

**PERAK TOURISM SEARCH USING GOOGLE MAP ANDROID APP  
DEVELOPMENT  
BY  
WONG WEI YING**

**A REPORT  
SUBMITTED TO  
Universiti Tunku Abdul Rahman  
in partial fulfillment of the requirements  
for the degree of  
BACHELOR OF COMPUTER SCIENCE (HONOURS)  
Faculty of Information and Communication Technology  
(Kampar Campus)**

**FEBRUARY 2025**

## COPYRIGHT STATEMENT

© 2025 Wong Wei Ying. All rights reserved.

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

## **ACKNOWLEDGEMENTS**

I would like to express thanks and appreciation to my supervisor, Dr. Ooi Chek Yee, who has given me the opportunity to engage in this tourism search mobile application development. His guidance and and invaluable experience have encouraged me throughout my project. A million thanks to you.

I would like to express my gratefulness to my family for their love, and continuous support throughout the project.

# ABSTRACT

In the age of globalization, the Internet plays a vital role in people's lives. To stay connected to the Internet, various technologies, including mobile devices have undergone rapid development. Therefore, various mobile applications are developed to fulfil the growth of user's demand in various sectors, such as the tourism sector. Hence, this project will delve into the functionalities and features proposed by the existing systems and their limitations. Based on the analysis on the current applications, this project aims to develop a tourism search application to overcome the limitations by proposing innovative solutions. It seeks to enhance user experience by addressing issues like unnecessary features that burden the system, imbalance food priority and fake reviews. It will focus exclusively on tourism search to provide a simple, straightforward user interface for the user by eliminating non-essential features. It will include a categorization system for easy exploration and discovery of various cuisines options. Real identity verification via face detection and face recognition is implement to maintain trust and authenticity on our platform, therefore minimize the fake review. This project implements Agile Development for the whole development processes. This application is developed using Flutter with Dart programming language. Google Map API is integrated to display the route, calculate duration and total distance between user's location and destination. Firebase cloud-based data management server is integrated and its services, such as authentication and firestore are utilized. Supabase storage is integrated to manage the photos.

Area of Study (Minimum 1 and Maximum 2): **Mobile App Development**

Keywords (Minimum 5 and Maximum 10): **Tourism Search, User-friendly Application, Mobile Application, Flutter and Dart, Firestore Implementation**



# TABLE OF CONTENTS

<b>TITLE PAGE</b>	<b>i</b>
<b>COPYRIGHT STATEMENT</b>	<b>ii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>TABLE OF CONTENTS</b>	<b>v</b>
<b>LIST OF FIGURES</b>	<b>x</b>
<b>LIST OF TABLES</b>	<b>xvi</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xviii</b>

<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Problem Statement and Motivation	1
1.2 Objectives	2
1.3 Project Scope and Direction	3
1.4 Contributions	4
1.5 Report Organization	4

<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>5</b>
2.1 Existing Travel App	5
2.1.1 Trip Advisor	5
2.1.2 Traveloka	9
2.1.3 Visit a City	12
2.2 Limitations on Previous Application	13
2.3 Proposed Solution	15

<b>CHAPTER 3 SYSTEM REQUIREMENT &amp; DESIGN</b>	<b>16</b>
3.1 System Use Case Diagram	16
3.2 System Use Case Description	17
3.2.1 Sign Up	17
3.2.2 Login	18
3.2.3 Log out	19

3.2.4	Reset Password	20
3.2.5	Direct Message	20
3.2.6	View and Filter Reviews	21
3.2.7	View Map	22
3.2.8	Write Review	23
3.2.9	Edit Profile	24
3.2.10	Real-Time Search	24
3.2.11	Filter Meerchant	25
3.2.12	Add/ Remove Merchant from Savelist	26
3.2.13	Edit Business Detail	27
3.3	Activity Diagram	28
3.3.1	Sign Up	28
3.3.2	Login	29
3.3.3	Log out	30
3.3.4	Reset Password	30
3.3.5	Direct Message	31
3.3.6	View and Filter Reviews	31
3.3.7	View Map	32
3.3.8	Write Review	32
3.3.9	Edit Profile	33
3.3.10	Real-Time Search	34
3.3.11	Filter Meerchant	34
3.3.12	Add/ Remove Merchant from Savelist	35
3.3.13	Edit Business Detail	36
3.4	Sequence Diagram	37
3.4.1	Sign Up	37
3.4.2	Login	38
3.4.3	Log out	38
3.4.4	Reset Password	39
3.4.5	Direct Message	39
3.4.6	View and Filter Reviews	40
3.4.7	View Map	40
3.4.8	Write Review	41

