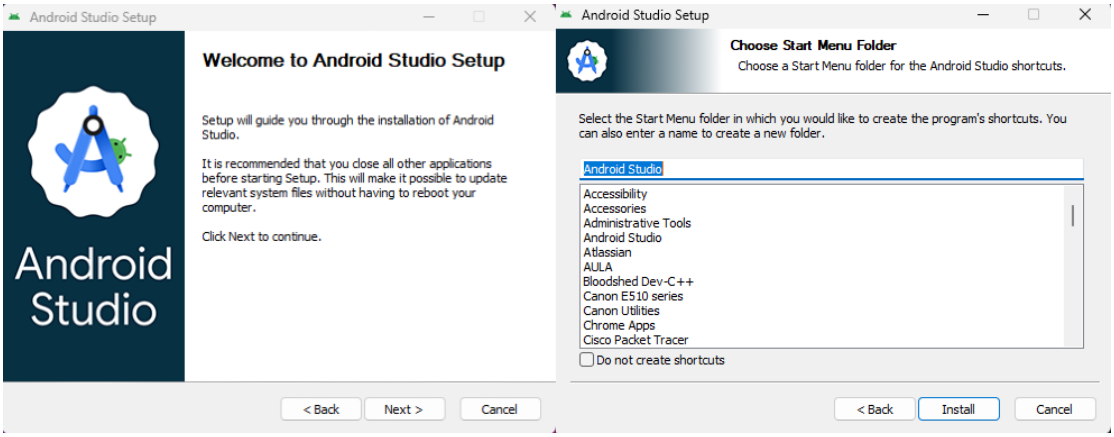


Figure 4.3: Android Studio Setup

Creation of Project

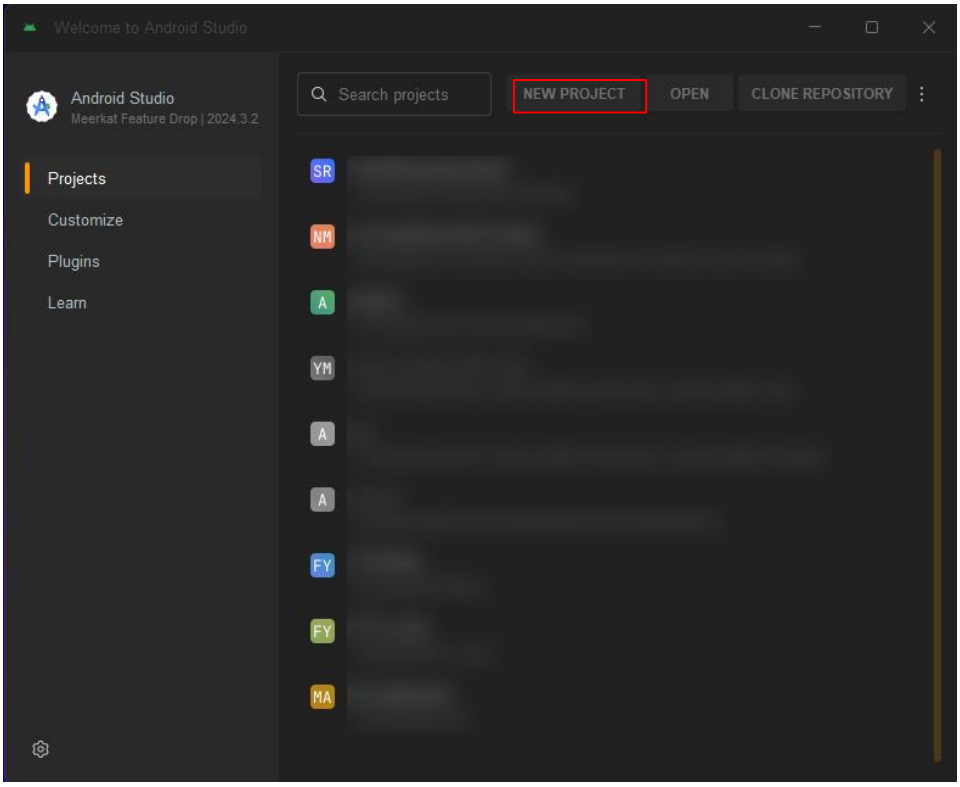


Figure 4.4: Main page of Android Studio

After the installation of Android Studio, a new project can be created by clicking the button of "New Project" as shown in Figure 4.4.

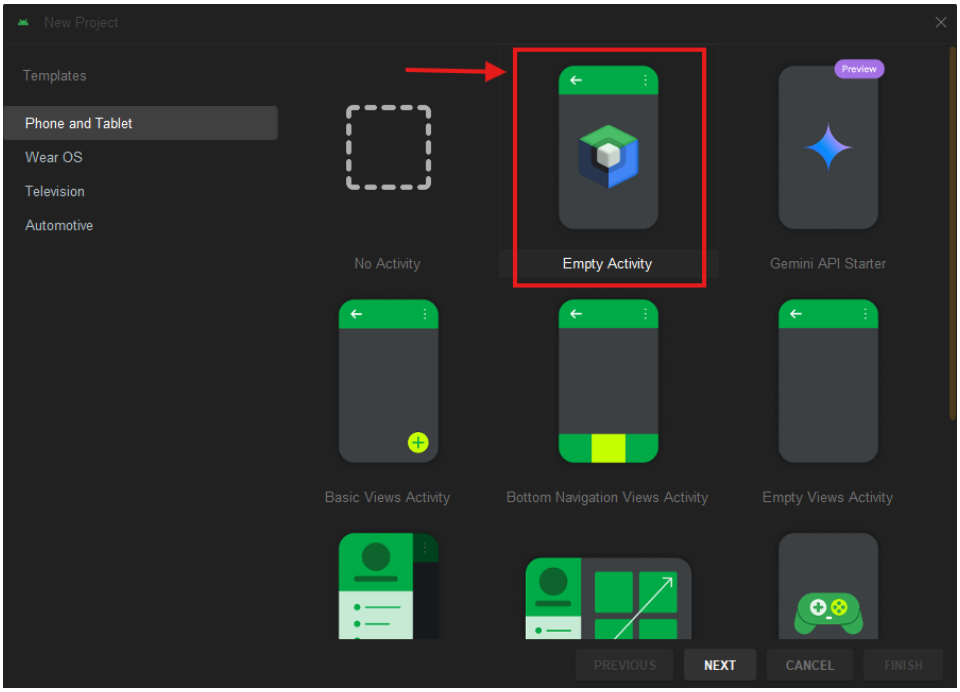


Figure 4.5: Project Template

Next, in Figure 4.6 shows the Android project configuration begins with the project name, which represents the application's identity, and the package name, which uniquely identifies the app in the Play Store and on the device. One would then choose the programming language (could be Java or Kotlin) depending on the needs of the development. Furthermore, the minimum SDK version is set so as to inform about the lowest Android OS version the app will officially be supporting. After all configurations are done, a click on the Finish button will initialize the project, generating the files and directories needed for development.

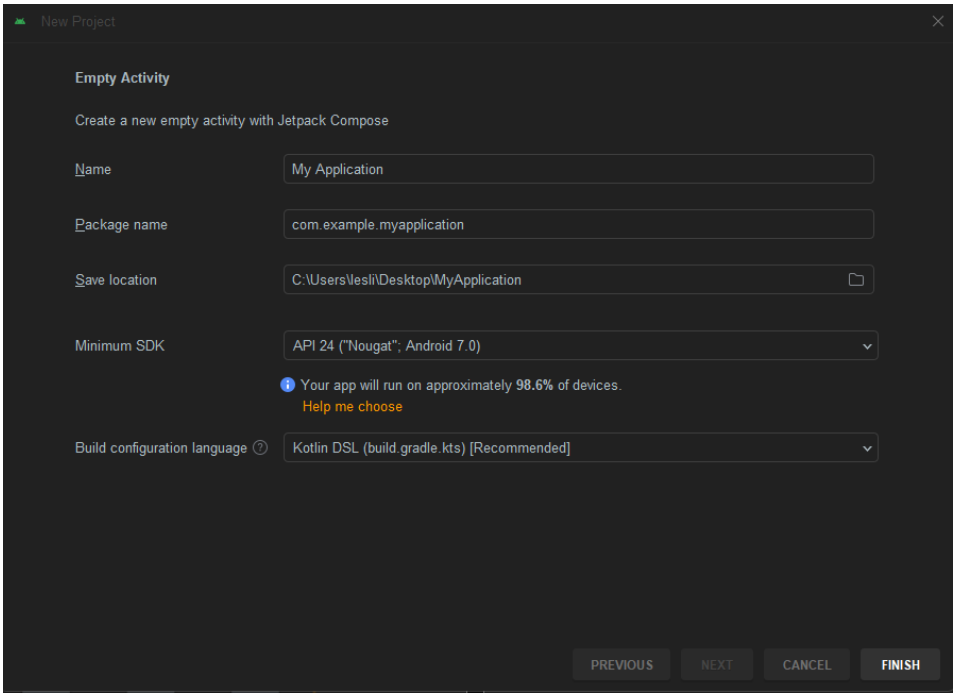


Figure 4.6: Android Project Configuration

Setup of Virtual and Physical Device

Android Studio enables developers to test and run their applications on virtual Android devices and physical mobile devices. Developers utilize a virtual device named an emulator to mimic Android hardware on a computer which allows them to assess their applications across multiple screen sizes and Android versions. To acquire accurate hardware performance results developers can test their applications with USB-connected phones or tablets. The two testing methods work together to assess application performance across different operating conditions.

At the right side of the page of the project, a button of device manager which is used to setup virtual and physical device can be clicked as shown in Figure 4.7. In Figure 4.8, users are allowed to choose whatever virtual device they want and yet can configure their virtual device as shown in Figure 4.9.

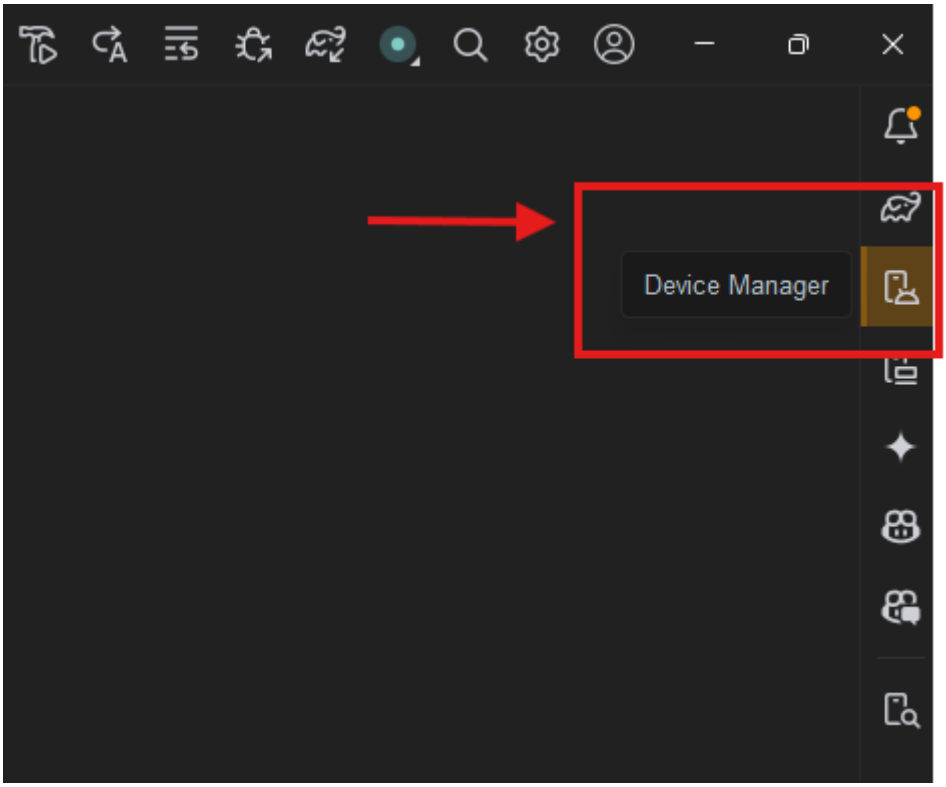


Figure 4.7: Device Manager

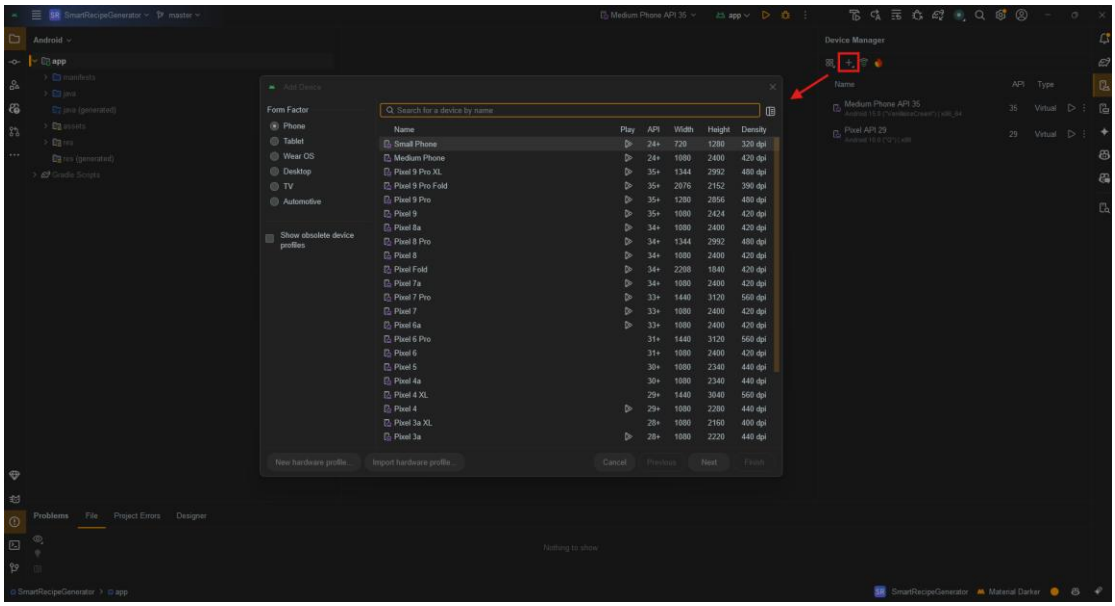


Figure 4.8: Add Virtual Device

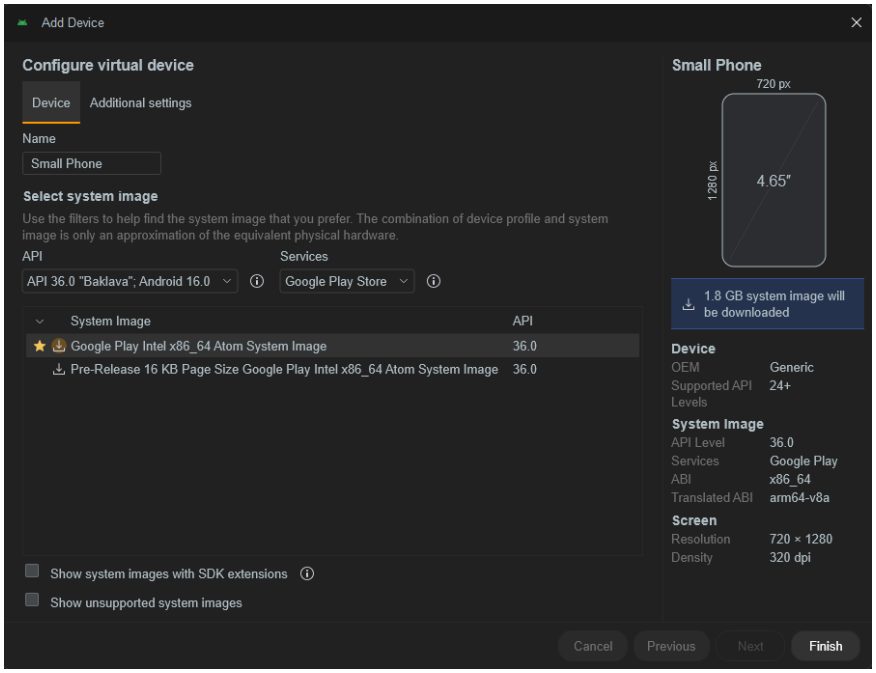


Figure 4.9: Configuration of Virtual Device

Besides, physical devices can be connected too in Android Studio to run and test their final application. But, before enabling the USB debugging, users have to keep on tapping the build number so that develop option is unlocked and enabled to access connection as shown in Figure 4.10 which is an example of demonstrating on Huawei Nova 5T with EMUI. Then, connect the physical device to the computer by using USB.

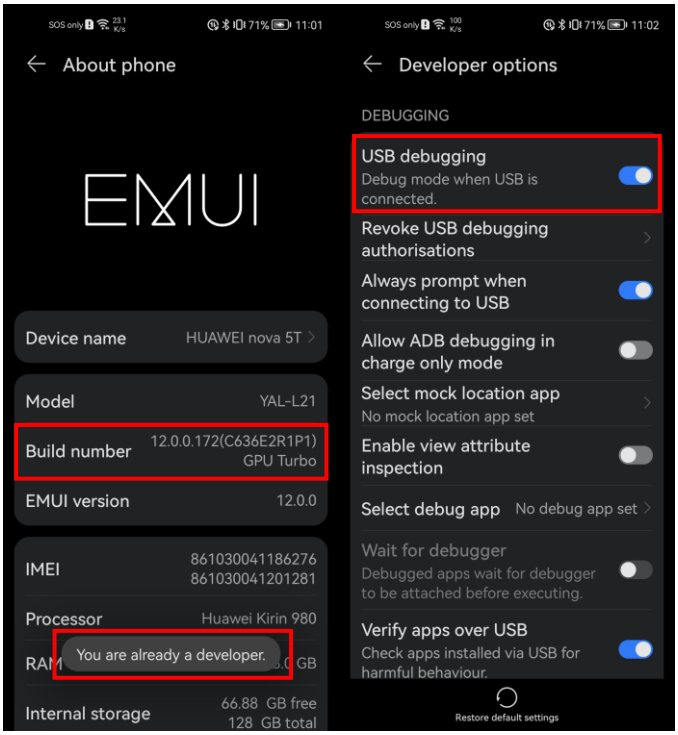


Figure 4.10: Enable USB Debugging

Figure below shows a list of devices that created virtually and physically in Device Manager.

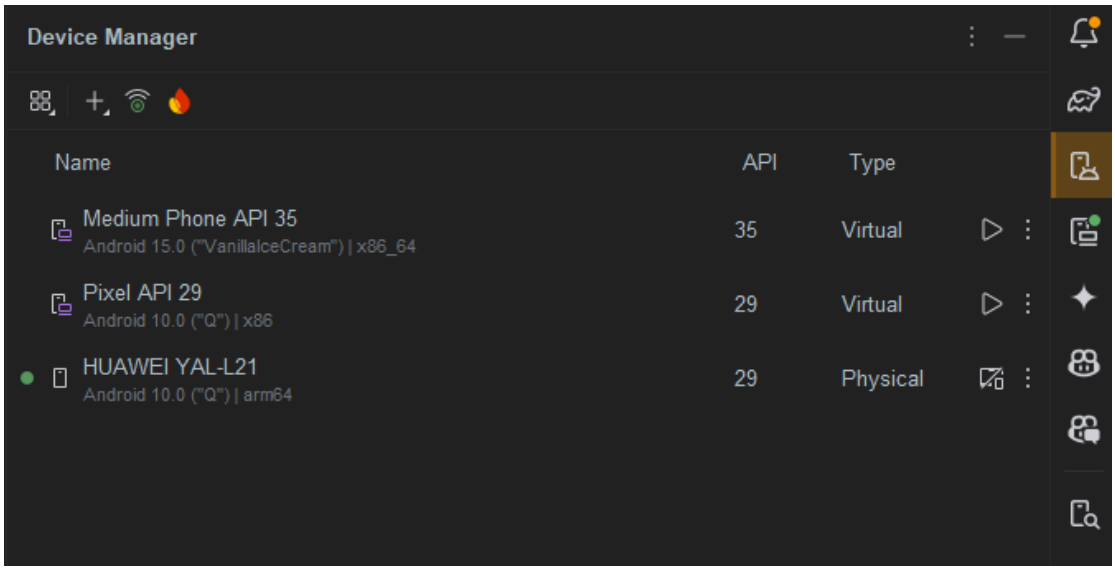


Figure 4.11: Android Studio Virtual and Physical Devices

Load Project from GitHub into Android Studio

Normally, users can load project by using two methods. First, is directly downloading the whole project as zip file, then extract it.

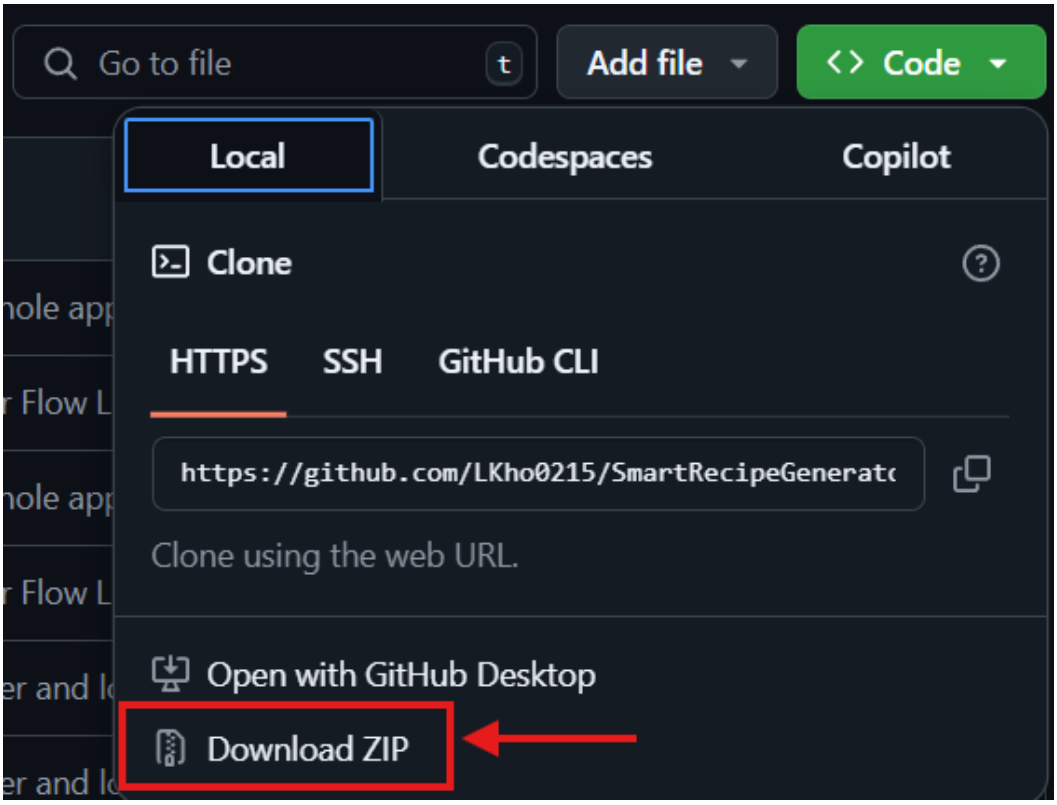


Figure 4.12: Download ZIP from GitHub

After extracting the zip file, the project can be opened or accessed in Android Studio as shown in Figure 4.13 in the various type of view but mostly either in Android view or Project view.