3.4.9	Edit Profile	41
3.4.10	Real-Time Search	42
3.4.11	Filter Meerchant	42
3.4.12	Add/ Remove Merchant from Savelist	43
3.4.13	Edit Business Detail	44
3.5	Block Diagram	45
3.5.1	User Block Diagram	45
3.5.2	Merchant Block Diagram	47
<b>CHAPTER 4 METHODOLOGY, TOOLS AND SYSTEM SET UP</b>		<b>48</b>
4.1	Methodology	48
4.2	Tools to use	48
4.2.1	Hardware	48
4.2.2	Software	49
4.3	System Set Up	51
4.3.1	Flutter	51
4.3.2	Firebase	53
4.3.3	Supabase	59
4.3.4	Google Map API	62
4.3.5	Face Recognition API	63

<b>CHAPTER 5 IMPLEMENTATION</b>	<b>64</b>
5.1 Database Design	64
5.1.1 Firebase Authentication	64
5.1.2 Firebase Firestore	65
5.1.3 Supabase Database	72
5.2 User Interface Design	73
5.2.1 Login Page	73
5.2.2 Select Role Page	74
5.2.3 Disclaimer Page	75
5.2.4 Scan IC Page	76
5.2.5 Face Comparison Page	77
5.2.6 Sign Up Page	78
5.2.7 Merchant Business Detail Page	79
5.2.8 Reset Password Page	80
5.2.9 User Home Page	81
5.2.10 Savelist Page	82
5.2.11 Food List Page	83
5.2.12 Place List Page	85
5.2.13 Activity List Page	86
5.2.14 Food Detail Page	87
5.2.15 Activity Detail Page	88
5.2.16 Place Detail Page	89
5.2.17 View Map Page	90
5.2.18 Write Review Page	91
5.2.19 Review List Page	92
5.2.20 Chat Room Page	93
5.2.21 Chat List Page	94
5.2.22 Profile Page	95
5.2.23 Edit Profile Page	96
5.2.24 Edit Profile Form Page	97
5.2.25 Merchant Home Page	98
5.2.26 Merchant Setting Page	99

5.2.27	Edit Businss Detail Page	100
5.2.28	Edit Businss Detail Form Page	101
<b>CHAPTER 6</b>	<b>SYSTEM TESTING</b>	<b>102</b>
6.1	Test Cases	102
6.1.1	Sign Up Functionality	102
6.1.2	Login Functionality	104
6.1.3	Log out Functionality	105
6.1.4	Reset Password Functionality	106
6.1.5	Direct Message Functionality	107
6.1.6	View and Filter Reviews Functionality	108
6.1.7	View Map Functionality	109
6.1.8	Write Review Functionality	110
6.1.9	Edit Profile Functionality	111
6.1.10	Real-Time Search Functionality	113
6.1.11	Filter Meerchant Functionality	113
6.1.12	Add/ Remove Merchant from Savelist Functionality	117
6.1.13	Edit Business Detail Functionality	118
<b>CHAPTER 7</b>	<b>CONCLUSION</b>	<b>121</b>
7.1	Project Review	121
7.2	Implementation Issues and Challenges	121
7.3	Future Work	122
<b>REFERENCES</b>		<b>124</b>
<b>POSTER</b>		<b>126</b>

## LIST OF FIGURES

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 2.1.1.1	Example of Exploration in Tokyo	5
Figure 2.1.1.2	Example of Reservation Services	5
Figure 2.1.1.3	Example of Geolocation Services	5
Figure 2.1.1.4	Example of New York City Travel Forum	6
Figure 2.1.1.5	Example of Itinerary Planning Feature	6
Figure 2.1.1.6	Examples of AI Building A Trip	6
Figure 2.1.1.7	Example of “Add A Missing Place” Feature	7
Figure 2.1.1.8	Details of “Add A Missing Place” Feature	7
Figure 2.1.1.9	Example of “Improve This Listing” Feature	7
Figure 2.1.1.9	Example of User Review’s Achievement System	8
Figure 2.1.2.1	Example of Flight Schedule	9
Figure 2.1.2.2	Example of Hotel Recommendation	9
Figure 2.1.2.2	Example of Hotel Recommendation	9
Figure 2.1.2.4	Example of Exploration of a Country: Singapore	10
Figure 2.1.2.5	Example of Promotions Offered	10
Figure 2.1.2.6	Example of Review & Rating	11
Figure 2.1.3.1	Example of Countries for Travel	12
Figure 2.1.3.2	Example of Exploration of A Country: Rome	12
Figure 2.1.3.3	Example of Travel Tips for Rome	12
Figure 2.1.3.4	Example of Rating and Review	13
Figure 3.1	Overall Use Case Diagram	16
Figure 3.3.1	Sign Up Activity Diagram	28
Figure 3.3.2	Login Activity Diagram	29
Figure 3.3.3	Log out Activity Diagram	30
Figure 3.3.4	Reset Password Activity Diagram	30
Figure 3.3.5	Direct Message Activity Diagram	31
Figure 3.3.6	View and Filter Review Activity Diagram	31
Figure 3.3.7	View Map Activity Diagram	32

Figure 3.3.8	Write Review Activity Diagram	32
Figure 3.3.9	Edit Profile Activity Diagram	33
Figure 3.3.10	Real-time Search Merchant Activity Diagram	34
Figure 3.3.11	Filter Merchant Activity Diagram	34
Figure 3.3.12	Add/ Remove Merchant from Savelist Activity Diagram	35
Figure 3.3.13	Edit Business Detail Activity Diagram	36
Figure 3.4.1	Sign Up Sequence Diagram	37
Figure 3.4.2	Login Sequence Diagram	38
Figure 3.4.3	Reset Password Sequence Diagram	38
Figure 3.4.4	Log out Sequence Diagram	39
Figure 3.4.5	Direct Message Sequence Diagram	39
Figure 3.4.6	View and Filter Reviews Sequence Diagram	40
Figure 3.4.7	View Map Sequence Diagram	40
Figure 3.4.8	Write Review Sequence Diagram	41
Figure 3.4.9	Edit Profile Sequence Diagram	41
Figure 3.4.10	Real-time Search Merchants Sequence Diagram	42
Figure 3.4.11	Filter Merchants Sequence Diagram	42
Figure 3.4.12	Add/ Remove Merchants from Savelist Sequence Diagram	43
Figure 3.4.13	Edit Business Detail Sequence Diagram	44
Figure 3.5.1	User Block Diagram	45
Figure 3.5.2	Merchant Block Diagram	47
Figure 4.1	Agile Development Cycle [12]	48
Figure 4.3.1.1	Open command palette on Visual Studio Code	51
Figure 4.3.1.2	Create New Flutter Project	52
Figure 4.3.1.3	Create New Application	52
Figure 4.3.1.4	Select a folder where this application is located	52
Figure 4.3.1.5	Enter project name	52
Figure 4.3.1.6	Main.dart file is generated, confirming successful setup of the application	52
Figure 4.3.1.7	Set up user permissions on Android Manifest.xml	53
Figure 4.3.2.(A).1	Go to Firebase Console	53