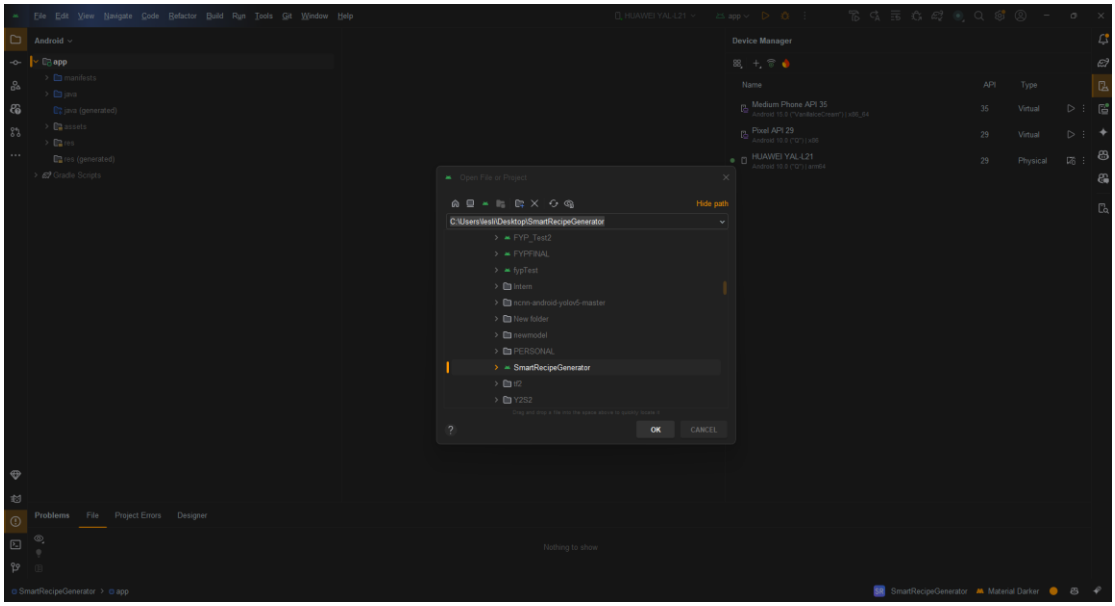


Figure 4.13: Open Project in Android Studio

Another way to load the project from GitHub into Android Studio is using git clone. Version control systems like Git help developers monitor source code alterations as well as allow team members to work together and maintain project versions. Developers frequently employ the git clone command to establish local duplicates of distant repositories. Platform support for services like GitHub enables developers to maintain project synchronization between their local copies and remote code repositories.

But, users must download and install latest git first before using git clone through https://gitforwindows.org/. Once the installer has started, follow the instructions as provided in the Git Setup wizard screen until the installation is complete. After that, open the windows command prompt (or Git Bash if you selected not to use the standard Git Windows Command Prompt during the Git installation). Lastly, type git version to verify Git was installed as shown in Figure 4.14.

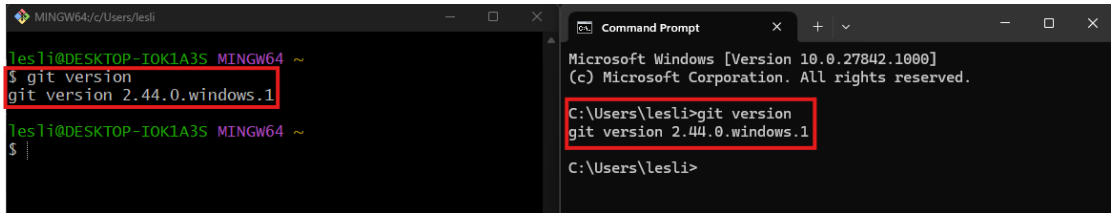


Figure 4.14: Check Version of Git in Git Bash/ CMD

Then, user can clone the project by key in the command “git clone [url]”. The url can be copied at the same panel which users download the ZIP file.

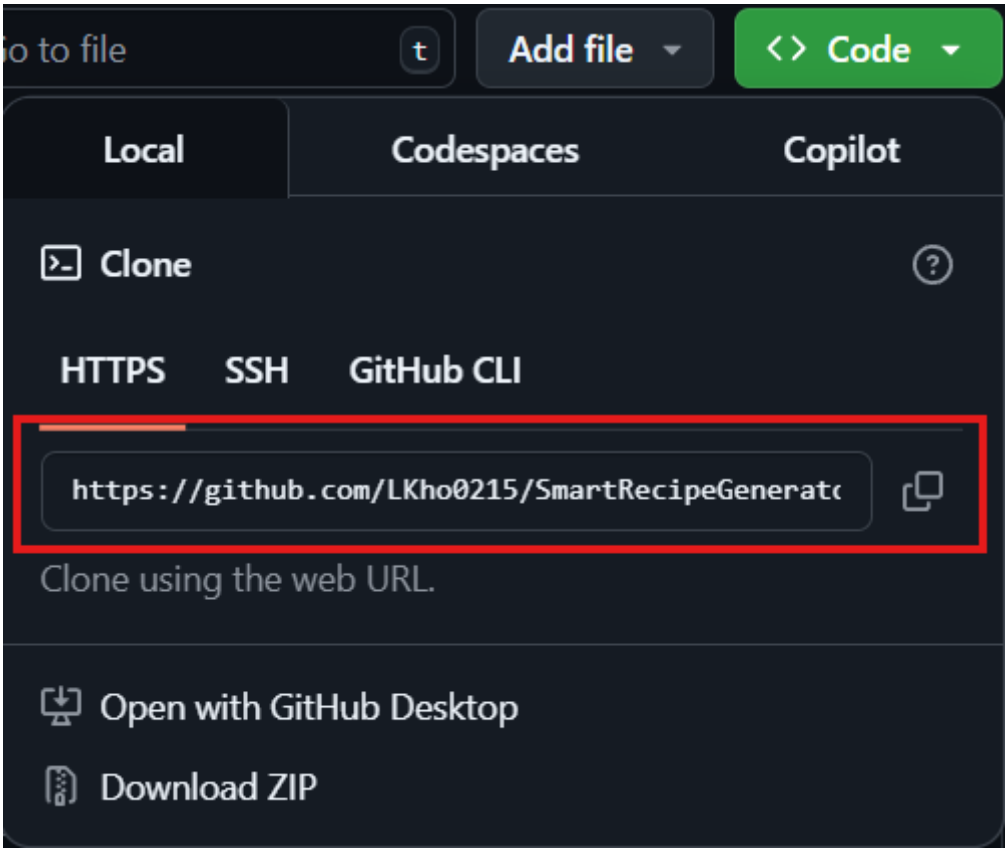


Figure 4.15: Copy URL of the Repo

Lastly, users can cd the directory wherever they want the repository to be installed as shown in Figure 4.16. So, Android Studio can open the project through the directory which the repository is cloned.

Figure 4.16: Git Clone the Repository

Build and Install APK (Android Package Kit)

The Android operating system utilizes APK or Android Package Kit files for application distribution and installation purposes. The APK file consists of complete application elements such as program code together with resources and assets and a manifest file. Developers can distribute and install their apps through the installation of APK files on Android devices.

After load the project from the GitHub to the Android Studio successfully, users can build the APK themselves and install in own phone. First, go to Build option on the top of the Android Studio. Inside Build, there is an option of "Generate App Bundles or APKs", then choose to generate APKs. After waiting for the Grandle to be built, a notification of generate APKs complete is shown at right corner of the Android Studio as shown in Figure 4.17.

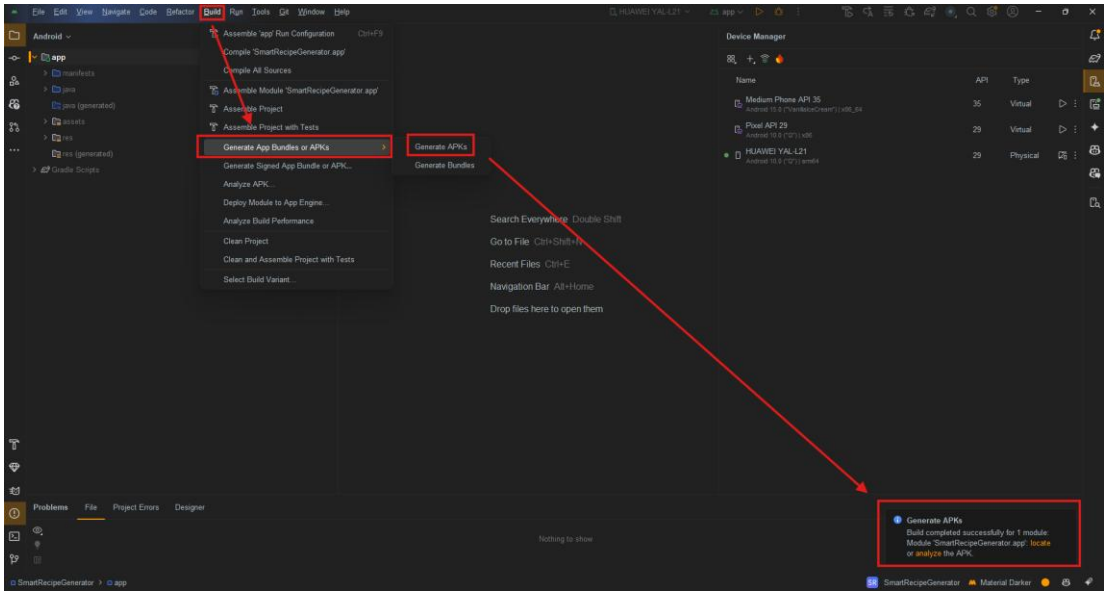


Figure 4.17: Build APKs

If the users click on the “locate” on the notification, users will be navigated to the directory of the APKs generated which is in app\build\outputs\apk\debug as shown in Figure below.

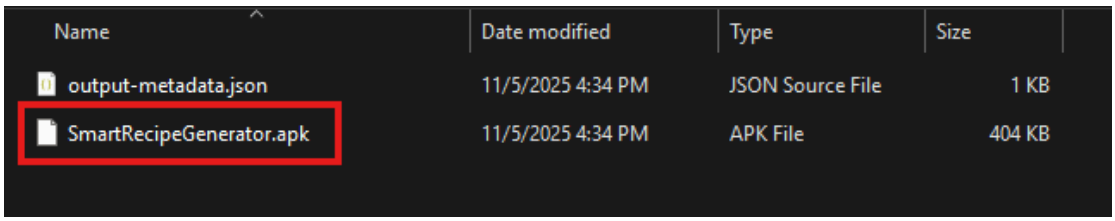


Figure 4.18: APKs Generated

Then, users can send the APK file to the Android device through various methods. For example, users can send the APK file through social media to their own account like Whatsapp and Telegram. Google Drive can be used too as a way to let the Android device access the APK file.

To install the APK, users just need to tap on the APK file where the users send and then tap on install as well as wait for the apps to be installed successfully in the Android device as shown in Figure 4.19.

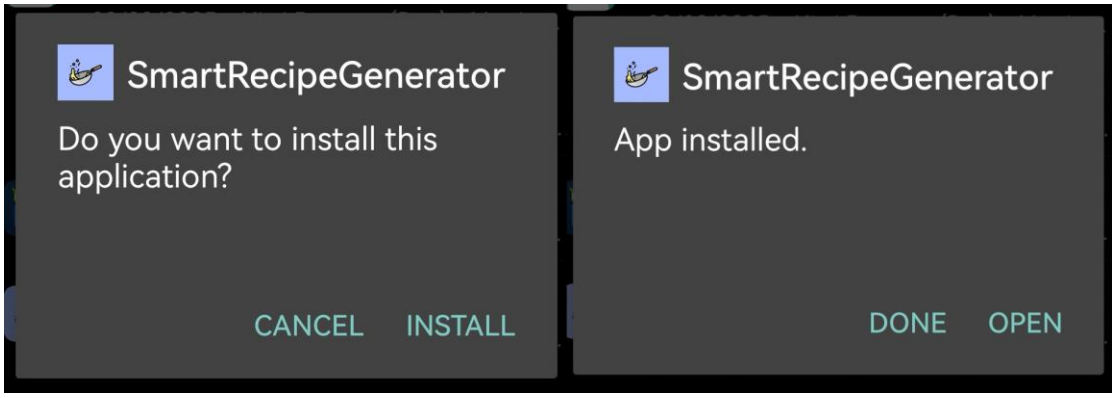


Figure 4.19: Install APK Successfully

The application functions on an at least Android 8.0 system which has an API level 26. The application works on systems that contain ARM (armeabi-v7a, arm64-v8a) and x86 (x86, x86_64) architectures. Users should choose a device with a minimum of 2 GB RAM and functional camera capabilities since the application runs CameraX and TensorFlow Lite for its image processing functions. Users need to have

sufficient storage space to accommodate both image files and PDF documents. The core functions of image capture and network operations alongside file management require camera accessibility as well as storage read/write permissions and internet access.

4.2.2 Artificial Intelligence (AI) Model Training

Machine learning operations refer to the method that enables machines to learn patterns and data-based decision making. The model gains knowledge through examination of significant datasets which leads to modifications of its core settings to enhance precision. Through this process AI systems develop their ability to execute operations like classification and prediction and recognition better as time progresses.

Prepare Dataset

In this project, a set of datasets which is named “FOOD-INGREDIENTS” is downloaded in Roboflow Universe for the future training. After entering the page of the dataset, various versions of dataset can be downloaded. Then, users can click on the “download dataset” option to download the dataset.

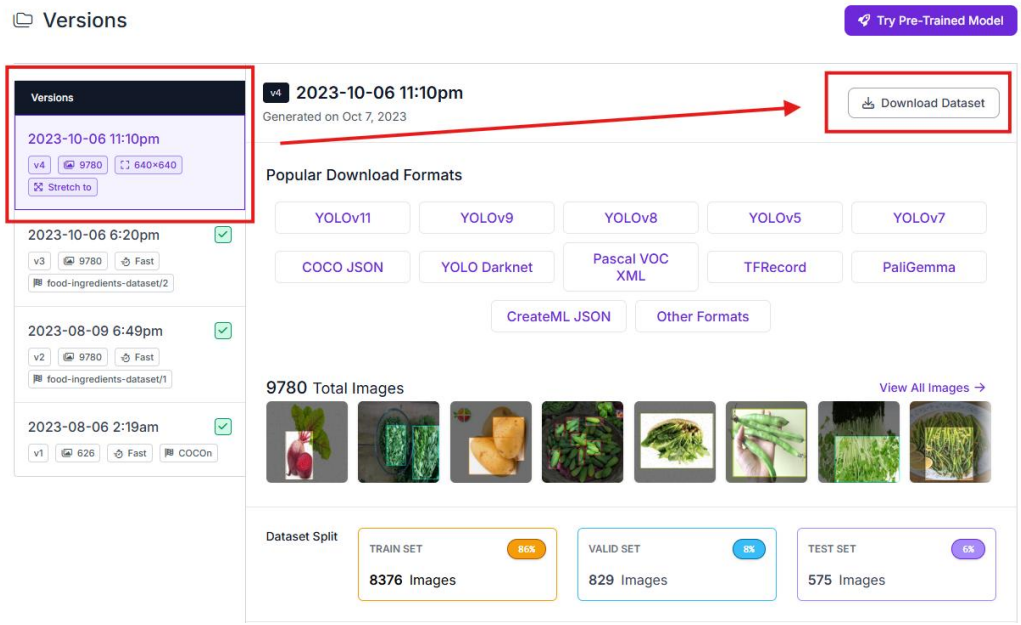


Figure 4.20: Version of Dataset

Users are allowed to select the format of the dataset which contains three files (train, valid and set). The model uses the train file to acquire patterns from the training data through its teaching process. During the training process the model optimizes its performance through the validation file to minimize overfitting. After training the test file assesses the model's capacity to produce accurate results with new data points in real-world situations. In this case, dataset in the format of YOLO v5 PyTorch is downloaded as shown figure below.

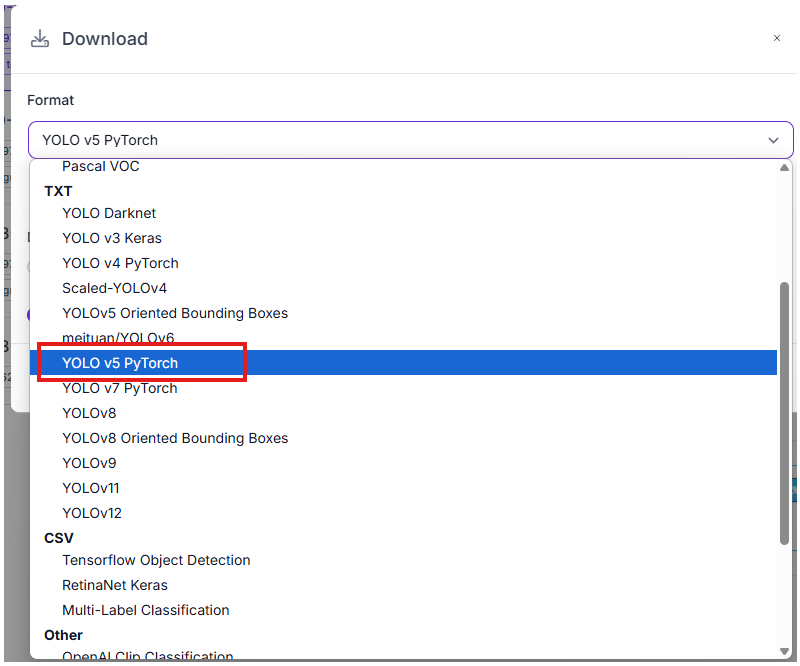


Figure 4.21: Download YOLO v5 PyTorch Format

Train Model

First, environment of model training must be setup before start training. Python is the major language for the training, hence, the latest version of python must be download through official website, https://www.python.org/downloads/. After installing the Python, users are required edit the environment variable. Users can search the keyword to lead to the window as shown in figure below.

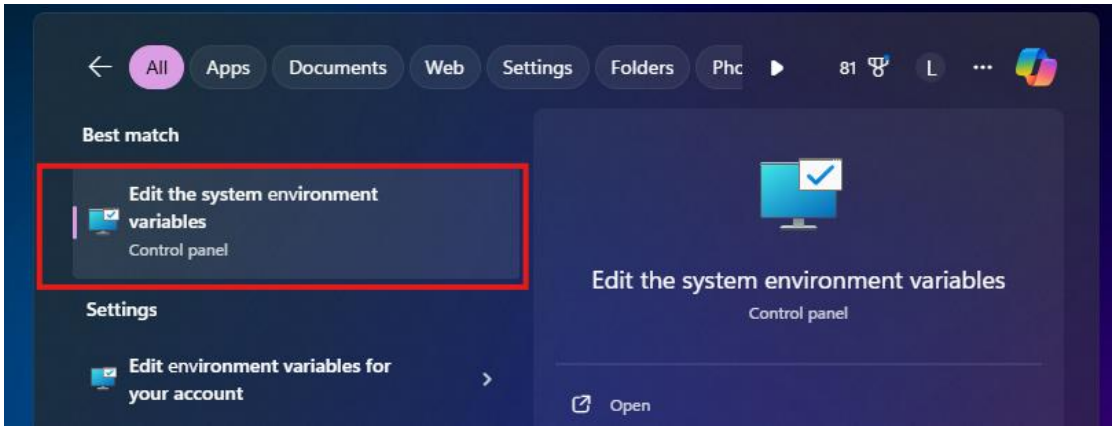


Figure 4.22: Search to Edit System Environment Variables

Then, users have to click on “Environment Variables” button then double click on “Path”. To add the python into system variable by clicking on “New” button and paste the path of python installed as well as clicking on “OK” button then all done. Figure 4.23 shows the complete steps on edit the system environment variables.

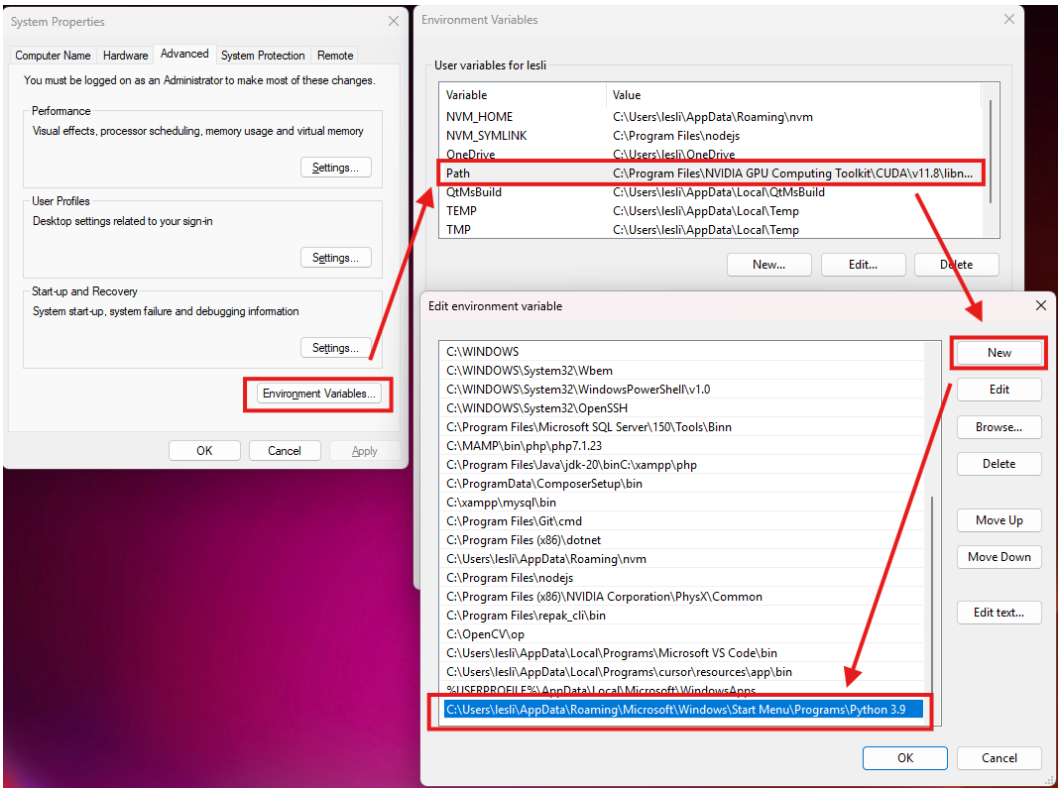


Figure 4.23: Steps to Edit the System Environment Variables

Next, miniconda is recommended to create the environment to train the model. It is because it offers users a lightweight and adaptable solution for maintaining independent Python workspaces alongside their dependencies. Users can avoid package

conflicts while enabling different projects to run various library versions simultaneously without creating any issues. Users find the system valuable when handling intricate system configurations and when running machine learning tools that need precise Python or library versions. Users can download the latest version of miniconda through the link of https://repo.anaconda.com/miniconda/.

After installing miniconda, users can access miniconda by searching keyword of “miniconda” in the search bar of window. The first time users, base environments can be observed from miniconda as shown in Figure 4.24. In miniconda, the term base refers to the default environment that is created automatically when Miniconda is installed.

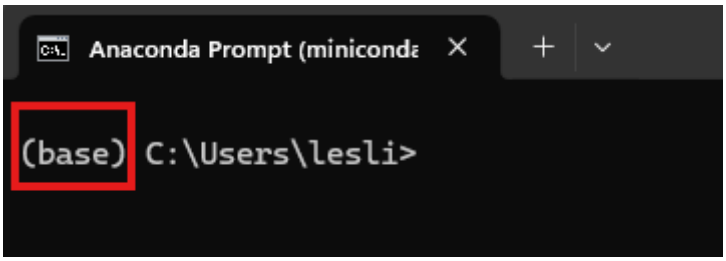


Figure 4.24: Base Environment

Then, users can start creating a new environment specifically for the model training by entering the command below:

```
conda create -n [env name] python=3.10 -y
```

Users can check their environment list by entering the command:

```
conda env list
```

to show all the enviroment created in a list as shown in Figure below:

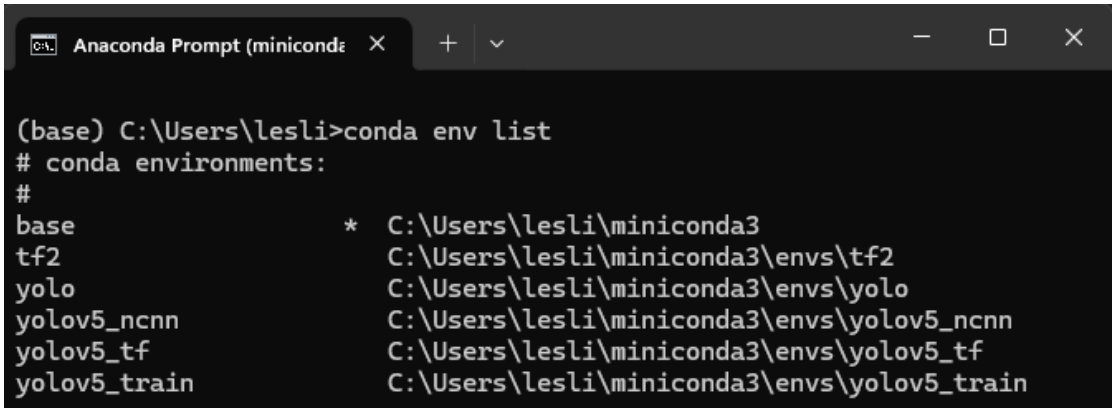


Figure 4.25: Environment List

User can active the evinronment by entering the command:

```
conda activate [env name]
```

After that, a new environment is applied as shown in figure below.

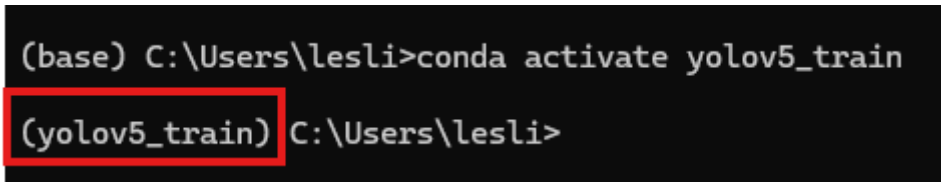


Figure 4.26: New Environment Applied

In this case, the computer consists of GPU, hence GPU is used to train model. First, users can check the GPU model in your own computer or laptop through device manager or command: `nvidia-smi`

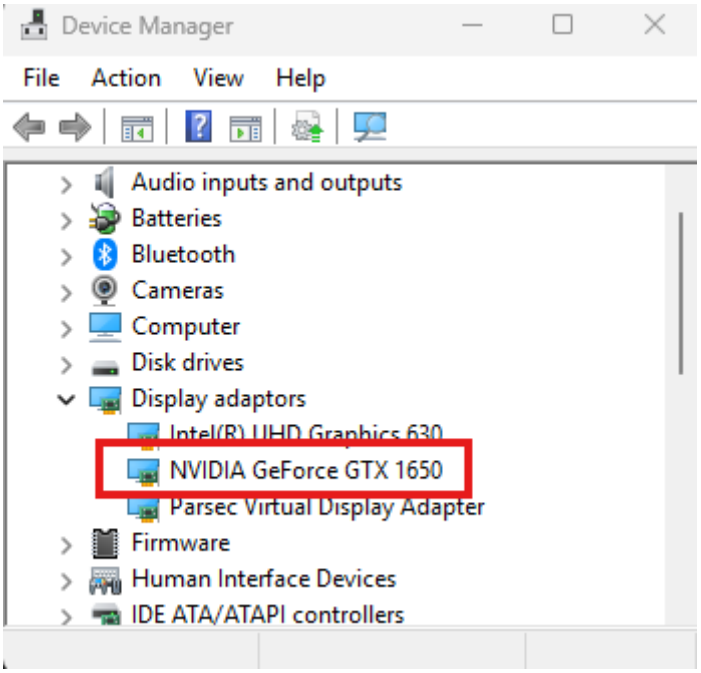


Figure 4.27: Check GPU inside Device through Device Manager

```
(yolov5_train) C:\Users\lesli>nvidia-smi
Mon May 12 16:47:10 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 555.97                 Driver Version: 555.97         CUDA Version: 12.5     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                     Driver-Model | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |                  MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA GeForce GTX 1650      WDDM  |   00000000:01:00.0 Off |                  N/A |
| N/A   47C    P0             11W /   50W |       0MiB /   4096MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```

Figure 4.28: Check GPU inside Device through Command Line

Then, check CUDA compute capability of the GPU in https://developer.nvidia.com/cuda-gpus as shown in table below.

8.0	NVIDIA A100 NVIDIA A30	
7.5	NVIDIA T4 T1000 T600 T400 T2000 T1200 T500	RTX 8000 RTX 6000 RTX 5000 RTX 4000 RTX 3000 GeForce GTX 1650 Ti NVIDIA TITAN RTX GeForce RTX 2080 Ti GeForce RTX 2080 GeForce RTX 2070 GeForce RTX 2060

Figure 4.29: Check CUDA compute capability

Based on the mapping table below to choose the correct CUDA version:

Compute Capability	Max CUDA Version Supported
8.6	CUDA 11.8+
8.0	CUDA 11.x
7.5	CUDA 11.x
7.0	CUDA 11.x
6.1	CUDA 11.x
5.0–6.1	Up to CUDA 10.2–11.1

Table 4.1: Mapping Table of Compute Capability and Max CUDA Version Supported

With the GPU of GeForce GTX 1650, CUDA version of 11.8 will be used. The command to install the specified version of CUDA, torch, torchvision and torchaudio can be copied in https://pytorch.org/get-started/locally/ as show below.

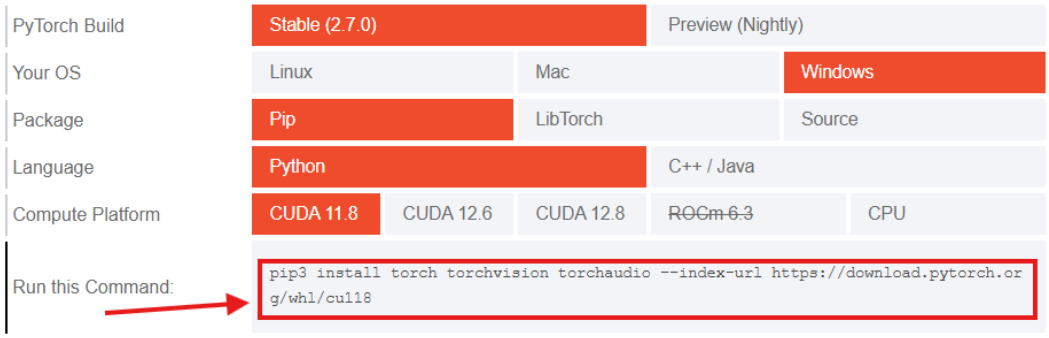


Figure 4.30: Command to Install CUDA, torch, torchvision and torchaudio

After completing the installation of Pytorch, use git clone to clone theYOLOv5 repository and install dependencies. The command should be key in in order as show below:

```
git clone https://github.com/ultralytics/yolov5.git

cd yolov5

pip install -r requirements.txt
```

Next, it is ready to train the model, before start make sure the three file which are train, valid and test as well as the data.yaml. After confirming the exisiting of those files, command as shown below is key in to start the training:

```
python train.py \
  --img 640 \
  --batch 16 \
  --epochs 100 \
  --data /absolute/path/to/dataset/data.yaml \
  --weights yolov5s.pt \
  --name custom_yolov5s
```

The “train.py” is already existed when git clone the YOLOv5 repository. The --img 640 option sets the input image resolution to 640×640 pixels, which is commonly used and balances speed with accuracy. The --batch 16 parameter defines the number of images processed simultaneously during training, with a batch size of 16 being a moderate choice that fits on most GPUs. The --epochs 100 option instructs the model to train for 100 complete passes over the dataset, allowing the model to learn progressively. The --data flag specifies the path to the data.yaml file, which contains information about the dataset structure, including the training and validation paths, number of classes, and class names. The --weights yolov5s.pt parameter loads the pretrained YOLOv5-small model weights to start training from an already-learned base, which usually speeds up convergence and improves results. Finally, the --name custom_yolov5s option sets the name of the training run folder, making it easier to organize and track results, which will be saved in the runs/train/custom_yolov5s/

directory. Finally, the output of best.pt file which is stand for pytorch model file, normally the output of best.pt means the model weights that achieved the lowest validation loss (best performance) during training.

Convert Model

The recommendation for Android deployment points to TFLite because it provides specialized optimization for mobile and embedded devices instead of PyTorch models. Mobile CPUs and GPUs can execute TFLite models faster while maintaining smaller file sizes and reduced latency compared to standard PyTorch models. TFLite models benefit from NNAPI and GPU delegate hardware acceleration while using less memory and providing better integration with Android applications through Android Studio built-in support thus making them superior for on-device inference in comparison to standard PyTorch model deployment.

Inside the YOLOv5 repository already consists of export.py to export pytorch model in any form. The only users have do is changing some of the parameter inside the existing code and run it so that the previous trained model can be converted into tflite model which can be implemented in Android device. The figure below has shown what is the main parameters that have to be changed to convert the trained pytorch model to tflite model.

```
parser = argparse.ArgumentParser()
    parser.add_argument("--data", type=str, default=ROOT /
"data1.yaml", help="dataset.yaml path")
    parser.add_argument("--weights", nargs="+", type=str,
default=ROOT / "runs/train/exp/weights/best.pt", help="model.pt
path(s)")
```

```
    parser.add_argument("--imgsz", "--img", "--img-size",
nargs="+", type=int, default=[640, 640], help="image (h, w)")
    parser.add_argument("--batch-size", type=int, default=1,
help="batch size")
    parser.add_argument("--device", default="0", help="cuda device,
i.e. 0 or 0,1,2,3 or cpu")
    parser.add_argument("--half", action="store_true", help="FP16
half-precision export")
    parser.add_argument("--inplace", action="store_true", help="set
YOLOv5 Detect() inplace=True")
    parser.add_argument("--keras", action="store_true", help="TF:
use Keras")
    parser.add_argument("--optimize", action="store_true",
help="TorchScript: optimize for mobile")
    parser.add_argument("--int8",
default="true",action="store_true", help="CoreML/TF/OpenVINO INT8
quantization")
    parser.add_argument("--per-tensor", action="store_true",
help="TF per-tensor quantization")
    parser.add_argument("--dynamic", action="store_true",
help="ONNX/TF/TensorRT: dynamic axes")
    parser.add_argument("--cache", type=str, default="",
help="TensorRT: timing cache file path")
    parser.add_argument("--simplify", action="store_true",
help="ONNX: simplify model")
    parser.add_argument("--mlmodel", action="store_true",
help="CoreML: Export in *.mlmodel format")
    parser.add_argument("--opset", type=int, default=17,
help="ONNX: opset version")
```

```
    parser.add_argument("--verbose", action="store_true",
help="TensorRT: verbose log")
    parser.add_argument("--workspace", type=int, default=4,
help="TensorRT: workspace size (GB)")
    parser.add_argument("--nms", action="store_true", help="TF: add
NMS to model")
    parser.add_argument("--agnostic-nms", action="store_true",
help="TF: add agnostic NMS to model")
    parser.add_argument("--topk-per-class", type=int, default=100,
help="TF.js NMS: topk per class to keep")
    parser.add_argument("--topk-all", type=int, default=100,
help="TF.js NMS: topk for all classes to keep")
    parser.add_argument("--iou-thres", type=float, default=0.45,
help="TF.js NMS: IoU threshold")
    parser.add_argument("--conf-thres", type=float, default=0.25,
help="TF.js NMS: confidence threshold")
    parser.add_argument(
        "--include",
        nargs="+",
        default=["tflite"]
        help="torchscript, onnx, openvino, engine, coreml,
saved_model, pb, tflite, edgetpu, tfjs, paddle",
    )
    opt = parser.parse_known_args()[0] if known else
parser.parse_args()
    print_args(vars(opt))
    return opt
```

Above which is highlighted is the part of the parameters that's need to be change. The --data change to own YAML file that that defines the dataset structure including

class names and paths to training/validation data. The --weigths should be changed to the path of the trained YOLOv5 model, ensuring the correct weights are used for inference or conversion. The --imgsz changed to [640, 640] as it should match the input size used during training for optimal accuracy and performance. Then, the --device changed to “0” specifies which GPU to use during execution. Next, set the --int8 default to true so that the converted tflite model will be in int8 format because int8 significantly reduces model size and improves inference speed, while using less memory and power. Lastly, change the default into the format of tflite.

After running the command below on the terminal,

```
python export.py
```

the tflite will be converted successfully. Next, the properties model can be checked in “Netron” to make sure the input and output is correct as shown in Figure 4.31. Netron functions as a robust open-source visualization tool which professionals use to evaluate deep learning and machine learning and neural network models. The tool accommodates various format types which include PyTorch (.pt) and TensorFlow Lite (.tflite) and ONNX (.onnx) and Keras and CoreML among others. The clear graphical interface of Netron enables users to examine model architecture components such as layers and shapes and operation connections in order to simplify model debugging and analysis and model validation during deployment.

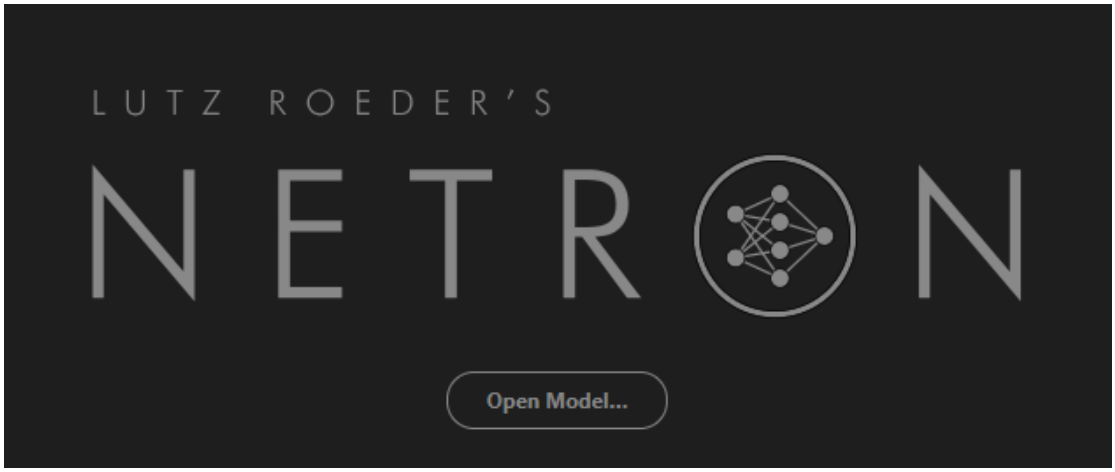


Figure 4.31: Netron

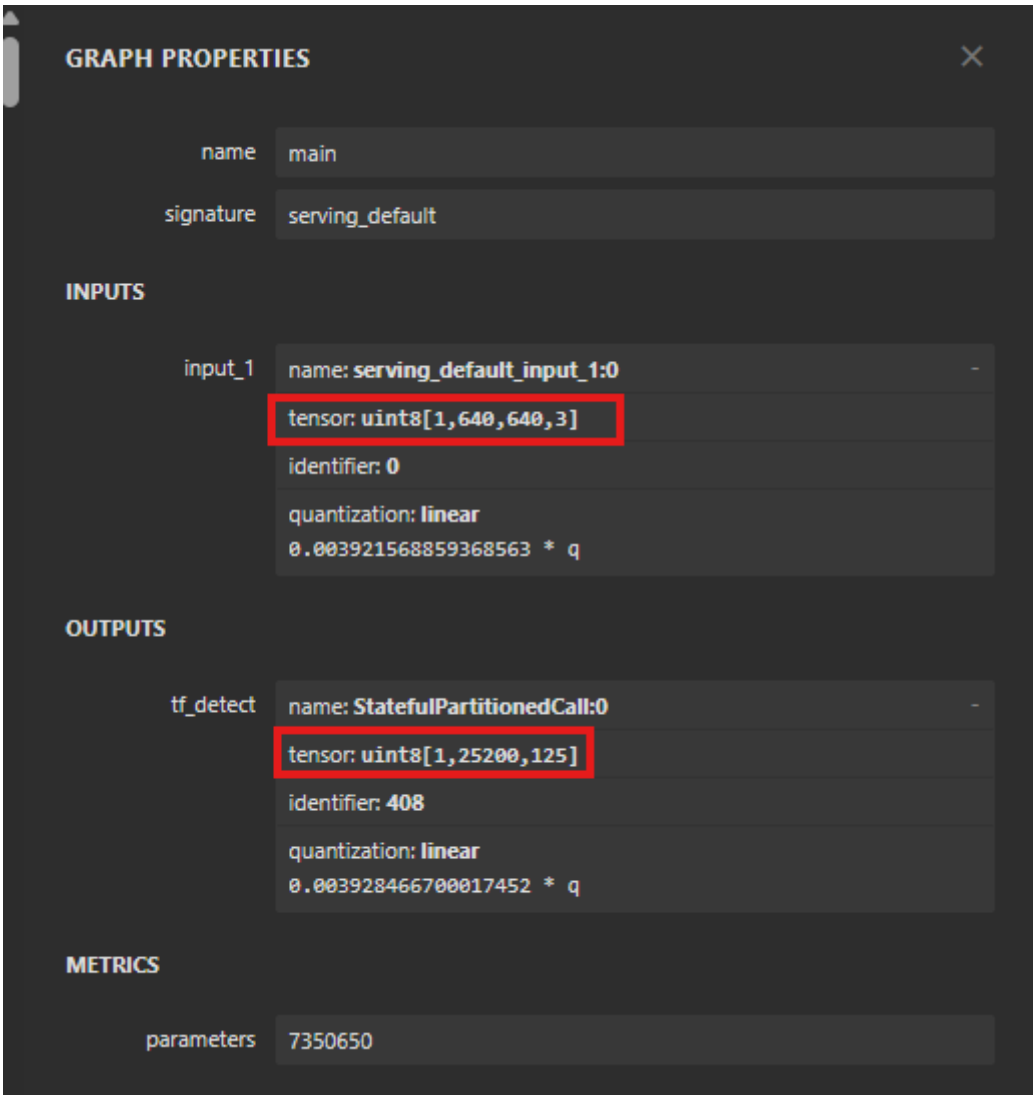


Figure 4.32: Graph Properties of Tflite Model

The input tensor uses the dimensions uint8[1, 640, 640, 3] to depict a 640×640 resolution image with 3 RGB color channels while using an 8-bit unsigned integer data

type to optimize mobile device performance. The output tensor uses the shape uint8[1, 25200, 125] to present the detection outcomes produced by the model. The number 25200 in this context represents the sum of bounding boxes which the model forecasts through various scales and features while the number 125 indicates the content associated with every prediction that typically includes bounding box coordinates together with objectness scores and class probabilities. Quantized YOLOv5 models utilize this specific data organization because it enables efficient processing on edge devices.

4.2.3 Food Ingredients Detection

The implementation starts by fetching the YOLOv5 model (`food-ingredients-int8-640.tflite`) and its corresponding labels (`labels.txt`) from the app's asset folder. To provide precise inference results the algorithm preprocesses the input image through three steps: cropping it to 640×640 square dimensions and enhancing contrast and color before resizing it to the model's required input format. The TFLite interpreter accepts the processed image as a byte buffer to achieve predictive results through its inference mechanism. The final result of the TFLite interpreter involves both raw detection outputs and intermediate processing of the model's predictions. The model outputs need to go through parsing to obtain bounding boxes together with class scores and confidence information. The implementation uses Non-Maximum Suppression (NMS) to handle multiple detections of the same object by removing duplicate results.

Before implementation of the model, dependencies about tensorflow lite must be implemented in the gradle as shown below.

```
    implementation("org.tensorflow:tensorflow-lite:2.13.0")
    implementation("org.tensorflow:tensorflow-lite-support:0.4.4")
    implementation("org.tensorflow:tensorflow-lite-metadata:0.4.4")
```

Once the users take photo from the food ingredients, the original image will be cropped to square by using bitmap.

```
Bitmap squareBitmap = createSquareBitmap(bitmap);
```

The createSquareBitmap() function will get the width and height of the original image, then calculate the crop area from the center of the image and lastly crop into square bitmap.

```
private Bitmap createSquareBitmap(Bitmap originalBitmap) {
        int width = originalBitmap.getWidth();
        int height = originalBitmap.getHeight();
        int size = Math.min(width, height);

        int x = (width - size) / 2;
        int y = (height - size) / 2;

        return Bitmap.createBitmap(originalBitmap, x, y, size,
size);
    }
```

Then, cropped image will be enhanced to help the model detect objects more reliably, especially in food images.

```
Bitmap enhancedBitmap = enhanceImage(squareBitmap);
```

The enhanceImage() function will enhance the cropped image by increasing the contrast and slightly boost color saturation.

```
private Bitmap enhanceImage(Bitmap input) {
        Bitmap output = Bitmap.createBitmap(input.getWidth(),
input.getHeight(), input.getConfig());

        Canvas canvas = new Canvas(output);
        Paint paint = new Paint();

        ColorMatrix colorMatrix = new ColorMatrix();

        colorMatrix.set(new float[] {
            1.2f, 0, 0, 0, -20,
```

```
            0, 1.2f, 0, 0, -20,
            0, 0, 1.2f, 0, -20,
            0, 0, 0, 1, 0
        });

        paint.setColorFilter(new
ColorMatrixColorFilter(colorMatrix));

        canvas.drawBitmap(input, 0, 0, paint);

        return output;
    }
```

After that, the image will be resize into model input size that expected by the YOLOv5 model which is 640 x 640 pixels by using bitmap.

```
Bitmap resizedBitmap = Bitmap.createScaledBitmap(enhancedBitmap,
INPUT_SIZE, INPUT_SIZE, true);
```

Next, convert the bitmap into bytebuffer as the model expects a flat byte buffer containing the image's RGB pixel data, with possible enhancements for better detection. Besides, it also applies contrast and color enhancements if the image is low-contrast, which is common in food photography.

```
private void convertBitmapToByteBuffer(Bitmap bitmap) {
        inputBuffer.rewind();

        bitmap.getPixels(intValues, 0, bitmap.getWidth(), 0, 0,
bitmap.getWidth(), bitmap.getHeight());

        Log.d(TAG, "Converting bitmap to input buffer");

        long pixelSum = 0;
        long pixelSqSum = 0;
        int pixelCount = INPUT_SIZE * INPUT_SIZE;

        for (int i = 0; i < pixelCount; i++) {
            int pixel = intValues[i];
            int r = (pixel >> 16) & 0xFF;
            int g = (pixel >> 8) & 0xFF;
            int b = pixel & 0xFF;
            int gray = (r + g + b) / 3;

            pixelSum += gray;
            pixelSqSum += gray * gray;
        }
```

```
        float mean = (float) pixelSum / pixelCount;
        float variance = ((float) pixelSqSum / pixelCount) - (mean
* mean);
        float stdDev = (float) Math.sqrt(variance);

        Log.d(TAG, String.format("Image statistics: mean=%.2f,
stdDev=%.2f", mean, stdDev));

        float enhanceContrastThreshold = 50;
        float foodSaturationBoost = 1.2f;

        float contrastEnhanceFactor = (stdDev <
enhanceContrastThreshold) ?
            40 / Math.max(stdDev, 10) : 1.0f;

        for (int y = 0; y < INPUT_SIZE; y++) {
            for (int x = 0; x < INPUT_SIZE; x++) {
                int pixelValue = intValues[y * INPUT_SIZE + x];

                int r = (pixelValue >> 16) & 0xFF;
                int g = (pixelValue >> 8) & 0xFF;
                int b = pixelValue & 0xFF;

                if (stdDev < enhanceContrastThreshold) {
                    r = enhancePixel(r, mean,
contrastEnhanceFactor);
                    g = enhancePixel(g, mean,
contrastEnhanceFactor);
                    b = enhancePixel(b, mean,
contrastEnhanceFactor);
                }

                float[] hsv = new float[3];
                Color.RGBToHSV(r, g, b, hsv);
                hsv[1] = Math.min(1.0f, hsv[1] *
foodSaturationBoost);
                int enhancedColor = Color.HSVToColor(hsv);

                r = (enhancedColor >> 16) & 0xFF;
                g = (enhancedColor >> 8) & 0xFF;
                b = enhancedColor & 0xFF;

                inputBuffer.put((byte) r);
                inputBuffer.put((byte) g);
                inputBuffer.put((byte) b);
            }
        }

        Log.d(TAG, "Input buffer prepared with size: " +
inputBuffer.capacity() + " bytes");
    }
```

First, in convertBitmapToByteBuffer() function, it resets the input buffer to

ensure the buffer is ready to be filled from the start. If there is any other previous data, it will be overwritten with the code below:

```
inputBuffer.rewind();
```

Then, the pixel data will be extracted by using bitmap. Filling the intValues array with the ARGB pixel values from the bitmap.

```
bitmap.getPixels(intValues, 0, bitmap.getWidth(), 0, 0,
bitmap.getWidth(), bitmap.getHeight());
```

The image statistics is calculated by computing the mean and standard deviation of the grayscale values of the image to determine if the image is low contrast.