Figure 4.3.2.(A).2	Create a Firebase Project	53
Figure 4.3.2.(A).3	Enter Project Name	54
Figure 4.3.2.(A).4	Enable Google Analytics	54
Figure 4.3.2.(A).5	Select firebase location	55
Figure 4.3.2.(A).6	Wait for the firebase to be prepared	55
Figure 4.3.2.(A).7	Firebase Dashboard displayed, confirming successful setup	55
Figure 4.3.2.(B).1	Set up Firebase into Flutter project using command prompt	56
Figure 4.3.2.(B).2	A firebase_option.dart is generated, confirming successful setup of firebase on application	56
Figure 4.3.2.(B).3	Add firebase dependencies on pubspec.yaml using flutter terminal	56
Figure 4.3.2.(B).4	Initialize Firebase on application	56
Figure 4.3.2.(C).1	Set up Firebase Authentication	57
Figure 4.3.2.(C).2	Select sign in method	57
Figure 4.3.2.(C).3	Save sign in method	57
Figure 4.3.2.(D).1	Set up Firestore	58
Figure 4.3.2.(D).2	Select Firestore location	58
Figure 4.3.2.(D).3	Select test mode for development	58
Figure 4.3.2.(D).4	Update security rules to allow only authenticated user to read and write the database	59
Figure 4.3.3.(A).1	Create new Supabase project	59
Figure 4.3.3.(B).1	Add Supabase dependency on Flutter	59
Figure 4.3.3.(B).2	Get Supabase anonKey	60
Figure 4.3.3.(B).2	Initialize Supabase on Flutter	60
Figure 4.3.3.(C).1	Create storage bucket to store photos	60
Figure 4.3.3.(C).2	Set up storage bucket's policies for CRUD operation	61
Figure 4.3.3.(C).3	Set up storage bucket's policies using provided customization option	61
Figure 4.3.3.(C).4	Write new storage bucket policy	61
Figure 4.3.3.(C).5	Policy updated	62
Figure 4.3.4.1	Enable geolocation api on google map platform	62



	console	
Figure 4.3.4.2	Get Google Map apiKey	62
Figure 4.3.4.3	Add Google Map apiKey in xml file	63
Figure 4.3.4.4	Add google map dependency in pubspec.yaml file	63
Figure 4.3.4.5	Sample Code to call API to use google map service	63
Figure 4.3.5.1	Get Face++ apiKey	63
Figure 4.3.5.2	Call API for face comparison using http with apiKey	63
Figure 5.1.1	Firebase Authentication Database	64
Figure 5.1.2	Overview of Firebase Firestore Database Design	65
Figure 5.1.2.1	UsersRole Collection	67
Figure 5.1.2.2	User Collection	67
Figure 5.1.2.3	Savelist Sub-Collection from User Collection	68
Figure 5.1.2.4	Merchant Collection	68
Figure 5.1.2.5	Places Collection	69
Figure 5.1.2.6	Food Collection	69
Figure 5.1.2.7	Activities Collection	70
Figure 5.1.2.8	Rating Subcollection from Food Collection	70
Figure 5.1.2.9	Messages Collection	71
Figure 5.1.2.10	Chats Collection	71
Figure 5.1.2.11	Chat rooms Collection	72
Figure 5.1.3	Supabase Stroage Bucket	72
Figure 5.2.1	Login Page	73
Figure 5.2.2	Select Role Page	74
Figure 5.2.3	Disclaimer Page	75
Figure 5.2.4.1	Scan IC Page	76
Figure 5.2.4.2	Upload IC options on the Scan IC Page	76
Figure 5.2.4.3	Successful IC Scan on the Scan IC Page	76
Figure 5.2.5.1	Analyze user's face on the Face Comparison Page	77
Figure 5.2.5.2	Verification fail on the Face Comparison Page	77
Figure 5.2.6.1	Sign Up Page for User	78
Figure 5.2.6.2	Sign Up Page for Merchant	78
Figure 5.2.7	Merchant Business Detail Page	79