```
long pixelSum = 0;
        long pixelSqSum = 0;
        int pixelCount = INPUT_SIZE * INPUT_SIZE;

        for (int i = 0; i < pixelCount; i++) {
            int pixel = intValues[i];
            int r = (pixel >> 16) & 0xFF;
            int g = (pixel >> 8) & 0xFF;
            int b = pixel & 0xFF;
            int gray = (r + g + b) / 3;

            pixelSum += gray;
            pixelSqSum += gray * gray;
        }

        float mean = (float) pixelSum / pixelCount;
        float variance = ((float) pixelSqSum / pixelCount) - (mean
* mean);
        float stdDev = (float) Math.sqrt(variance);
```

Again, enhance the image by increasing contrast and boosting some colour saturation. If the image is low-contrast (stdDev < 50), set a high contrast enhancement factor.

```
float enhanceContrastThreshold = 50;
float foodSaturationBoost = 1.2f;

float contrastEnhanceFactor = (stdDev < enhanceContrastThreshold) ?
40 / Math.max(stdDev, 10) : 1.0f;
```

Next, processing each pixel and fill the buffer. The procedure which analyses each pixel present in the original picture before passing it for model inferences is prepared. The first step involves retrieving red, green and blue (RGB) values from the original pixel data for each pixel. A custom contrast adjustment function named enhancePixel() enhances all colour channels when the system detects insufficient image contrast through standard deviation analysis. To enhance visual clarity of food pictures the system enhances colour saturation by converting RGB values to HSV (Hue, Saturation, Value) format and then increases the saturation before converting the result back to RGB. The modified RGB values go through sequential input buffer writing in the exact order required by the YOLOv5 tflite model to enable correct inference.

```
for (int y = 0; y < INPUT_SIZE; y++) {
    for (int x = 0; x < INPUT_SIZE; x++) {
        int pixelValue = intValues[y * INPUT_SIZE + x];

        int r = (pixelValue >> 16) & 0xFF;
        int g = (pixelValue >> 8) & 0xFF;
        int b = pixelValue & 0xFF;

        if (stdDev < enhanceContrastThreshold) {
            r = enhancePixel(r, mean, contrastEnhanceFactor);
            g = enhancePixel(g, mean, contrastEnhanceFactor);
            b = enhancePixel(b, mean, contrastEnhanceFactor);
        }

        float[] hsv = new float[3];
        Color.RGBToHSV(r, g, b, hsv);
        hsv[1] = Math.min(1.0f, hsv[1] * foodSaturationBoost);
        int enhancedColor = Color.HSVToColor(hsv);

        r = (enhancedColor >> 16) & 0xFF;
        g = (enhancedColor >> 8) & 0xFF;
        b = enhancedColor & 0xFF;

        inputBuffer.put((byte) r);
        inputBuffer.put((byte) g);
        inputBuffer.put((byte) b);
    }
}
```

After the converting of bitmap into bytebuffer, running the model inference. It

passes the input buffer into the tflite interpreter. The model runs and writes its output which is raw detection data into the output buffer.

```
    private void runInference() {
        try {
            outputBuffer.rewind();

            Map<Integer, Object> outputs = new HashMap<>();
            outputs.put(0, outputBuffer);
            Object[] inputs = {inputBuffer};

            tflite.runForMultipleInputsOutputs(inputs, outputs);

            Log.d(TAG, "Model inference completed successfully");
        } catch (Exception e) {
            Log.e(TAG, "Error running model inference", e);
        }
    }
```

Then, the processResults() will read the raw output from the YOLOv5 tflite model and decodes each predicted bounding box, objectness score and class score. The following code below loops through all 25,200 bounding boxes which the YOLOv5 tflite model produces. The code extracts 125 bytes of encoded detection data from the output buffer for every single bounding box. The model produces quantized outputs in 8-bit unsigned integer format (uint8) which requires dequantization to convert each byte back to its initial floating-point state according to the quantization parameters OUTPUT_SCALEand OUTPUT_ZERO_POINT. The dequantization process remains vital for accurate prediction interpretation because it restores numerical precision which is necessary to implement further processing operations for filtering and scoring and bounding box visualization.

```
for (int i = 0; i < OUTPUT_BOX_COUNT; i++) {
    byte[] boxData = new byte[BOX_DIMENSION];
    outputBuffer.get(boxData);

    float[] dequantized = new float[BOX_DIMENSION];
    for (int j = 0; j < BOX_DIMENSION; j++) {
        dequantized[j] = OUTPUT_SCALE * ((int) (boxData[j] & 0xFF)
- OUTPUT_ZERO_POINT);
    }
```

Besides, it will undergo early filtering by objectness which means that it will skip boxes with low objectness.

```
float objectness = dequantized[4];
if (objectness < OBJECTNESS_THRESHOLD) continue;
```

Then, it will extract the box coordinates by getting the center coordinates (x, y), width (w), and height (h) of the box which scaled to the model input with 640 x 640 pixels. Given a bounding box, the method lists the top three scores from among the 120 possible classes that can be predicted by the model. It iterates over each class score, starting from index 5 of the dequantized values, since the first few values correspond to box coordinates and objectness score. For every score, it appears if it is higher than any of the current top 3 scores and it gets inserted in the proper place while pushing lower scores down. The whole procedure ensures that only the three highest scoring classes are maintained, which is good for determining the most promising object categories in the box and improves the accuracy of future confidence calculation and display of results.

```
int[] topClasses = new int[3];
float[] topScores = new float[3];

for (int c = 0; c < OUTPUT_CLASS_COUNT; c++) {
    float score = dequantized[5 + c];
    for (int j = 0; j < 3; j++) {
        if (score > topScores[j]) {
            for (int k = 2; k > j; k--) {
                topClasses[k] = topClasses[k-1];
                topScores[k] = topScores[k-1];
            }
            topClasses[j] = c;
            topScores[j] = score;
            break;
        }
    }
}
```

It will also skip boxes where the best class is invalid or its score is too low.

```
if (topClasses[0] >= labels.size() || topScores[0] < 0.5) {
    continue;
}
```

The objectness score is modified by checking how distinctly the top predicted class stands out from the second-best. Class clarity is calculated by finding the difference in class scores between the top two classes relative to the top score. A larger difference means that the top class is more confident and unambiguously classified. Of course, the second class is seldom ambiguous. In such cases, the objectness score gets a slight bump, proportional to this clarity. Final objectness is then clipped at 1.0 so as to remain within the desired range. This results in a more confident detection produced when the model knows clearly the class it is classifying.

```
float classClarity = (topScores[0] - topScores[1]) / Math.max(0.1f,
topScores[0]);
float adjustedObjectness = objectness * (1.0f + 0.2f *
classClarity);
adjustedObjectness = Math.min(1.0f, adjustedObjectness);
```

By combining the objectness and class score, the final confidence is calculated. If the box with low confidence, it will be skipped.

```
float confidence = adjustedObjectness * topScores[0];
confidence = Math.min(1.0f, confidence);
if (confidence < CONFIDENCE_THRESHOLD) {
    continue;
}
```

The code below creates a RectF object defining the bounding box coordinates using the top-left (x1, y1) and bottom-right (x2, y2) points. Then a new Detection is instantiated containing the label name (from the labels list, using the detected class ID), confidence score, bounding box (rect), and the class ID itself. Finally, this detection is added to a list of valid detected instances that could be processed or displayed later in the application.

```
RectF rect = new RectF(x1, y1, x2, y2);
detectionList.add(new Detection(
        labels.get(detectedClass),
        confidence,
        rect,
        detectedClass));
```

In TFLite, NMS which is Non-Maximum Suppression is used to filter out overlapped bounding boxes and keep only the most confident detections. When a detector like YOLOv5 detects objects, it often generates multiple boxes for one. The NMS helps to remove these redundant boxes by comparing their confidence scores and overlap (IoU), keeping just the best one. It results in improved detection and clean, non-duplicated outputs for display or post-processing.

In NMS, there is a key which is calculating the IoU (Intersection over Union). The IoU value is used to determine if two detected boxes are likely the same object. The calculateIoU() function calculates the Intersection over Union (IoU) of two bounding boxes. IoU is a normal object detection metric that measures the overlap of two bounding boxes. To do this, the function finds the area of each box, then finds the coordinates of where the boxes overlap by taking the maximum of the two boxes' top left coordinates and the minimum of the two boxes' bottom right coordinates. If the boxes do actually overlap, the function finds the area of this box as well (intersection area). The last step is to divide the intersection area by the area that is covered by both boxes (union) to find the final value.

```
private float calculateIoU(RectF box1, RectF box2) {
    float areaBox1 = (box1.right - box1.left) * (box1.bottom -
box1.top);
    float areaBox2 = (box2.right - box2.left) * (box2.bottom -
box2.top);

    float intersectLeft = Math.max(box1.left, box2.left);
    float intersectTop = Math.max(box1.top, box2.top);
    float intersectRight = Math.min(box1.right, box2.right);
    float intersectBottom = Math.min(box1.bottom, box2.bottom);
```

```
    float intersectWidth = Math.max(0, intersectRight -
intersectLeft);
    float intersectHeight = Math.max(0, intersectBottom -
intersectTop);
    float intersectArea = intersectWidth * intersectHeight;

    return intersectArea / (areaBox1 + areaBox2 - intersectArea);
}
```

4.3 Launching the Proposed Application

This section shows the overview and the flow of the proposed mobile application which is named “Smart Recipe Generator” through screenshots. All the page of the proposed mobile application is created through WebView which includes html, css and js.

4.3.1 Login

The figure below shows the first time users who will be lead to which is login page. Users are told to fill in the email and password registered to login successfully into the application. If the users do not have an account, they are allowed to click on the “Sign Up” button to go to the Sign Up page to register page.

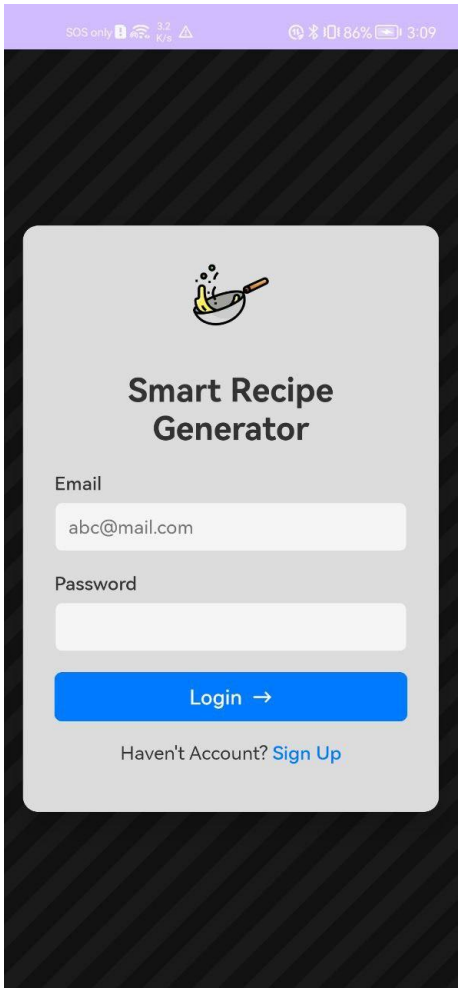


Figure 4.33: Login Page

There is validation on the login page which is the email and password must be filled else users are not allowed to login as show in Figure 4.34.

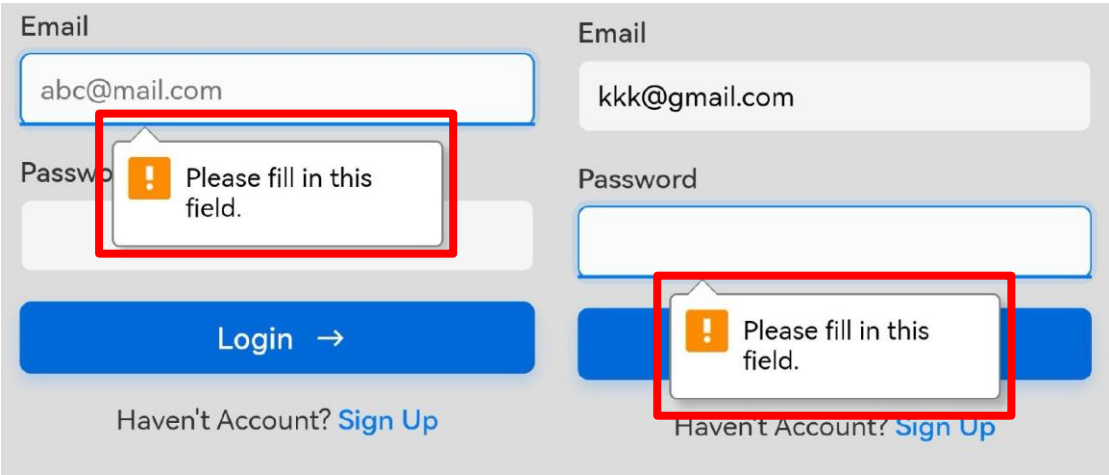


Figure 4.34: Validation on Login Page

Once the users login successfully, a loading page with notification “Login Successful” will be shown, else, users will be stay in login page and will be notified “Invalid Credentials” as shown below.

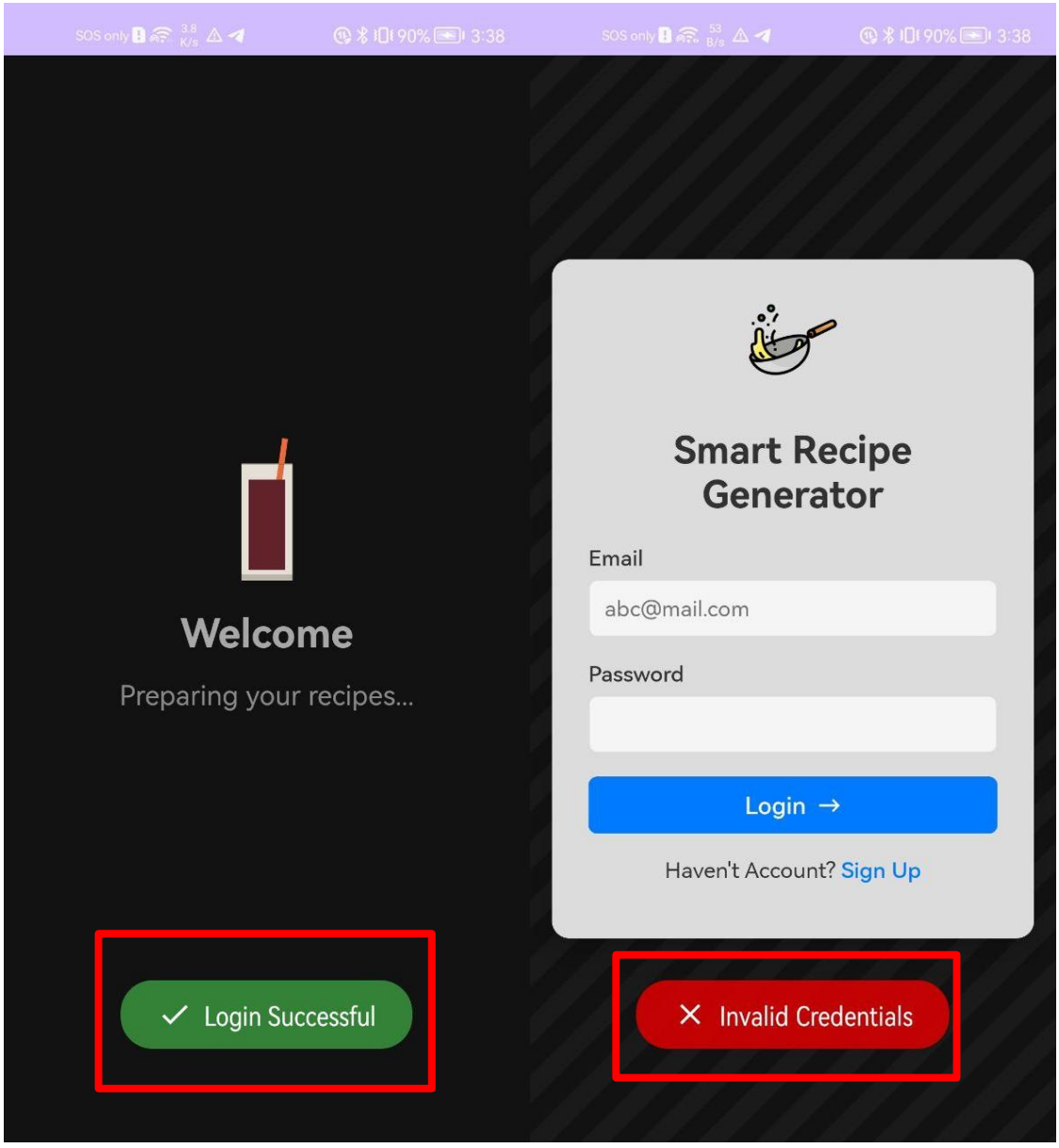


Figure 4.35: Success and Fail in Login Page

4.3.2 Sign Up

In Sign Up page as shown in Figure 4.36, users can register a new account for the proposed mobile application. Users are required to enter email address and password to register as a new user.

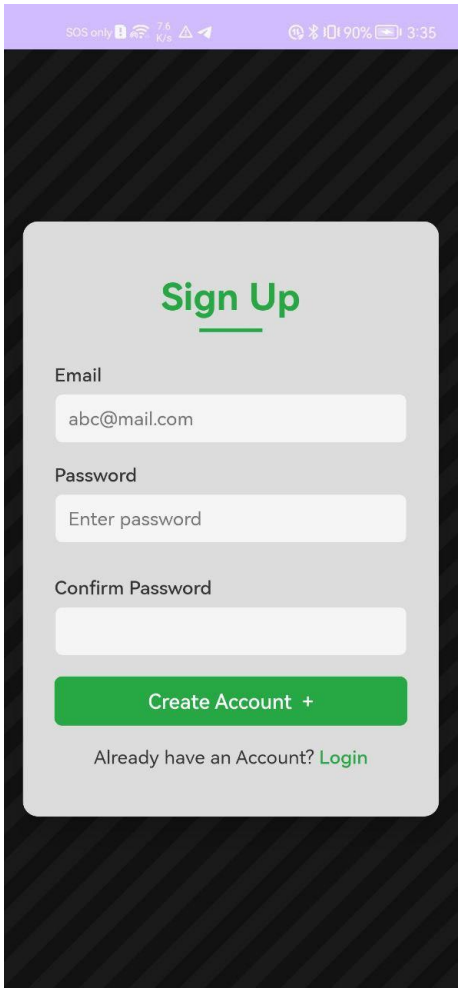


Figure 4.36: Sign Up Page

In Sign Up, users are required to enter the correct format of email else will be notified to enter correct format of email. Next, the password must be consist of at least 6 characters else users are not allowed to resgiter acccount. Besides, there is secure level of the users' password within range of weak, medium and strong. If the email has been used, a notification will be shown too and will not register successfully.

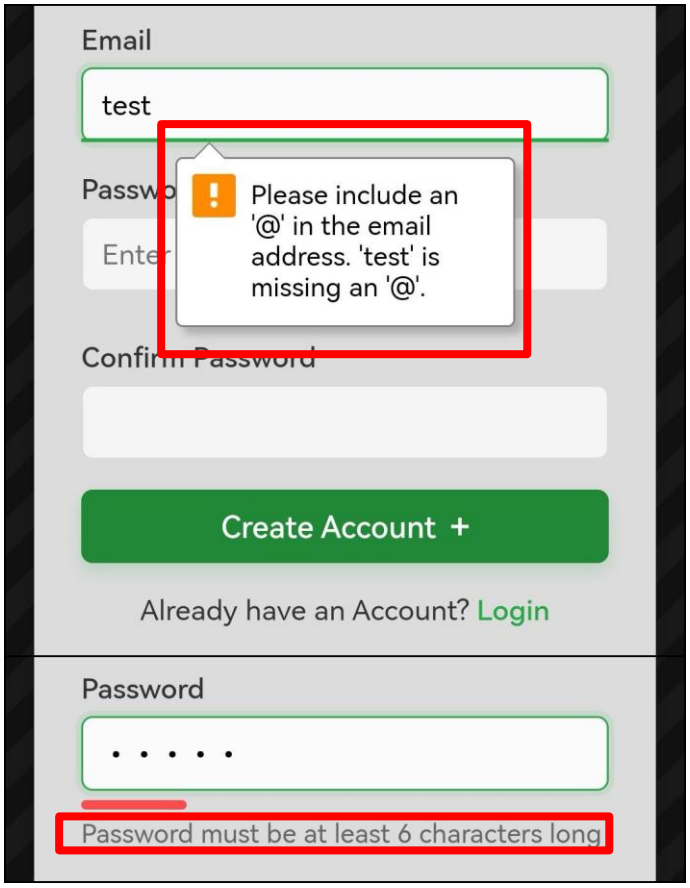


Figure 4.37: Validation of Correct Email Format and Password Length

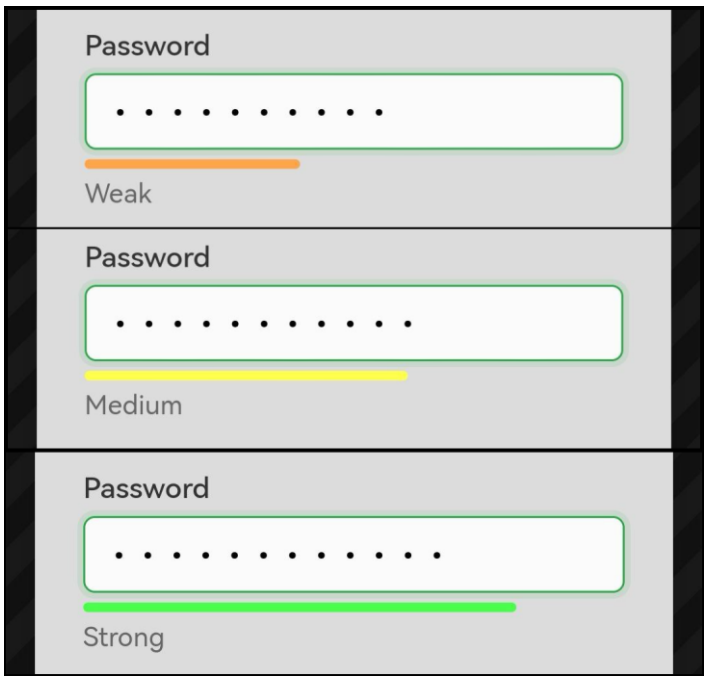


Figure 4.38: Secure Level of Password

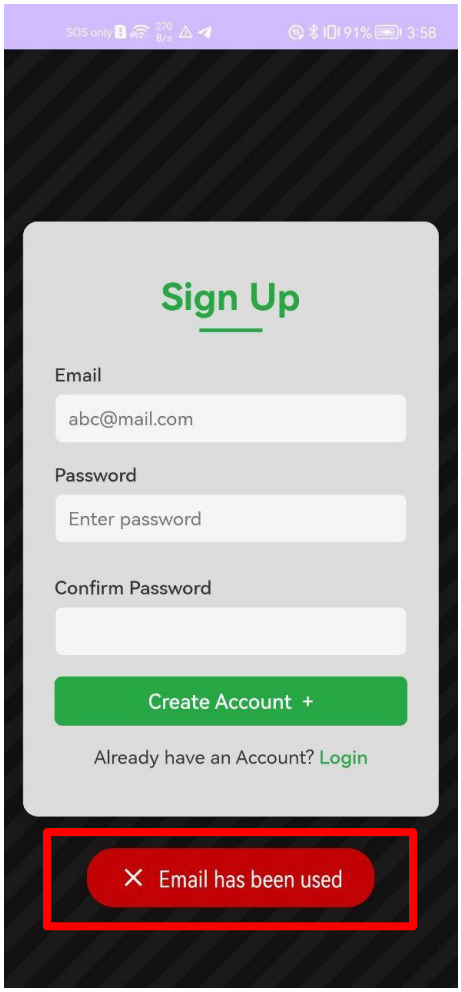


Figure 4.39: Registration with Used Email

Once the users register successfully, a notification will be shown and lead the users to the login page as shown below.

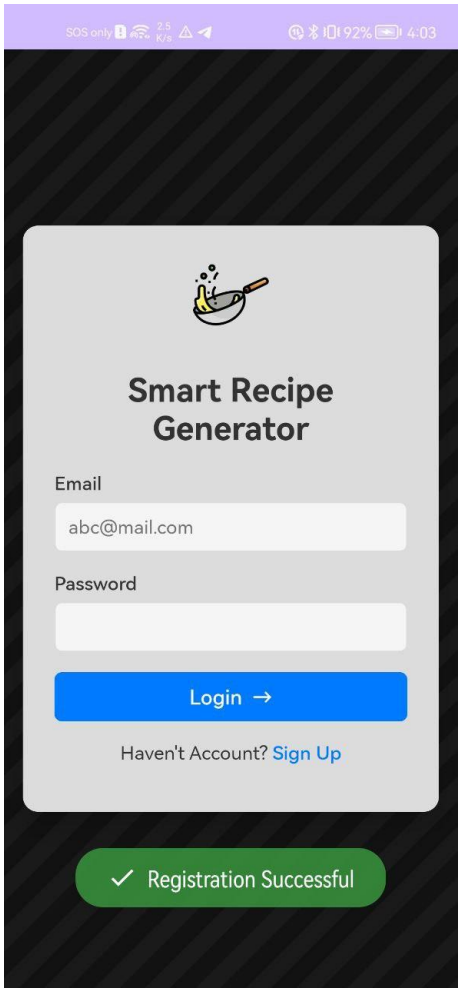


Figure 4.40: Registration Successful

4.3.3 Home Page

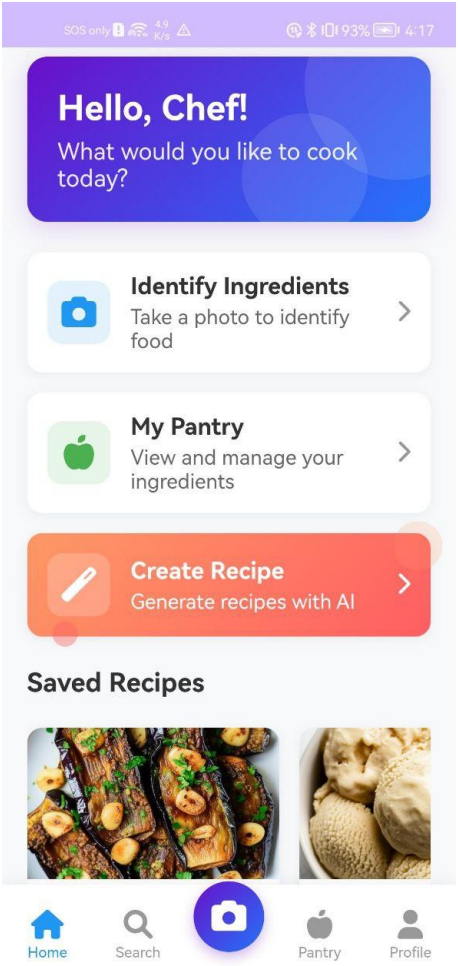


Figure 4.41: Home Page

In home page, there are severals sections can be observed. First, the “Identify Ingredients” section with a camera icon is for the users to take photo to identify food ingredients. Next, the “My Pantry” section with an apple icon is for the user to view and manage the identified and added food ingredients. Moreover, the “Create Recipe” with a magic stick icon is to generate recipes with AI. There is a sections called “Saved Recipe” is to show the recipes that saved by the users. There is also a bottom navigation that allows users to other pages which will be introduced in further part.

4.3.4 User Profile

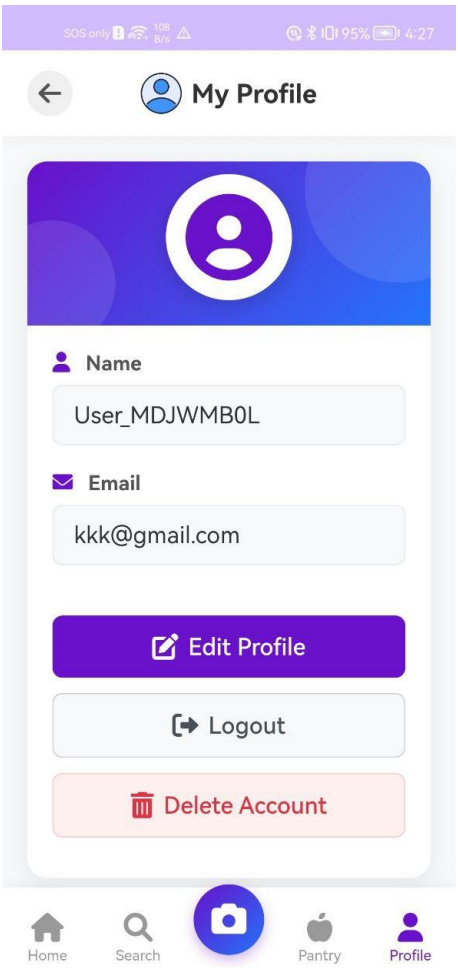


Figure 4.42: User Profile

In user profile, information of name and email address will be shown. In addiotion, users are allowed to edit profile, logout and delete current account. Since is a new user, username will be generated randomly. In “Edit Profile”, users are allowed to edit name, email address and password.

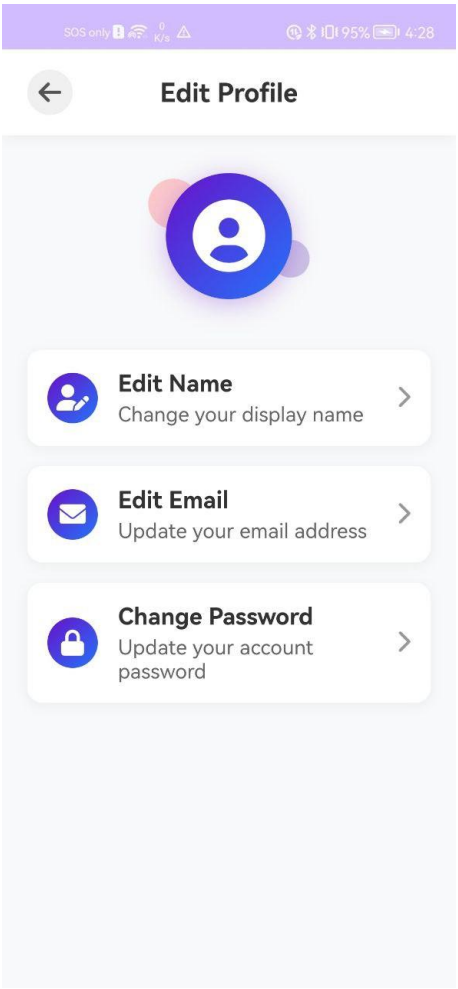


Figure 4.43: Edit Profile

4.3.5 Identify Ingredients

Users are allowed to take a photo to recognize or identify the food ingredients by tapping on the section of "Identify Ingredients" or the camera icon below the bottom navigation. Picture below is the example photo taken.

Figure 4.44: Example Photo

Once the users use camera to take photo, a list of possible items will be listed out. Users are given checkbox to select the true items to be stored in the pantry as shown in figure below. When added the items successfully, a notification of number of items are added to the pantry.

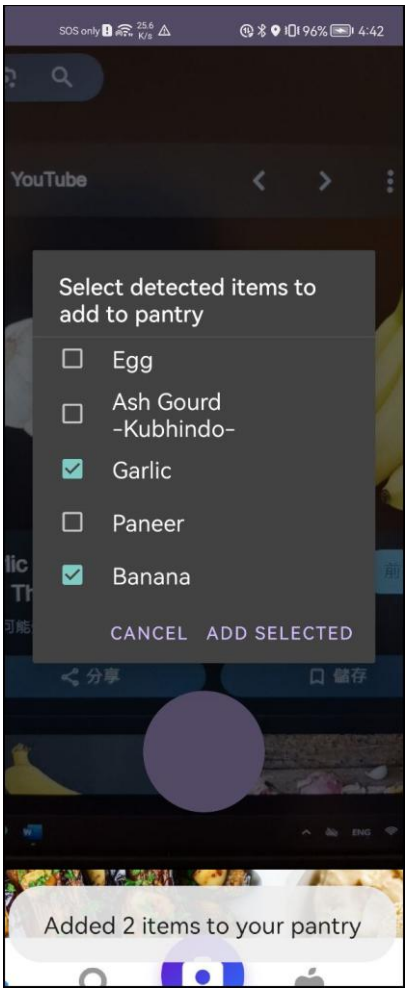


Figure 4.45: Identification of Food Ingredients

4.3.6 Pantry

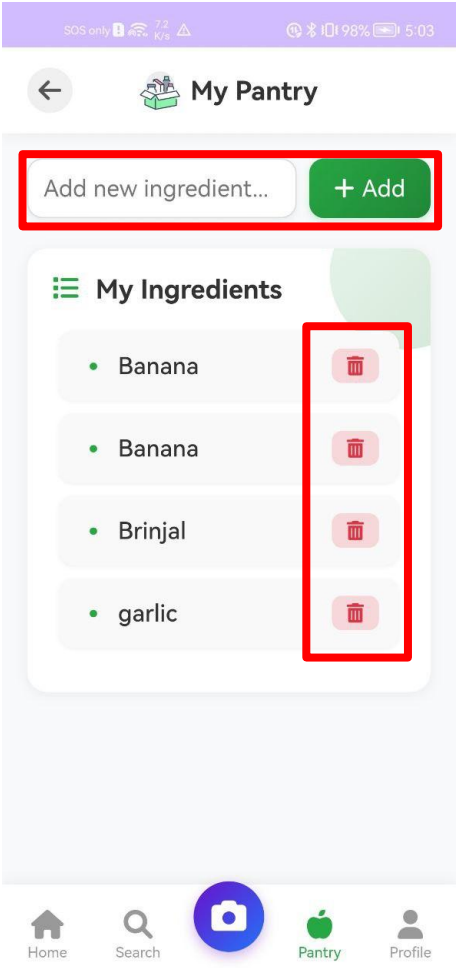


Figure 4.46: Panrty

In pantry, users can view and manage the food ingredients that identified in "Identify Ingredients" section. Due to the limitations of the tflite model trained, currently only support to identify up to 120 types of food ingredients, but, users are allowed to add the food ingredients manually into the pantry too if unsuccessful to recognize items. Besides, users are allowed to delete the ingredients too by tapping on the dustbin icon.

4.3.7 Create Recipe

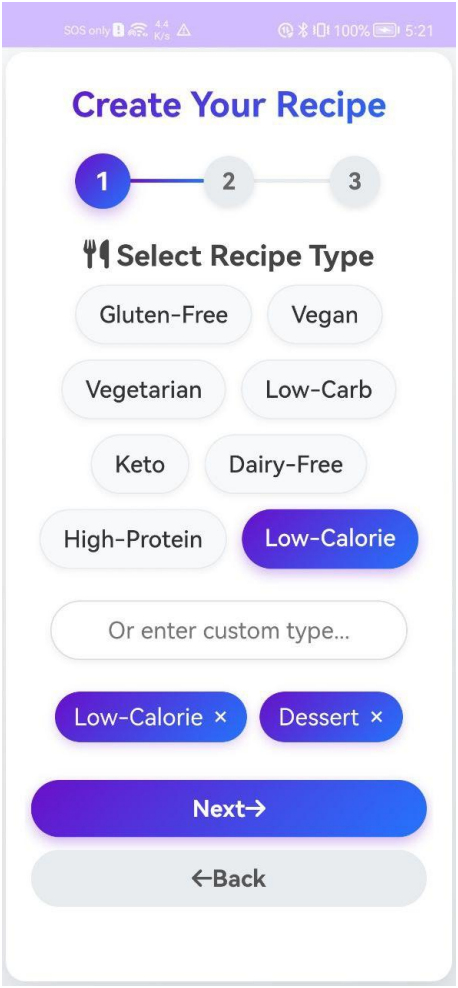


Figure 4.47: Select Recipe Type

In the section in “Create Recipe”, first, users are required to choose the type of recipe. There are some default choices built in for users to choose. Users can enter custom type too if default options are not satisfied.

Next, users need to select the ingredients that are involved in the recipe which have been stored in the pantry as shown in Figure 4.48.

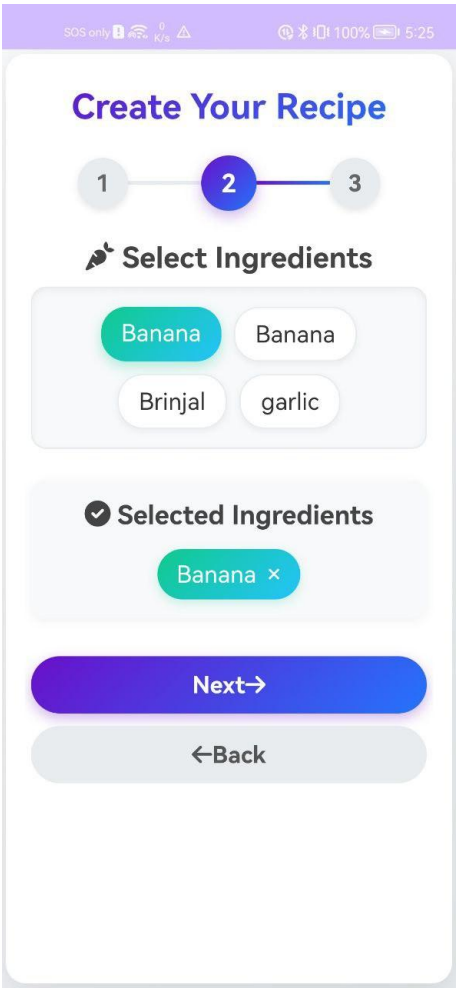


Figure 4.48: Select Ingredients

In the last page of the section, the summary of recipe type and selected ingredients are shown. Once confirmed, users just need to tap on generate recipe button as shown below.

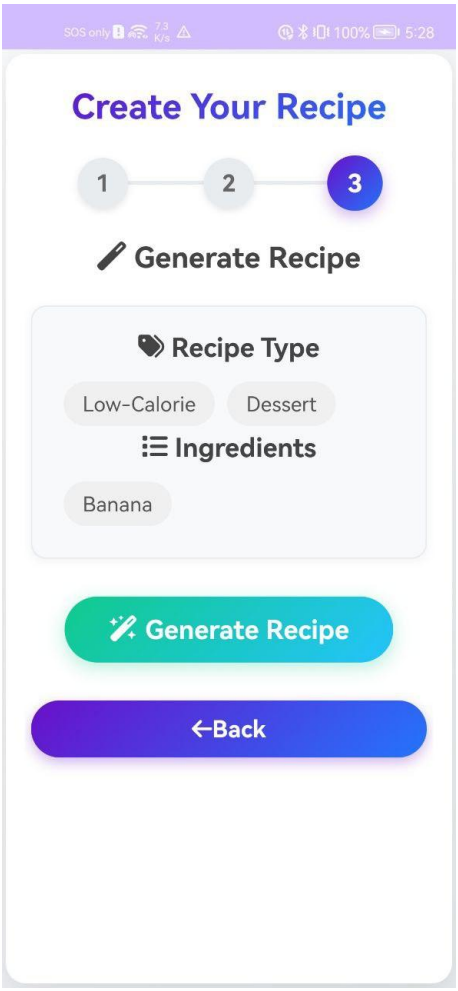


Figure 4.49: Summary of Recipe

4.3.8 Generated Recipe

Figure 4.50: Generated Recipe

The users are required to wait for a few seconds for the recipe picture to be generated. Inside the recipe, it consists of ingredients, instructions, cooking time and nutritional information.

Users are allowed to save the generated recipe by tapping the save recipe button at the bottom of the recipe as shown as below. After saving the recipe, users will be lead to home page again. The saved recipe will be shown at the “Saved Recipe” section in home page.

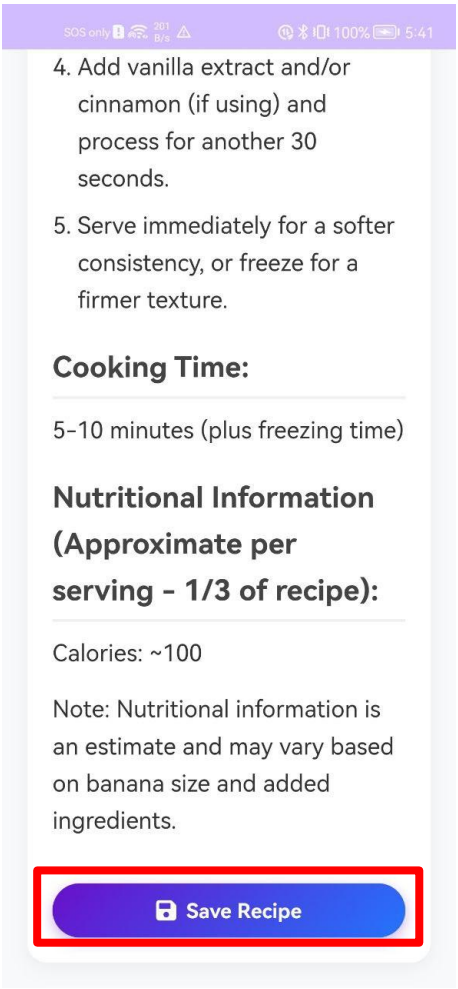


Figure 4.51: Save Recipe

4.3.9 Saved Recipe

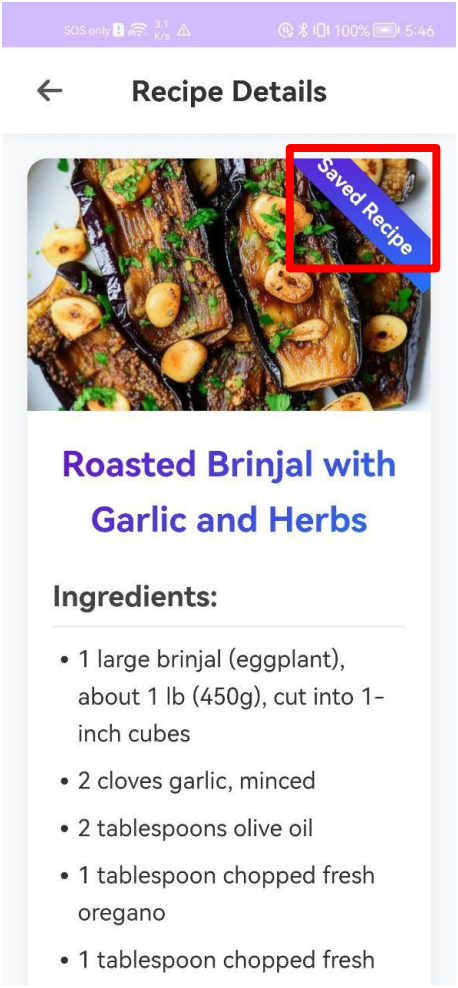


Figure 4.52: Saved Recipe

In the section of "Saved Recipe" in home page, users can browse the saved recipe. The saved recipe will be labelled as shown in Figure 4.52.

Saved Recipe is allowed to be shared through any social media by generating pdf and sent. In this case, the recipe is shared througn the email as shown as below. Furthermore, users can delete the saved recipe too.

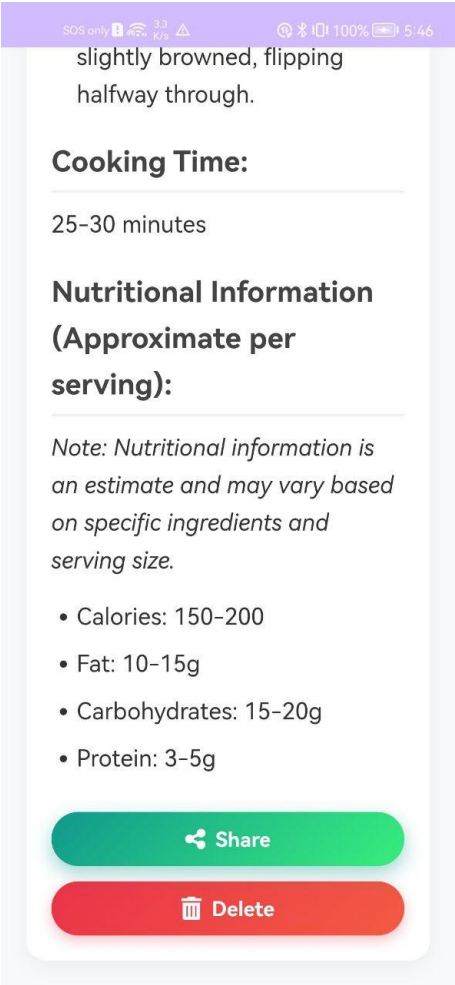


Figure 4.53: Share and Delete Saved Recipe

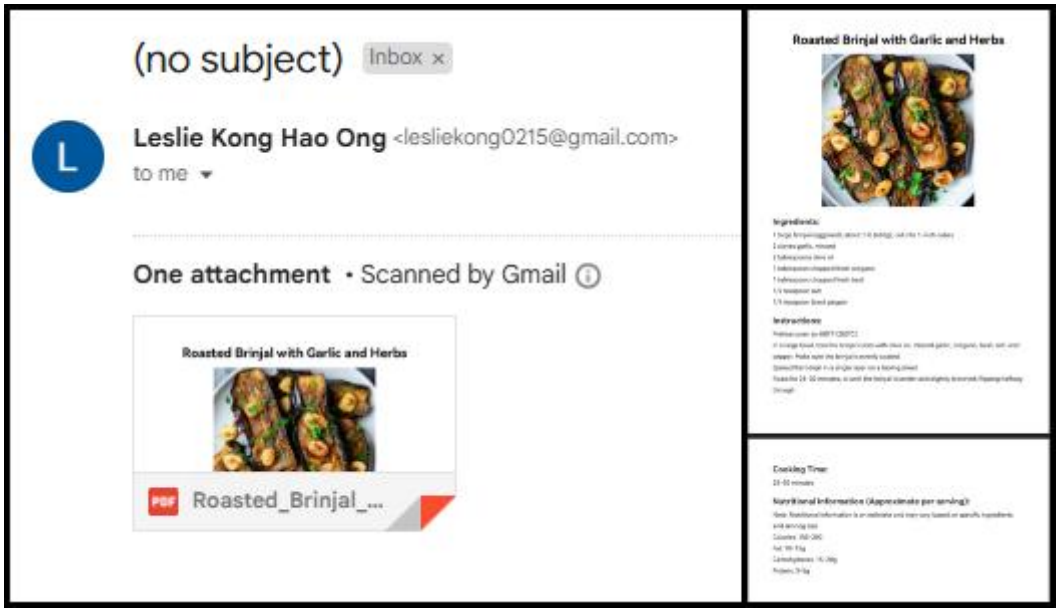


Figure 4.54: Share Recipe through Email

4.3.10 Search Recipes

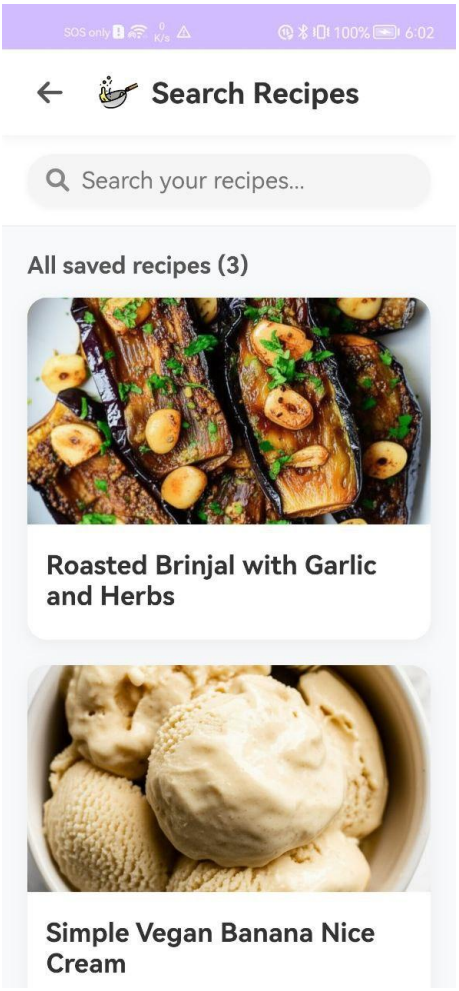


Figure 4.55: Search Recipes Page

Users can access the search recipes page by tapping the magnifier icon below the bottom navigation bar. Inside the page, all the saved recipes will be shown. Users are allowed to search the specific recipe by enter the keyword as below.

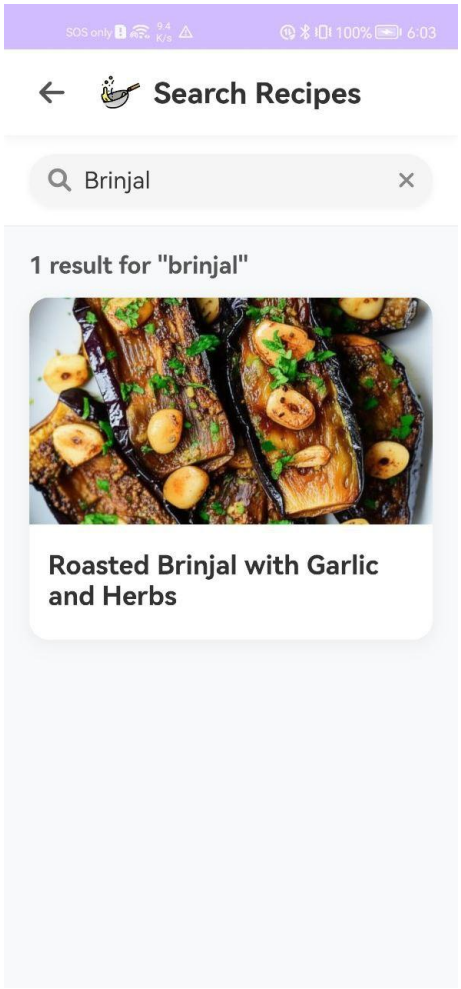


Figure 4.56: Search Recipe by Keyword

4.4 Summary

In Chapter 4, the implementation of the proposed mobile application is outlined in a step by step way. Starting with installation and configuration of main development tools like Android Studio and Miniconda, but also creating virtual environments for Miniconda and using Git to set up the project. Then the chapter explains the configuration of the Android project itself, including the overview of the use of virtual and/or physical devices for testing. It outlines important tasks like building and running the APK, taking care of the AI model and training it with datasets. Accompanying screenshots and interface overviews were included to explain how each area of the system was implemented. Now while the system is fully developed, the next step is a through testing stage to ensure system functionality and performance.

CHAPTER 5: TESTING

5.1 Introduction

Testing carries paramount importance in a software development lifecycle since it guarantees the correctness, performance, and reliability of the application before deployment. This chapter describes the different testing methods being applied to the system that has been developed using the Agile methodology. In keeping with Agile practices, testing was performed on an iterative manner at the end of each sprint so that issues may be uncovered and fixed as soon as possible. The process of testing may be found to fall into either functional or non-functional testing types. Functional testing focuses on a feature or module to ensure that it works as intended and also includes unit and integration testing. Non-functional testing concerns itself with the overall system such as verifying its performance and usability before and after it worked through various scenarios to ensure its stability. Each testing type, the various testing scenarios implemented, and their results in validating the system for practical use are explained in this chapter.

5.2 Functional Testing

Functional testing concentrates on checking that the defined systems respond correctly with respect to the nature of the input and produce the desired output. With this regard, unit tests will focus on single modules while integration tests will check the interaction among those modules. Unit testing isolates each function to test its correctness and debug it at an early stage if required. After each unit behaves as expected, integration testing will check whether the data flow between modules is correct and functional so that communication and functionality across the system are smooth. The tests were

distinguished as manual, wherein the actual results were compared with the expected results to check if the system behaved under defined conditions.

5.2.1 Unit Testing

Unit testing is a method of software testing in which individual components or functions of the application are tested independently to verify that they behave as expected. The purpose of unit testing is to verify that each module or feature of the application operates correctly in isolation prior to being integrated with other components. This methodology identifies bugs early and builds a stable base for further development. In the project at hand, unit testing was performed on some features and functions of the application to secure their accuracy and reliability using a Huawei Nova 5T device. The output results of the various test cases are listed in the tables below to verify that expected results have been attained.

5.2.1.1 Test Cases for the Proposed Mobile Application

Test Case ID: 1 Test Case Objective: To test the register functionality					
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Enter **valid** 'Email', 'Password', 'Confirm Password' to register by tapping the 'Sign Up' button.	• Email: test@example.com • Password: Test1234 • Confirm Password: Test1234	User Account is registered. A message "Registration Successful" will be shown and direct user to the login page.	As Expected	Pass
2	Enter invalid email address format to register by tapping the 'Sign Up' button.	• Email: testexample	Error message "Please include an '@' in the email address. 'testexample' is missing an '@' " will be shown.	As Expected	Pass

3	Enter existing email address to register by tapping the 'Sign Up' button.	• Email: test@example .com	Error message "Email has been used" will be shown.	As Expected	Pass
4	Sign Up with any fields empty	Leave any fields empty.	Error message "Please fill in this field" will be shown.	As Expected	Pass

Table 5.1: Test Case 1

Test Case ID: 2 Test Case Objective: To test the login functionality					
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Enter valid 'Email', 'Password' to login by tapping the 'Login' button.	• Email: test@example .com • Password: Test1234 • Confirm Password: Test1234	User Account is login. A message "Login Successful" will be shown and direct user to the main page.	As Expected	Pass
2	Enter invalid email address format to login by tapping the 'Login' button.	• Email: testexample	Error message "Please include an '@' in the email address. 'testexample' is missing an '@' " will be shown.	As Expected	Pass
3	Enter unregistered/ wrong email address or password to login by tapping the 'Login' button.	• Email: test123@exa mple.com • Password: Test4567	Error message "Invalid Credential" will be shown.	As Expected	Pass
4	Sign Up with any fields empty	Leave any fields empty.	Error message "Please fill in this field" will be shown.	As Expected	Pass

Table 5.2: Test Case 2

Test Case ID: 3 Test Case Objective: To test the edit profile functionality					
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Enter valid 'Name' to update the username.	• New Name: Test	User Profile will show the new name updated.	As Expected	Pass
2	Username field empty.	Leave username fields empty.	Error message "Name cannot be empty " will be shown.	As Expected	Pass
3	Enter valid 'Email' to update the user email address.	• New Email: test123@example.com	User Profile will show the new email address updated.	As Expected	Pass
4	Enter invalid email address format to update the user email address.	New Email: test1234	Error message "Please enter a valid email address" will be shown.	As Expected	Pass
5	Email field empty.	Leave email fields empty.	Error message "Email cannot be empty " will be shown.	As Expected	Pass
6	Enter valid 'Current Password', 'New Password', 'Confirm New Password' to update the password.	• Current Password: Test1234 • New Password: Test666 • Confirm Password: Test666	New User Password has been updated.	As Expected	Pass
7	Enter wrong current password	• Current Password: Test555	Error message "Current Password is incorrect" will be shown.	As Expected	Pass
8	Enter different 'New Password' and 'Confirm New Password'	• New Password: Test666 • Confirm New Password: Test6667	Error message "New passwords do not match" will be shown.	As Expected	Pass

Table 5.3: Test Case 3

Test Case ID: 4 Test Case Objective: To test the logout functionality					
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Logout from the application	• Tap on "Logout" button in User Profile page.	User is logged out and navigated to the Login screen.	As Expected	Pass