Figure 5.2.8.1	Reset Password Page	80
Figure 5.2.8.2	Reset Password Link Sent Via email	80
Figure 5.2.8.3	Reset Password with Reset Password Link	80
Figure 5.2.9	User Home Page	81
Figure 5.2.10	Savelist Page	82
Figure 5.2.11.1	Food List Page	83
Figure 5.2.11.2	Search Feature on Food List Page	83
Figure 5.2.11.3	Filter drawer on Food List Page	84
Figure 5.2.11.4	Example of Filtered Food List on Food List Page	84
Figure 5.2.12.1	Place List Page	85
Figure 5.2.12.2	Search Feature on Place List Page	85
Figure 5.2.12.3	Filter drawer on Place List Page	85
Figure 5.2.13.1	Activity List Page	86
Figure 5.2.13.2	Filter drawer on Activity List Page	86
Figure 5.2.13.3	Example of Filtered Activity List on Activity List Page	86
Figure 5.2.14	Food Detail Page	87
Figure 5.2.15	Activity Detail Page	88
Figure 5.2.16	Place Detail Page	89
Figure 5.2.17.1	Ask user's permission on getting device's location on the View Map Page	90
Figure 5.2.17.2	User's permission is not granted on View Map Page	90
Figure 5.2.17.3	Travel Information by Car on the View Map Page	90
Figure 5.2.18.1	Write Review Page	91
Figure 5.2.18.2	Example of a Refreshed Review Record After a Review is Updated on the Previous Page	91
Figure 5.2.19.1	Review List Page	92
Figure 5.2.19.2	Filter feature on Review List Page	92
Figure 5.2.20.1	Chat Room Page (Sender)	93
Figure 5.2.20.2	Chat Room Page (Receiver)	93
Figure 5.2.21.1	Chat List Page (Sender from Figure 5.2.19.1)	94
Figure 5.2.21.2	Chat List Page (Receiver from Figure 5.2.19.2)	94
Figure 5.2.22.1	Profile Page	95
Figure 5.2.22.2	Log out on Profile Page	95

Figure 5.2.23.1	Edit Profile Page	96
Figure 5.2.23.2	Update Profile Picture on Edit Profile Page	96
Figure 5.2.24.1	Edit Profile Form Page	97
Figure 5.2.24.2	Example of updating username on Edit Profile Form Page	97
Figure 5.2.25.1	Merchant Home Page (Review Tab)	98
Figure 5.2.25.2	Merchant Home Page (Chat List Tab)	98
Figure 5.2.26	Merchant Setting Page	99
Figure 5.2.27	Edit Business Detail Page	100
Figure 5.2.28.1	Update text data on Edit Business Detail Form Page	101
Figure 5.2.28.2	Update address on Edit Business Detail Form Page	101
Figure 5.2.28.3	Update tags on Edit Business Detail Form Page	101

## LIST OF TABLES

<b>Table Number</b>	<b>Title</b>	<b>Page</b>
Table 3.2.1	Sign Up Use Case Description	17
Table 3.2.2	Login Use Case Description	18
Table 3.2.3	Log out Use Case Description	19
Table 3.2.4	Reset Password Use Case Description	20
Table 3.2.5	Direct Message Use Case Description	20
Table 3.2.6	View and Filter Reviews Use Case Description	21
Table 3.2.7	View Map Use Case Description	22
Table 3.2.8	Write Review Use Case Description	23
Table 3.2.9	Edit Profile Use Case Description	24
Table 3.2.10	Real-time Search Use Case Description	25
Table 3.2.11	Filter Merchant Use Case Description	25
Table 3.2.12	Add/Remove Merchant from Savelist Use Case Description	26
Table 3.2.13	Edit Business Detail Use Case Description	27
Table 4.2.1.1	Specifications of laptop	48
Table 4.2.1.2	Specifications of Android Mobile Device	49
Table 6.1.1	Sign Up functionality Test Cases	102
Table 6.1.2	Login functionality Test Cases	104
Table 6.1.3	Log out functionality Test Cases	105
Table 6.1.4	Reset Password functionality Test Cases	106
Table 6.1.5	Direct Message functionality Test Cases	107
Table 6.1.6	View and Filter Reviews functionality Test Cases	108
Table 6.1.7	View Map functionality Test Cases	109
Table 6.1.8	Write Review functionality Test Cases	110
Table 6.1.9	Edit Profile functionality Test Cases	111
Table 6.1.10	Real-time Search functionality Test Cases	113
Table 6.1.11	Filter Merchant functionality Test Cases	113
Table 6.1.12	Add/Remove Merchant from Savelist functionality	117