Table 5.4: Test Case 4

Test Case ID: 5 Test Case Objective: To test the delete account functionality					
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Delete User Account	• Tap on "Delete" button in User Profile page.	Message "Account Deleted Successfully" will be shown and users will be navigated to login page.	As Expected	Pass

Table 5.5: Test Case 5

Test Case ID: 6 Test Case Objective: To test the ingredient identification functionality					
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Start recognizing the ingredients	• Tap on "Identify Ingredients" button in home page or tap on the cameara icon at the bottom navigation bar.	Camera will be activated.	As Expected	Pass
2	Verify error if camera is unable to open.	• Deny of the permission of camera	Error message "Camera permission not granted" will be shown and navigated back to home page due to the failing of activation of camera.	As Expected	Pass

3	Detect the possible ingredients	• Make sure the ingredients are in camera view	Possible ingredients detected will be shown on the screen in form of checkbox.	As Expected	Pass
4	Verify error handling when no ingredients are detected in the camera view.	• Absence of any ingredients in the camera view	A message of "No objects detected – please take another photo" will be shown.	As Expected	Pass
5	Does not include correct ingredients in the possible ingredients list.	• Take the photo of the ingredients	A cancel button to close the possible ingredients list to let the users retake photo.	As Expected	Pass
5	Include correct ingredients in the possible ingredients list.	• Take the photo of the ingredients	Checkbox and button "add selected" button are allowed the users to choose and add the ingredients into pantry.	As Expected	Pass

Table 5.6: Test Case 6

Test Case ID: 7 Test Case Objective: To test the pantry functionality					
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Show added ingredients	• Tap on "My Pantry" button in home page or tap on the "Pantry" button at the bottom navigation bar.	All added ingredients will be shown in a list.	As Expected	Pass
2	Delete added ingredients.	• Tap on the dustbin icon beside each of the ingredients.	Message of "item removed from pantry" will be shown.	As Expected	Pass
3	Manually add the ingredients	• Enter the ingredients in the field provided and tap on add button.	Message of "item added to pantry" will be shown.	As Expected	Pass

Table 5.7: Test Case 7

Test Case ID: 8 Test Case Objective: To test the creation of recipe functionality					
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Generate Recipe	• Select recipe type and exist ingredients.	A generated recipe depends on the recipe type will be generated.	As Expected	Pass
2	Does not select any recipe type	• Without choosing or inserting any recipe type.	Users are not allowed to go to next step.	As Expected	Pass
3	Does not select any existing ingredients	• Without choosing any existing ingredients.	Users are not allowed to go to next step.	As Expected	Pass
4	Does not exist any ingredients	• Without adding any ingredients in pantry.	Users are not allowed to go to next step.	As Expected	Pass

Table 5.8: Test Case 8

Test Case ID: 10 Test Case Objective: To test the recipe functionality					
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Save Recipe	• Tap on the save recipe button below the generated recipe.	A message of "recipe saved" will be shown and navigate user to home page.	As Expected	Pass
2	Share Recipe	• Tap on the share button below the saved recipe.	Users are allowed to share the recipe in pdf format through social media.	As Expected	Pass
3	Delete Recipe	• Tap on the delete button below the saved recipe.	A message of "recipe deleted successfully" will be shown.	As Expected	Pass

Table 5.9: Test Case 9

Test Case ID: 11 Test Case Objective: To test the search recipe functionality					
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Search Recipe	• Key in the keyword of the recipe in the search bar.	Recipes that fulfil the keywords will be shown.	As Expected	Pass

Table 5.10: Test Case 10

5.3 Non-Functional Testing

Non-functional testing is the main testing type that basically focuses on operational aspects of the intervention application, on how the system performs rather than what it does. Essentially, it must be tested so that it is workable, reliable, efficient, and easy to use under differing circumstances. Whereas a functional testing will test a specific function, non-functional testing will test all the quality attributes, such as its performance, reliability, and general feel about the use. For this project, non-functional testing will include reliability testing to check the system for stability, performance testing to test the speed of the app and its responsiveness, and usability testing to check for user happiness through user acceptance testing (UAT). These tests make sure that the application is fit for public release and can handle actual world usage effectively. The following sections present the techniques and results of each non-functional testing conducted.

5.3.1 Reliability Testing

Reliability testing is a vital arm of non-functional testing to assess the ability of a system to perform consistently without fail in unanticipated situations. With these tests, an effort is made to verify that the proposed application passes with stable performance when caused by invalid inputs, edge cases, or any others that may be considered 'abnormal' conditions. Systems need to make sure that the system responds properly to the errors instead of crashing or producing wrong results. The results help detect any weakness in error handling and become a great aid in improving the overall fault tolerance of the application.

Test Case ID: 12 Test Case Objective: To test the reliability and stability of the mobile application					
ID	Test Case	Input Data	Expected Result	Actual Result	Status
1	Continuous Usage	• Use the app continuously for 2 hours.	App runs smoothly without crashing or slowing down.	As Expected	Pass
2	Incomplete Ingredient Input	• Provide empty input fields.	Empty pantry will be shown.	As Expected	Pass
3	Camera Image Disruption	• Provide blurred or poor-quality image for recognition.	A message of "No objects detected – please take another photo" will be shown.	As Expected	Pass
4	Network Disconnection	• Disconnect to Wi-Fi.	App will not be started properly and data will not shown.	As Expected	Pass
5	Mobile Data Switching	• Switch between mobile data and Wi-Fi mid-process.	App maintains session and continues functioning properly.	As Expected	Pass
6	Low Battery Usage	• Use the app with battery below 10%.	App continues functioning properly.	As Expected	Pass
7	High Load (Multi-tasks)	• Open several app features simultaneously.	App handles multiple operations without crashing.	As Expected	Pass
8	Background Switching	• Minimize and reopen the app.	App resumes previous state correctly.	As Expected	Pass
9	Install/Uninstall Cycle	• Install and uninstall app multiple times.	App installs/uninstalls cleanly without issues.	As Expected	Pass
10	Recipe Data Integrity	• Save and retrieve generated recipe.	Data is stored and retrieved accurately.	As Expected	Pass

11	Error Message Handling	• Trigger known errors (e.g., invalid API request).	Displays proper error messages without crashing.	As Expected	Pass
12	Database Connection Loss	• Disconnect from database during operation.	App handles error and retries or notifies the user.	As Expected	Pass

Table 5.11: Test Case 11

5.3.2 User Acceptance Testing

User Acceptance Testing (UAT), being a part of non-functional testing, serves the purpose of assessing the usability and satisfaction the system can offer to an end-user. It primarily aims to ensure that, from the end-user perspective, the mobile application does meet user expectations and functional requirements when an actual situation arises. The objective of UAT testing in this project is to make sure that the application works well on devices and offers accurate results along with a user-friendly and helpful experience.

These 15 participants supposed to give their views on the apps concerning functionality, interface layout, ease of use, and general satisfaction were given the APK file, user manual, and a structured Google Form survey for the evaluation. Android 8.0 (Oreo) or above were chosen to be able to run the app smoothly. After installing and using the app, all testers finished the survey and provided feedback from their personal experience. The responses were then studied to highlight common user concerns, verify core feature efficacy, and identify potential areas for further improvements to be made before the app was released publicly.

5.3.2.1 Analysis on User Acceptance Survey

Feedback was gathered via a questionnaire specifically designed to assess multiple aspects of the Smart Recipe Generator application. The questionnaire requested feedback in five sections: User Demographics, Usability and Interface, Functionality, Performance, and Feedback and Recommendations as shown in Appendix C. The following is a comprehensive analysis of the results received from 15 respondents.

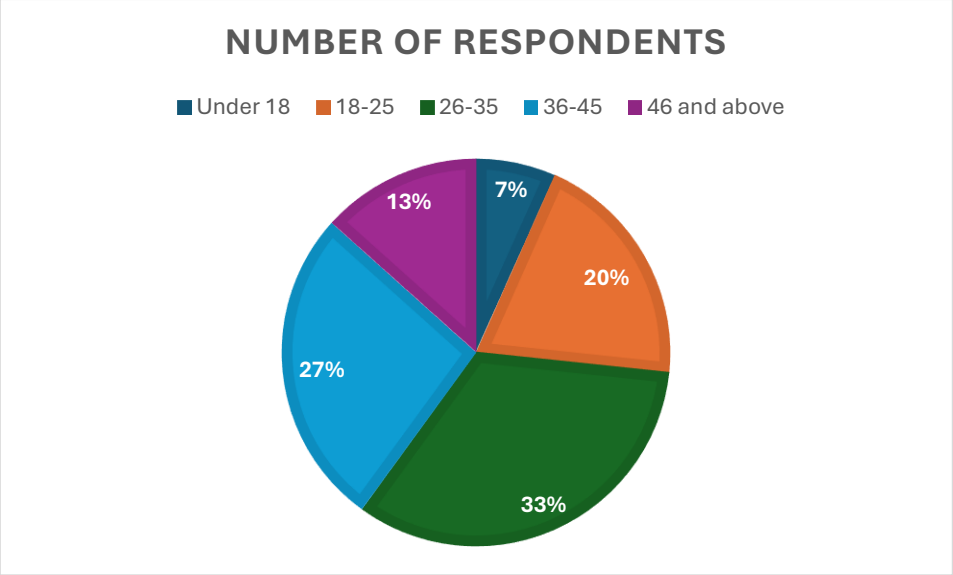


Figure 5.1: User Demographics

To investigate the usability and satisfaction with the proposed mobile app application, 15 respondents took part in the User Acceptance Testing (UAT). Respondents were selected across a range of ages to provide a more realistic population sample. Among the 15 respondents there were 5 respondents were aged 26–35 and 4 respondents were aged 36–45, which would represent the majority of active parents and the primary users of the app. Three of the participants were aged 18–25 representing younger adults and new parents who are often more familiar with mobile technologies. The remaining respondents included two who were aged 46 and over, and one respondent who was under 18. With the age representation across these participants, even with some narrow applicant reflection, the feedback gathered should reflect a wide representation of user experiences and expectations, thereby providing a more holistic assessment of the overall usability and performance of the application.

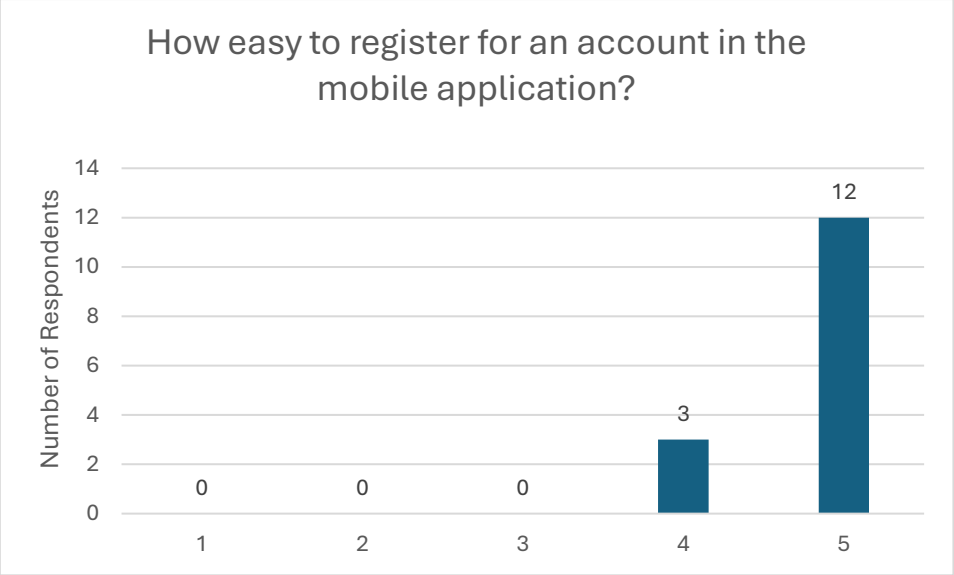


Figure 5.2: Ease of Registering an Account

Most respondents perceived the registration process as very easy to understand. Of the 15 testers in this study, 12 (80%) rated the registration process as very easy (Level 5), and the remaining 3 (20%) rated it easy (Level 4). Furthermore, none of the respondents had significant challenge during registration, which suggests the sign-up process in the mobile app is designed for ease of understanding and logically sequenced.

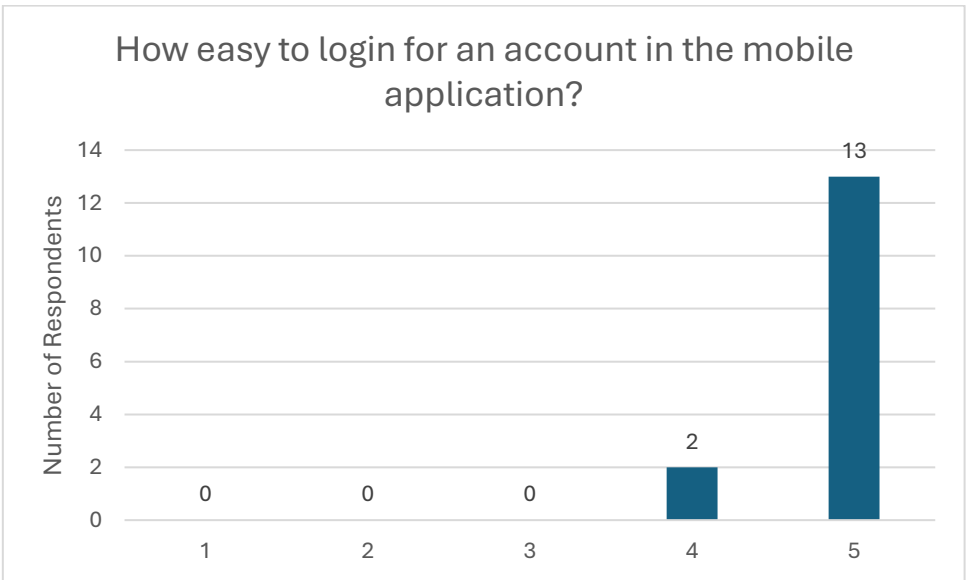


Figure 5.3: Ease of Logging into an Account

User responses to the login process were overwhelmingly and unequivocally positive. There were 13 respondents (87%) that rated the login component to be very

easy (Level 5) and 2 respondents (13%) that rated the login component to be easy (Level 4). In summary, these responses suggest that although the user authentication system is very reliable, it is also simple to use and allows end users to login seamlessly with minimal fuss.

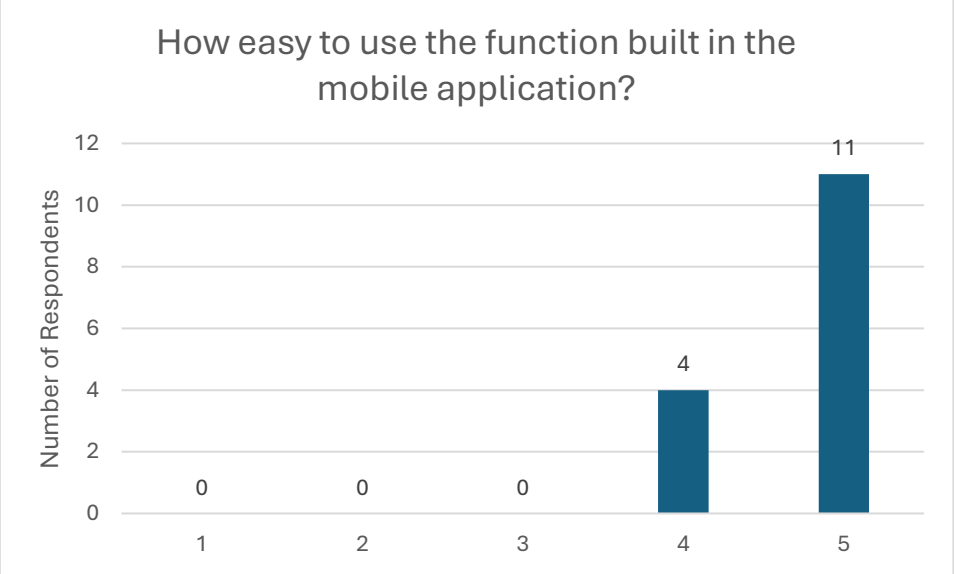


Figure 5.4: Ease of Using Built-in Functions

Concerning overall usability of features, most users were satisfied: 11 types of (73%) considered the use of built-in functions 'very easy' (level 5), while 4 users (27%) considered the use easy (level 4). These results reveal relatively simple, accessible features in the mobile application with a short learning curve resulting in an overall positive user experience.

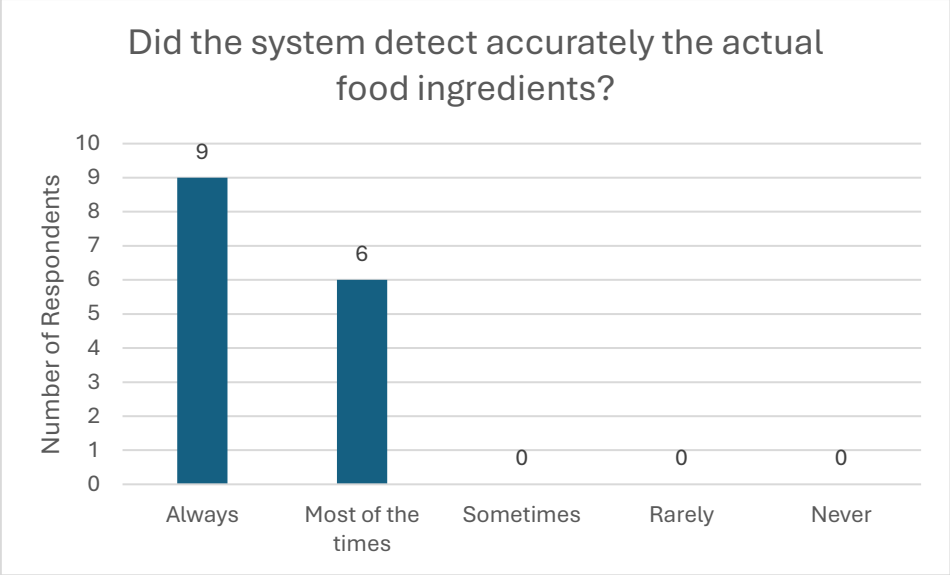


Figure 5.5: Detection Accuracy of Food Ingredients

The findings of the user acceptance test indicated considerable satisfaction with the ingredient identification accuracy of the system. A total of 9 of the respondents (60%) stated that the application always accurately detected the real food ingredients and the remaining 6 respondents (40%) said it accurately detected the actual food ingredients most of the times. None of the users indicated a lower satisfaction level, indicating the AI detection feature is reliably accurate, and performed consistently under normal usage.

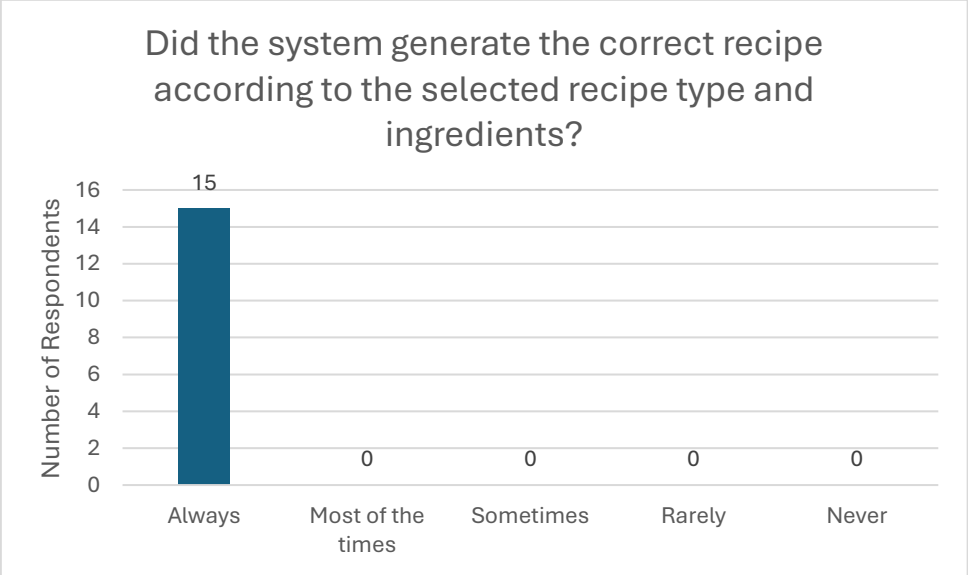


Figure 5.6: Accuracy of Recipe Generation

While asked if the system created the correct recipe for what type and ingredients the users selected, every respondent gave "Always" as an option. All (100%) respondents believed the system showed high levels of accuracy in creating recipes for users. This consistency demonstrates the application's recommendation engine corresponds with user expectations and selected inputs, which suggests it is reliable for day-to-day use.

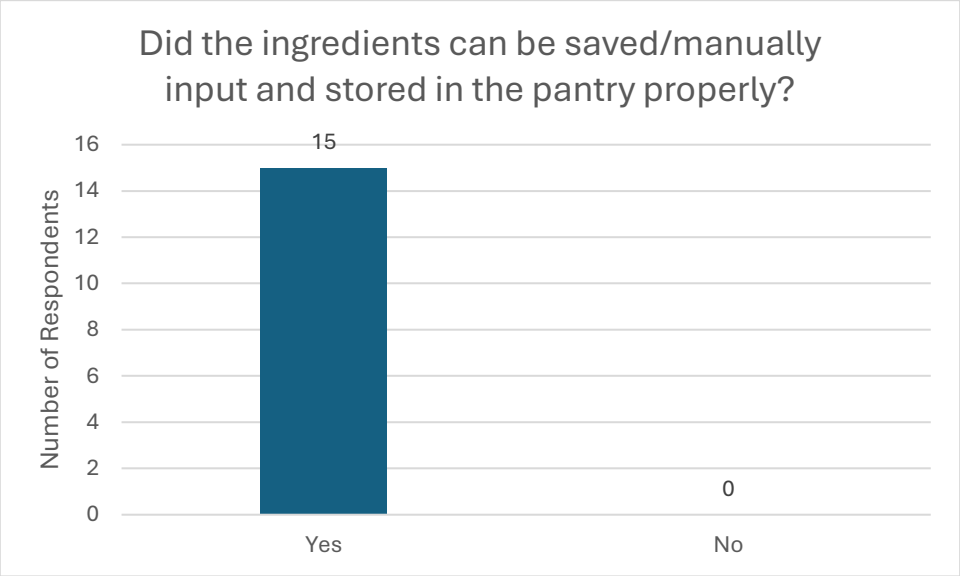


Figure 5.7: Ingredient Storage in Pantry

All 15 users (100%) said "Yes" when asked if the ingredients could be saved and/or inputted manually and stored correctly in the pantry feature. This demonstrates reliability in the application's pantry management functionality, as users were able to effectively manage the ingredients they have on hand through the app without difficulties or data loss.

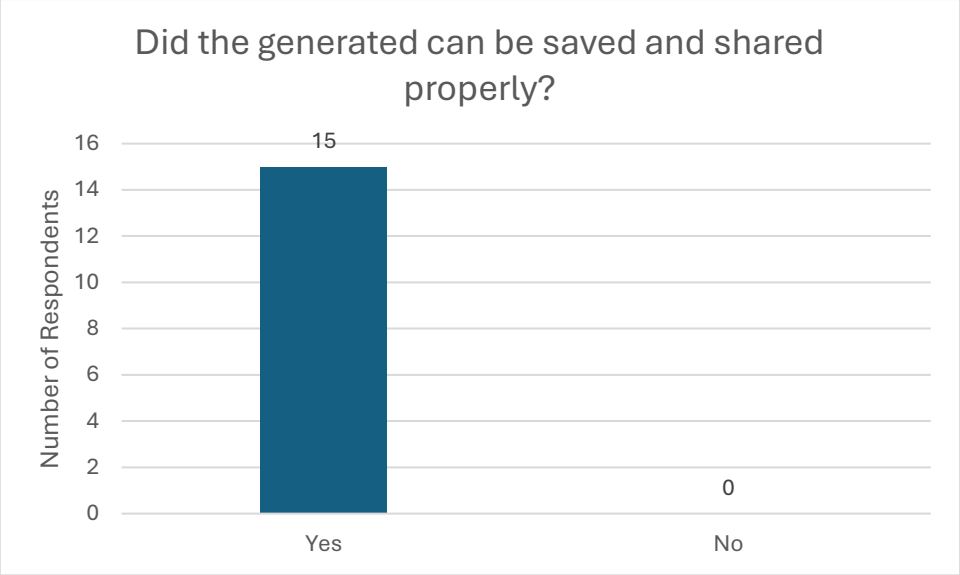


Figure 5.8: Recipe Saving and Sharing

Similarly, the ability to save and share generated recipes received a perfect score. All survey participants (100%) indicated that they were able to save and share recipes as they were supposed to. This indicates that users experienced no problems with this essential function and highlights the utility of the system as well as its user-centered design features.

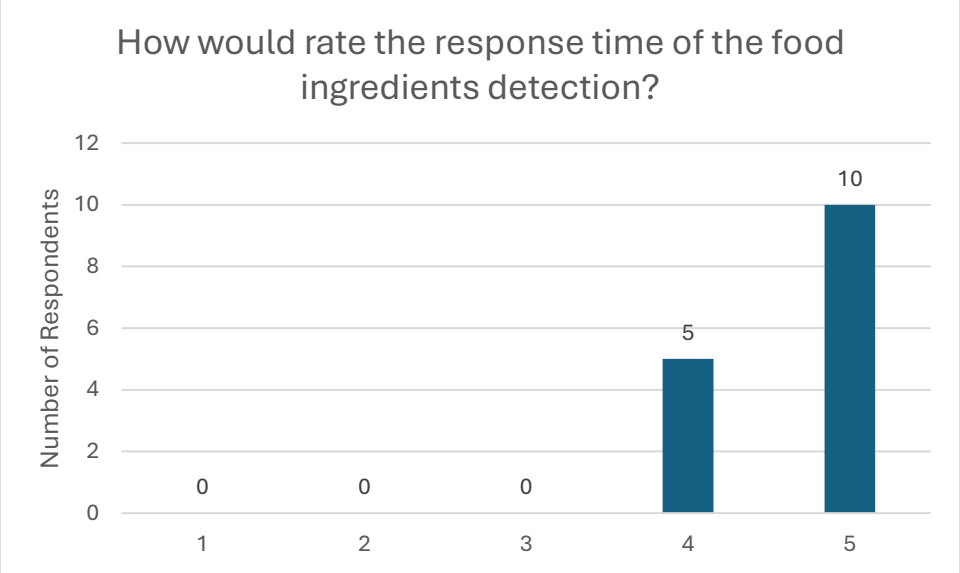


Figure 5.9: Response Time for Food Ingredients Detection

The application received positive feedback regarding the speed of food ingredient detection. Of the 15 users who participated, 10 of the respondents (67%)

reported that the detection speed was very fast. The remaining 5 users (33%) described it as fast. None of the users classicfied the speed as moderate or slow. These results suggest that the system has successfully analyzed and identified ingredients in near real-time, which would improve the overall user experience.

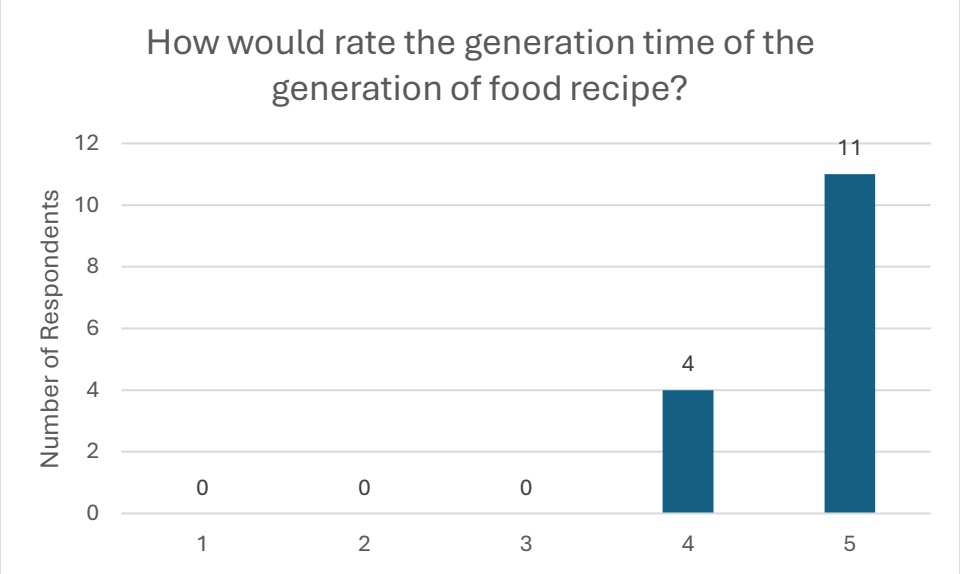


Figure 5.10: Recipe Generation Speed

Regarding the recipe generation performance, the system performed very well. Overall, 11 users (73%) rated the generation speed as being very fast. The 4 remaining users (27%) rated the speed as fast. There were no subjects reporting latency or slow performance at all, which shows that the system processes data quickly and generates recipes with little waiting time. This helps to create a smooth and responsive experience for users planning their meals.

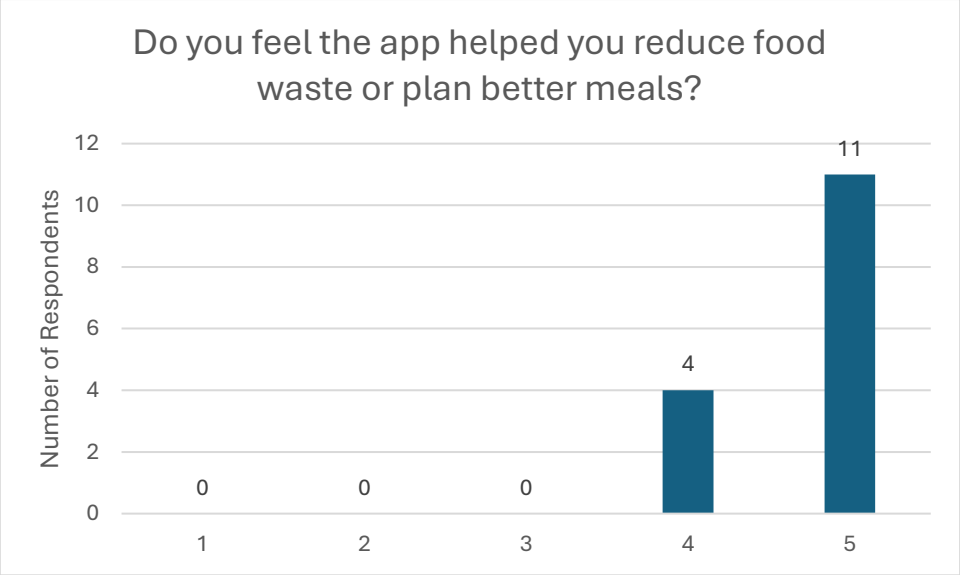


Figure 5.11: Usefulness in Reducing Food Waste and Meal Planning

The results show solid agreement on the app's ability to reduce food waste and facilitate better meal planning. Of the 15 participants, 11 users (73%) strongly agreed that the app had positively affected their food management and four users (27%) agreed, meaning that the central goal of helping users use ingredient amounts efficiently while reducing food waste has been accomplished.

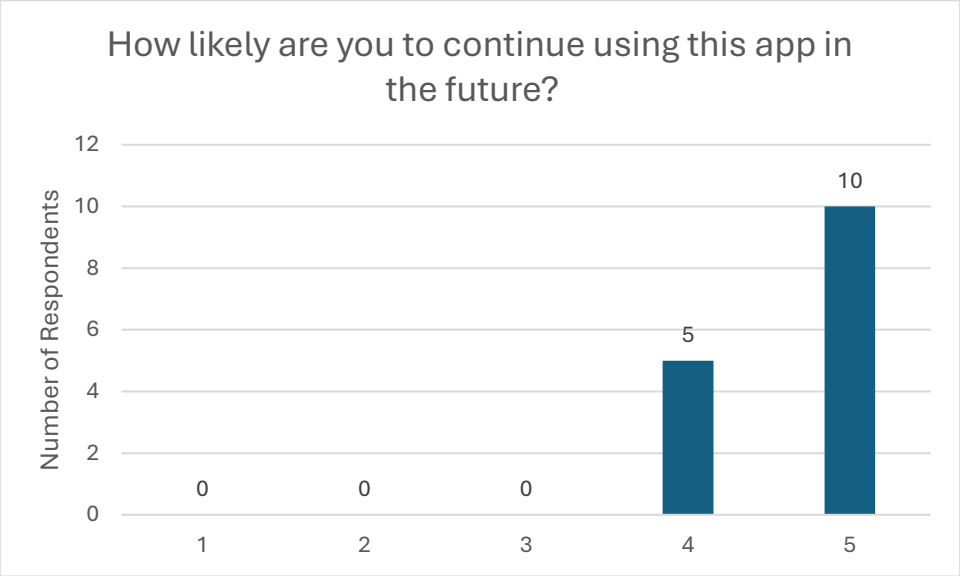


Figure 5.12: Likelihood of Continued Use

When asked about future use of the app, a large majority of respondents indicated they would continue to use the app. 10 users (67%) rated their likelihood of using the app again very likely and 5 users (33%) rated likely. This indicates at least a

high level of satisfaction and perceived value with the features and performance of the application.

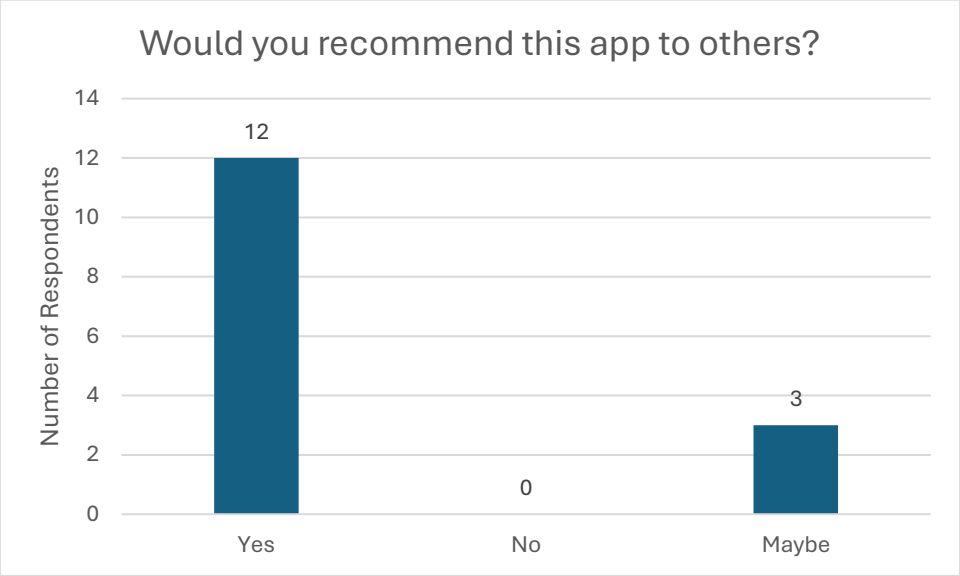


Figure 5.13: Willingness to Recommend the App

In terms of recommendations, 12 respondents (of 15 respondents) stated they would recommend the app to other users, while 3 users responded with "maybe." No users chose "no," perhaps suggesting the overall positive experience with the app. All of which indicates a strong level of trust and acceptance by users in app's functionality and user experience.

5.4 Discussion on Results

The overall goals of the proposed mobile application have mostly been satisfied; however, there are limitations that affect its suitability for larger-scale deployment or wide-ranging public deployment or release. The primary challenge that emerged during testing and use was a sub-optimal detection performance, especially in terms of real-time applications. The model currently has a lot of overhead time in processing, which may lead to delays in response and hinder the overall user experience.

Another significant issue would be the deep learning model's limited capability to recognize ingredients. The overall model can detect a maximum of 120 different food

ingredients, which currently does not allow for a thorough enough sample for many recipe generation scenarios. Users with a range of cooking needs or with specific categories of ingredient combinations may find this limitation to be restricting and ultimately impacting the quality and diversity of the recipes generated.

Despite these limitations, the system had a satisfactory level of performance in controlled situations (for example, relevant and clear images of ingredients and controlled lighting). Unfortunately, in multiple uncontrolled environments—varying degrees of lighting, blurry images and not recognizable food types (for example unfamiliar ingredients)—it appeared that the detection rates would decline, and the user would receive misleading recommendations. The additional challenge of not having different ingredient classes in the initial dataset also restrained broader question representation from that dataset and did not allow for a good cross-generalization of the proposed application across food types.

From a technical standpoint, the performance problems are due to the computational requirements of the current AI model, as the current model is not mobile-optimized. Lightweight models such as MobileNet or quantized versions of YOLO have been discussed, and more experimentation will be needed to make sure a balance can be struck between speed and accuracy. Also, the model must be expanded and retrained in the future to increase the number of ingredients, as at least this number would have to be high for the project to be usable in a practical context.

Overall, the system, while it provides a solid gateway to an intelligent ingredient recognition and recipe prediction system, requires improvement in two key areas: detection speeds and ingredient recognition coverage. These improvements will be necessary for the system to properly function and work in real-world situations. Future

iterations should focus on optimizing the AI model for speed, and should also work to expand the ingredient dataset to improve the system in terms of its ability to do what it was built for and to improve user engagement.

5.5 Summary

In summary, Chapter 5 has described a thorough assessment of the mobile application's proposed functions and features, from both functional and non-functional testing perspectives. Both forms of functional testing, including unit and integration testing, confirmed that the features that were evaluated - registration and login, detection of ingredients, pantry, and recipes - were correct and reliable as functional features within the mobile application. The functional tests achieved all expected results within the prescriptive characteristics of all of the tests. The body of non-functional testing has reinforced reliability, performance, and overall usability under varying conditions. From the reliability testing, it was demonstrated that the app consistently performed and handled errors gracefully. This Mobile application was further evaluated with 15 participants that participated in User Acceptance Testing (UAT). The participants gave very positive feedback on the usability, interface, functionality, and overall performance of the app. However, it was noted that the system had limitations with the speed of detecting ingredients and the AI model's limitations of recognizing more than 120 food ingredients. This suggests that future improvements are needed with the model to optimize for real-time mobile usage and training on more food ingredients. Overall, the mobile application demonstrates core functionalities and capabilities, and potential improvements, developments, and refinements have been outlined in this study.

Chapter 6: Conclusion and Future Work

6.1 Introduction

Summarizing the achievements, the chapter then concludes the development of the proposed mobile application, thinking about the entire system development process. The realization and testing phases elucidate how far the project objectives have been realized. On the other hand, possible areas of enhancement will be identified along with suggestions for future additions that would be beneficial for the system. Consequently, the discussion acts as a launchpad for future considerations towards improving the application in terms of performance, functionality, and user experience.

6.2 Objectives and Achievements

The suggested mobile application was developed with the propositional goal, and secondary goals established in Chapter 1. This section outlines how each of these goals was met and achieved during the project implementation phase.

The propositional goal of this project was to develop a mobile application that operates with Deep Learning, providing users with user-specific meal suggestions that align with ingredients available, dietary restrictions, and the goal of minimizing food waste. This goal has successfully been accomplished through the implementation of AI model that could be manipulated to understand ingredient input and provide the appropriate recipe suggestions accordingly.

Other specific objectives were also accomplished:

Objective	Achievement
To correctly identify fruits and vegetables from pictures taken by users using deep learning algorithms	A deep learning model was trained and implemented to recognize up to 120 common fruits and vegetables from user-submitted images with high accuracy under ideal conditions.
To generate new and interesting recipes based on user input and known items by using the Gemini API	The system successfully integrates with Gemini API to create diverse and contextually relevant recipes tailored to the recognized ingredients and user preferences.
To enable users to input dietary restrictions and adapt the recipe suggestions accordingly	A dietary preference module was developed, allowing users to filter and adapt recipes based on restrictions such as vegetarian, vegan, and gluten-free needs. The generated recipes dynamically adjust based on these user settings.

Table 6.1: Achievements

In general, the application fulfils the purpose, or functionality and performance requirements that were specified. This application not only provides an identification of the ingredients and a personalized recipe suggestion, but it also is the experience of the user, feedback on clear interfaces and performance yield consistent results throughout all phases of testing. These are significant achievements that highlight that

the original intentions, or objectives, were completed well within the parameters of the project.

6.3 Project Limitations and Constraints

Besides meeting the main objectives set forth at the beginning of this project, some limitations and constraints were encountered during the phases of development, testing, and user evaluation. These drawbacks affect the performance and usability of the proposed mobile application and should be considered in future improvements so as to attain maximum user experience and wide deployment.

1. Limited Ingredient Recognition Scope

The deep learning model that has been implemented in the application has currently been trained to recognize up to 120 different types of fruits and vegetables. This range encompasses most of the commonly used raw ingredients, yet it does not include processed or packaged items, or lesser known local ingredients. Therefore, users may find challenges with identifying certain ingredients not accounted for in the training dataset. This limitation puts constraints on the app's applicability to wider culinary situations, particularly users with culturally diverse diets or specific dietary requirements.

2. Slow Detection Performance

In the testing iterations, and by reviewing user feedback, speed of detection emerged as a clear issue. The model takes a noticeable time to process an image and return results (especially on mid-range smartphones) after an image is captured. While the time to process an image is limited by how much compute power is available to run the image-based deep learning inference on-device, it does affect the flow of the application as a whole and can lead to user dissatisfaction, especially when scanning multiple

ingredients all at once in the same session. There is a clear performance optimization area with maximum possible near real-time processing.

3. Device Dependency and Hardware Constraint

Currently, the application is developed and tested only on the Android platforms. This means the application does not support iOS and will have a limited potential user base. Also, the AI models used in seeking ingredient recognition rely upon using TensorFlow Lite for mobile applications. Therefore, the application's performance directly correlates to the processing capabilities of the mobile device. Mobile devices with less powerful or slower specifications could lead to delays and stuttering UI. It is possible that, for lower-end users, the app could even crash when running the ingredient recognition capability, leading the users to stop using the app.

4. Absence of a Cloud-Based Recognition Alternative

All AI inference is done on-device, which has much to do with some of the limits described. Currently the application has no capability to offload the computation to the cloud, which could increase recognition speed, allow for larger models and lessen hardware limitations on the user side. Cloud inference also adds limitations including the need for internet, and also greater security risks.

5. Static Dataset Limitation

The training dataset is fixed and cannot be updated dynamically (from the app interface). In other words, users cannot provide new data cases or correct the data to enable system learning over time. This limit restricts the evolution of the model and adaptability to new or region-specific ingredients, ultimately hindering personalization and accuracy across different user groups.

6. No Multi-Platform or Web Version Support

The app operating as only a mobile app with no web-based companion to manage data, view history, or sync between devices, limits the app's flexibility for users who want to manage a pantry or plan meals in a desktop/tablet setting.

In conclusion, while the proposed system delivered its main function and met the major objectives of ingredient recognition, AI based recipe generation, and dietary based filtering, there are still limitations that have been found that show future work needs to address the key issues. If these issues can be overcome, it will increase the satisfaction of prospective users, expand the potential user base, and improve the usefulness of the application in real-world scenarios.

6.4 Future Work

Although the proposed application has achieved its primary objectives to a significant degree, there is much potential for improvement in order to improve its performance, usability, and scalability. These future developments seek to enhance the app's deep learning feature, broaden its functionality, and improve its responsiveness to real-world applications.

One of the areas that need to be addressed is expanding the food ingredient recognition. It can currently recognize 120 ingredients, which might be its limitation for individuals that require more extensive scope of ingredient recognition. The future development would entail expanding the dataset and training the model to recognize more varying foods, including packaged food and local dishes.

Another area that can be improved is performance. The current system experiences moderate latency when detecting objects. Utilizing lighter frameworks such as MobileNetV2 or quantized TFLite models to optimize the deep learning model

should heavily reduce latency, thereby allowing faster real-time processing on lower-end smartphones.

Additionally, more complete support for platforms should be included. Only Android is supported currently, so new development would include iOS support to expand the number of users and provide more users with the benefits of the app.

User customization features can also be enhanced. The future can see added such as custom dietary recommendations, allergy marking, filterable recipes, and user-stored preferences to enhance the relevance of the meal suggestions.

Another feature of improvement that has been proposed is cloud-based inference. This involves offloading model running onto cloud servers, reducing on-device processing load and will enable more sophisticated models, particularly in real-time speed compared to lightweight models.

Improvements on the interface can be done through such as making layouts more intuitive, being able to track recipe progress, recipe sharing abilities, and tutorials built-in for new users-all of which would enhance usability, engagement, and long-term use.

In addition, a new feature is proposed to allow users to generate recipes strictly using only the selected food ingredients without automatically suggesting extra ingredients. This option gives users the flexibility to switch between “flexible” and “strict” recipe generation, depending on their preference. This can be achieved by sending different prompts to the Gemini API depending on the user's chosen mode.

Below are the summary of Future Work:

- Expand ingredient detection beyond 120 food items with a larger dataset.
- Optimize model for faster detection using lightweight or quantized architectures.
- Develop iOS version for cross-platform accessibility.
- Add customizable dietary preferences and allergy filters.
- Enable cloud-based model inference to reduce device load.
- Introduce interactive UI features like bookmarks, progress tracking, and community recipe sharing.
- Improve model training for more diverse food categories and complex dishes.
- Include onboarding tutorials and help guides for first-time users.
- Add strict recipe generation option using only selected ingredients, configurable by user mode.

6.5 Summary

Chapter 6 encapsulates the developmental journey of the proposed AI-based mobile application in a reflection on the objectives, limitations, and potential improvements. Chapter 6 begins by asserting that the project has been successful in realizing its primary purpose of proposing individual meals based on ingredient availability and diet constraints. All the objectives articulated—identification of ingredients, recipe creation using Gemini API, and modification according to diet—have been achieved. But certain limitations were found, i.e., limited scope of recognition (only up to 120 ingredients), lower detection speed, reliance on the device, and lack of cross-platform or cloud support. These limitations indicate that there is even more to be improved in future work. Lastly, the chapter proposes a couple of enhancements for the future, i.e., expanding the recognition capability of the model, enhancing performance with smaller

models, adding processing from the cloud, and expanding platform support to iOS. In general, this chapter outlines not just the achievements of the project but also lays a foundation for further development to ensure broader applicability and better performance.

REFERENCES/BIBLIOGRAPHY

ACApplications, LLC. (2023). *Chefapp - Ai Recipe Creator*. App Store. https://apps.apple.com/us/app/chefapp-ai-recipe-creator/id6450523267.

Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., & Thomas, D. (2001). The Agile Manifesto. Retrieved from https://agilemanifesto.org.

Bruton, L. (2022). What is wireframing? A complete guide. Retrieved from https://www.uxdesigninstitute.com/blog/what-is-wireframing/

Buduan, K. (2019, August 29). *Huawei Nova 5T: 'Fashionable' and powerful*. GadgetMatch. https://www.gadgetmatch.com/huawei-nova-5t-specs-price-availability/

Chi, C. (2024, September 27). A Beginner's Guide to Data Flow Diagrams. *Hubspot*. https://blog.hubspot.com/marketing/data-flow-diagram

Chia, A. (2023). What is Data Dictionary? Retrieved from

Chien, Y. (2020). *Effective requirement planning in Agile software development*. Journal of Software Development Studies, 12(3), 45-56.

Cockburn, A. (2006). *Agile Software Development: The Cooperative Game*. Addison-Wesley Professional.

Codecademy. (2024.). *What is SQLite?* Codecademy. https://www.codecademy.com/article/what-is-sqlite

Contributor, T. (2023, January 23). *Android Studio*. Mobile Computing. https://www.techtarget.com/searchmobilecomputing/definition/Android-Studio

Cprime Inc. (2023, April 17). *What is Agile? - What is Scrum? - Agile FAQ's | Cprime*. Cprime. https://www.cprime.com/resources/what-is-agile-what-is-scrum/

Developers, A. (2011). What is android. *Dosegljivo: http://www. academia. edu/download/30551848/andoid--tech. pdf*.

Drexler, O. (2022, November 3). The Pros and Cons of Android Studio and App Tools. Pangea. https://pangea.ai/resources/pros-and-cons-of-android-studio-and-app-tools

Durdov.B, Prvan.M, Čoko.D and Musić.J, "Training Dataset Pruning Algorithm with Evaluation on Medical Datasets," *2024 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, Croatia, 2024, pp. 1-8, doi: 10.23919/SoftCOM62040.2024.10721725.

Eclipse Apps LLC. (2023). *MyPantry: Generate Recipes*. App Store. https://apps.apple.com/us/app/mypantry-generate-recipes/id6463172475.

Egeonu, E. (2022). The Advantages and Disadvantages of the RAD Model. Retrieved from https://distantjob.com/blog/rad-model-advantages-and-disadvantages/

Fitzgerald, C., Meiselman, H. L., & Lynch, P. (2020). *The Role of Personalization in Meal Planning Applications for Health and Wellness*. Journal of Consumer Health, 14(1), 65-79.

Food and Agriculture Organization of the United Nations. (2021). Global food losses and waste. https://www.fao.org/fsnforum/resources/reports-and-briefs/global-food-losses-and-waste/

Food ingredients Computer Vision Dataset by seongmin. (n.d.). Roboflow. https://universe.roboflow.com/seongmin/food-ingredients-pnisb/dataset/1/images

Gargenta, M. (2011). *Learning android*. " O'Reilly Media, Inc.".

GeeksforGeeks. (2023, June 15). *What is a Flowchart and its Types?* GeeksforGeeks. https://www.geeksforgeeks.org/what-is-a-flowchart-and-its-types/

GeeksforGeeks. (2024, March 21). *Context diagrams*. GeeksforGeeks. https://www.geeksforgeeks.org/context-diagrams/

Glaschenko, A. (2023). What is Rapid Application Development. Retrieved from https://www.jmix.io/rapid-application-development/

Grand View Research. (2023). Mobile Application Market Size, Share & Trends Analysis Report by store (Google Store, Apple Store, others), by application (Gaming, Music & Entertainment, Health & Fitness, social Networking), and region, segment Forecasts, 2024 - 2030. https://www.grandviewresearch.com/industry-analysis/mobile-application-market/

Hanna, K. T., & Biscobing, J. (2024, March 26). *entity relationship diagram (ERD)*. Search Data Management. https://www.techtarget.com/searchdatamanagement/definition/entity-relationship-diagram-ERD

Hetler, A. (2024, July 31). What is ChatGPT? https://www.techtarget.com/whatis/definition/ChatGPT

Highsmith, J. (2009). *Agile Project Management: Creating Innovative Products*. Addison-Wesley Professional.

https://www.evertop.pl/en/frontend-vs-backend/

https://www.splunk.com/en_us/blog/learn/data-dictionary.html

Kang, B. (2023). System Design and System Architecture. Retrieved from https://bootcamp.uxdesign.cc/system-design-and-system-architecture-e963d030bc7b

LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015). https://doi.org/10.1038/nature14539

Lee, J. L., & Kim, Y. (2024). Evaluation of Mobile Applications for Patients with Diabetes Mellitus: A Scoping Review. Healthcare, 12(3), 368. https://doi.org/10.3390/healthcare12030368

Lee. R. (2019, August 2019). *Huawei Nova 5T Malaysia: Everything you need to know*. SoyaCincau. https://soyacincau.com/2019/08/27/huawei-nova-5t-malaysia-official-launch/

Legierski, B. (2021). Frontend vs Backend. Retrieved from

Makadia, H. (2023). Your Complete Guide To Rapid Application Development (RAD). Retrieved from https://marutitech.com/rapid-application-development/

Mintel. (2014). Non-celiacs drive gluten-free market growth. https://www.mintel.com/insights/food-and-drink/gluten-free-consumption-trends/

Olson, A. (2023). *Dishgen: Ai Recipes*. App Store. https://apps.apple.com/jp/app/dishgen-ai-recipes/id6473455744?l=en-US.

Orbograph. (2020, August 7). Understanding AI: What is a Deep Learning Node? | OrboGraph. *OrboGraph*. https://orbograph.com/understanding-ai-what-is-a-deep-learning-node/

Paganini, C. (2019). Primer: Understanding Software and System Architecture. Retrieved from https://thenewstack.io/primer-understanding-software-and-system-architecture/

Pang, B., Nijkamp, E., & Wu, Y. N. (2019). Deep Learning With TensorFlow: A Review. Journal of Educational and Behavioral Statistics, 45(2), 227–248. doi:10.3102/1076998619872761

Patterson, J., & Gibson, A. (2017). *Deep Learning: a Practitioner's approach*. http://cds.cern.ch/record/2282134

Pradiptamu, K. (2023). What is a Flowchart and its Types? Retrieved from https://www.geeksforgeeks.org/what-is-a-flowchart-and-its-types/

Robinson, S., & Nolle, T. (2024, August 2). *What is a data flow diagram (DDF)?* Search Data Management. https://www.techtarget.com/searchdatamanagement/definition/data-flow-diagram-DFD

S, L. (2024, December 31). Interaction between frontend and backend: you should know. *GUVI Blogs*. https://www.guvi.in/blog/interaction-between-frontend-and-backend/

S, R. A. (2024, July 7). *What is SQLite? Everything You Need to Know*. Simplilearn.com. https://www.simplilearn.com/tutorials/sql-tutorial/what-is-sqlite

Salix Dijital Pazarlama Anonim Sirketi. (2023). *Delicio: Ai Chef, Food Recipes*. App Store. https://apps.apple.com/us/app/delicio-ai-chef-food-recipes/id6445913119.

Solanki, A. (2024, April 11). *CNN vs. RNN vs. ANN: A comprehensive introduction*. Softwebsolutions. https://www.softwebsolutions.com/resources/difference-between-cnn-rnn-ann.html

Sreekumar, D., & Sreekumar, D. (2024, December 6). What is Research Methodology? Definition, Types, and Examples. Paperpal Blog. https://paperpal.com/blog/academic-writing-guides/what-is-research-methodology

United Nations Environment Programme. (2021, March 4). UNEP Food Waste Index Report 2021. UNEP - UN Environment Programme; UNEP. https://www.unep.org/resources/report/unep-food-waste-index-report-2021

What is SQLite? (2024, October 26). SQLite Tutorial. https://www.sqlitetutorial.net/what-is-sqlite/

What is TensorFlow? (n.d.). NVIDIA Data Science Glossary. https://www.nvidia.com/en-

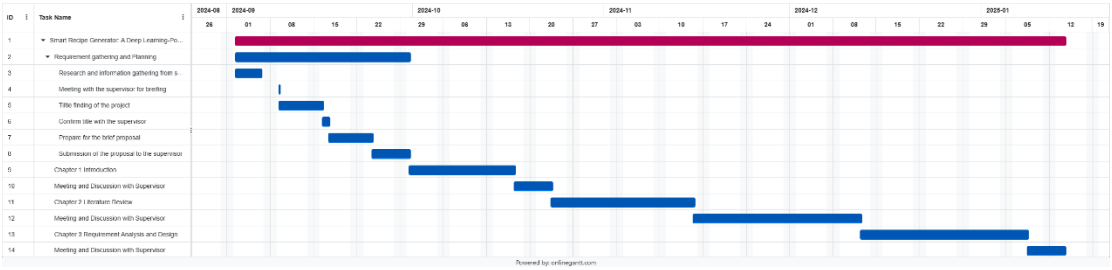
us/glossary/tensorflow/?srsltid=AfmBOoqsSMqQ_0Bkp13WkvyPbRuVWl4sMkDTLVVz6uhkyjYsXY0rQv01

Wijebahu, D. (2024, February 28). Developer's guide to Android emulator setup and secrets tips. *Medium*. https://medium.com/@wijebahuwmpwdgb.20/developers-guide-to-android-emulator-setup-and-secrets-tips-501c44cf25a2

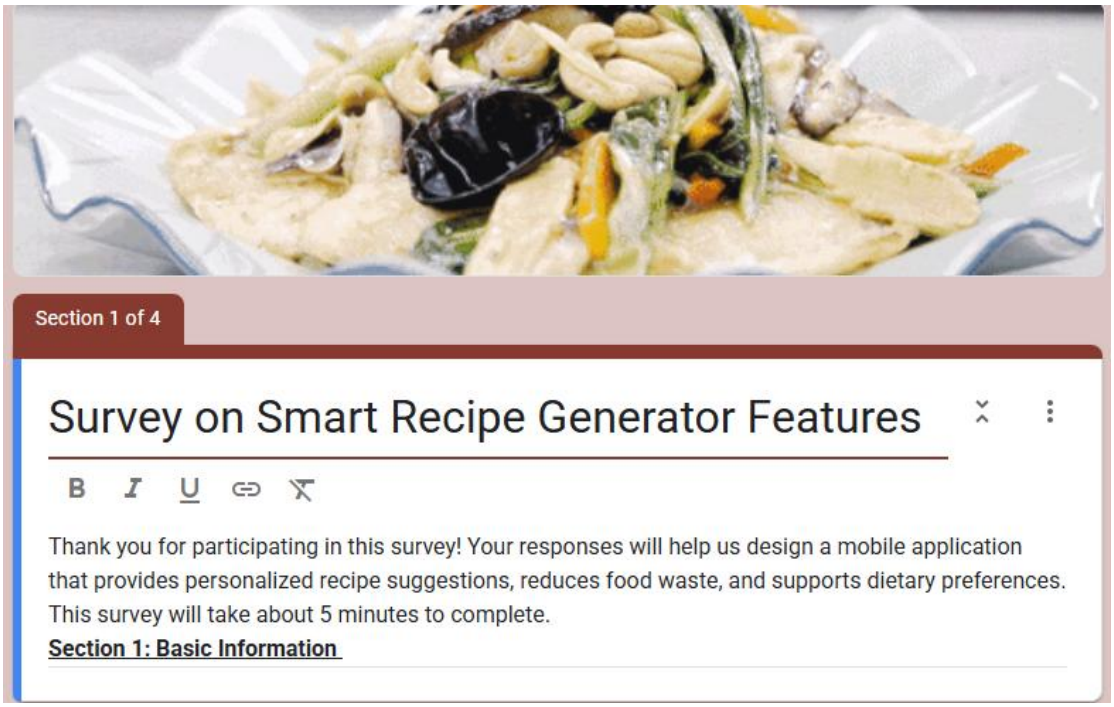
Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., & Xiao, J. (2015). Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

APPENDICES

Appendix A



Appendix B



Section 1: Basic Information

What is your age group? *

- ○ Under 18
- ○ 18–25
- ○ 26–35
- ○ 36–45
- ○ 46 and above

What is your cooking frequency? *

- ○ Daily
- ○ A few times a week
- ○ Occasionally
- ○ Rarely

Do you use any mobile apps for meal planning or recipe ideas? *

- ○ Yes
- ○ No

Section 2 of 4

Section title (optional)

Section 2: Current Practices and Challenges

How do you currently decide what to cook? *

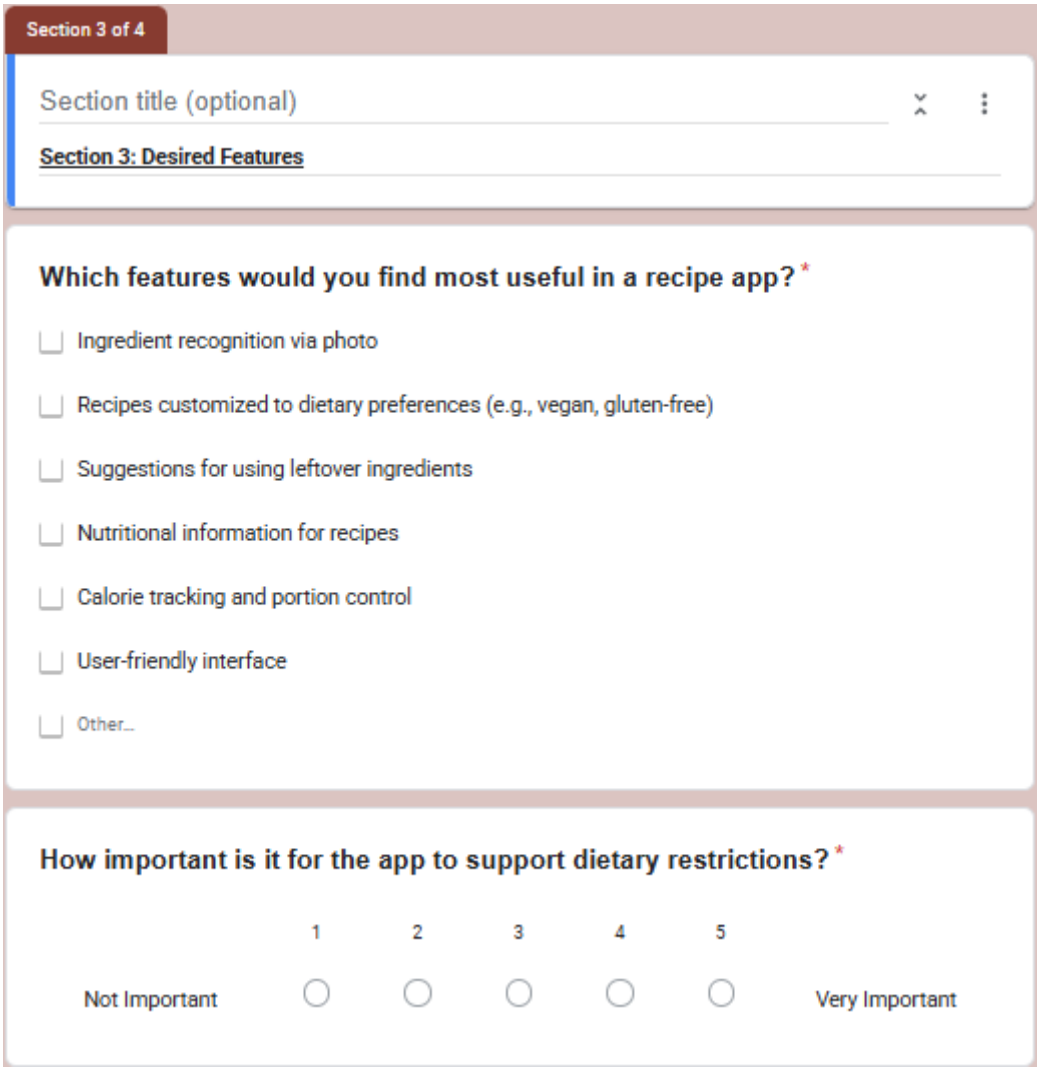
- [ ] Based on available ingredients
- [ ] Following recipes from the internet/books
- [ ] Spontaneously, without a plan
- [ ] Other...

What challenges do you face when planning meals? *

- [ ] Identifying what to cook with available ingredients
- [ ] Adjusting recipes to dietary restrictions/preferences
- [ ] Reducing food waste
- [ ] Time constraints for cooking
- [ ] Lack of inspiration/ideas
- [ ] Other...

Have you ever discarded food because it expired before you could use it? *

- ( ) Often
- ( ) Sometimes
- ( ) Rarely
- ( ) Never


Section 3 of 4
Section title (optional)
Section 3: Desired Features
Which features would you find most useful in a recipe app? *
Ingredient recognition via photo
Recipes customized to dietary preferences (e.g., vegan, gluten-free)
Suggestions for using leftover ingredients
Nutritional information for recipes
Calorie tracking and portion control
User-friendly interface
Other...
How important is it for the app to support dietary restrictions? *
1
2
3
4
5
Not Important
Very Important

Appendix C

User Demographic

What is your age

- ◯ Under 18
- ◯ 18–25
- ◯ 26–35
- ◯ 36–45
- ◯ 46 and above

Usability and Interface

How easy to register for an account in the mobile application? *

	1	2	3	4	5	
Very Difficult	◯	◯	◯	◯	◯	Very Easy

How easy to login for an account in the mobile application? *

	1	2	3	4	5	
Very Difficult	◯	◯	◯	◯	◯	Very Easy

How easy to use the function built in the mobile application? *

	1	2	3	4	5	
Very Difficult	◯	◯	◯	◯	◯	Very Easy

Functionality

Did the system detect accurately the actual food ingredients? *

- ◯ Always
- ◯ Most of the times
- ◯ Sometimes
- ◯ Rarely
- ◯ Never

Did the system generate the correct recipe according to the selected recipe type and ingredients? *

- ◯ Always
- ◯ Most of the times
- ◯ Sometimes
- ◯ Rarely
- ◯ Never

Did the ingredients can be saved/manually input and stored in the pantry properly? *

- ◯ Yes
- ◯ No

Did the generated can be saved and shared properly? *

- ◯ Yes
- ◯ No

Performance

How would rate the response time of the food ingredients detection? *

	1	2	3	4	5	
Very Slow	○	○	○	○	○	Very Fast

How would rate the generation time of the generation of food recipe? *

	1	2	3	4	5	
Very Slow	○	○	○	○	○	Very Fast

Feedback and Recommendation

Do you feel the app helped you reduce food waste or plan better meals? *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

How likely are you to continue using this app in the future? *

	1	2	3	4	5	
Very Unlikely	○	○	○	○	○	Very Likely

Would you recommend this app to others? *

- ○ Yes
- ○ No
- ○ Maybe