	Test Cases	
Table 6.1.13	Edit Business Detail functionality Test Cases	118

## **LIST OF ABBREVIATIONS**

API	Application Programming Interface
SDLC	Software Development Life Cycle
IC	Identity Card
ETA	Estimated Time of Arrival
SDK	Software Development Kit
URL	Uniform Resource Locator

### Chapter 1 Introduction

In the context of globalisation, the Internet plays a significant role in people's lives. More than half of the population around the world uses the Internet to perform daily tasks, which in turn drives the development and evolution of technology in various sectors, including the tourism sector [1]. The rapid growth and deployment of mobile technology has transformed the interaction approach between the business owner and customer [1]. Mobile applications for tourism search have profoundly altered the travel styles and trends by providing easy access to vast amounts of information about destinations, accommodations, transportations, foods, activities and reviews. With the increasing demand for tourism search applications, various features and functionalities are being developed. Hence, the traveler could use mobile applications to plan their trips extensively.

#### 1.1 Problem Statement and Motivation

##### i. Unnecessary Added Features and A Complex User Interface

In order to keep the user engaged and continue using the app, the app publishers tend to add more features to improve the user experience. However, when more features are added in, it will add complexity to the system, causing various errors or bugs to occur. If these bugs are not fixed, it will impact the user experience. It becomes even worse when the features are not organized and structured well on the user interface. In short, adding new features to engage the customer has been a common way or concept on the market, however, it is not a correct direction for all the applications.

##### ii. Imbalanced Food Priority

Current mobile applications for tourism search on the market often prioritize attractions, accommodations and transportation selections, but ignore the significance of culinary aspects for certain travellers. This contradicts the new

prevailing food trend among Gen Z and millennial, who look for the unique and authentic taste of local cuisine.

### **iii. Fake Review**

Review helps the customer to make decisions. However, review has become a new tactic to affect customer's selection. This deceptive practice involves business owners paying for the individual to write positive reviews or write negative reviews on other businesses. This unethical practice not only deceives the customer but also distorts the marketplace.

## **1.2 Objectives**

- To optimize the application by focusing exclusively on tourism search features. The user can explore food, activities in Perak, Malaysia. Remove non-essential features can improve system performance and leverage user experience.
- To implement a balanced priority system that ensures fair and accurate food recommendations based on user preferences. Categorization system will be implemented for personalized search results.
- To ensure credibility-driven reviews and minimize fake submissions, the user is required to scan their identification card (IC) and perform facial verification by comparing their live photo with the photo on the IC. This verification step ensures that only genuine user can post reviews, improving the reliability of feedback on the platform.



### 1.3 Project Scope and Direction

The scope of this project encompasses the development of a tourism search mobile application. This application will optimize the performance and leveraging user experience by focusing on tourism search features and functionalities and removing the non-essential features based on the analysis done on the current travel applications. The basic features includes Create, Read, Update and Delete (CRUD) operations on user authentication, account managements, search and filter of food and attractions in Perak, Malaysia.

In addition, this application will integrate a variety of categories and chip selections to efficiently present the food options, such as different ethnic cuisine options and food flavors. It helps to streamline the searching process by allowing the users to filter the content based on their preferences in a short time.

Moreover, this application will implement an identity verification system to prevent potential fake reviews. The user is required to scan their identification card (IC) and perform facial recognition, where their live image will be compared with the photo on the IC. This verification process ensures that only authenticated user can submit reviews, enhancing the trustworthiness of the content and reducing the likelihood of fake or malicious feedback.

Last but not least, this application is limited to the Android platform. It will integrate with the open source cloud data management platform, which is Firebase, to utilize its services, such as authentication for user verification, and firestore for data management. Supabase is integrated to handle photo management because of its cost-effectiveness.. Google Maps API is integrated to calculate the route path, and route duration in different transportation.

### **1.4 Contributions**

This project aims to optimize the application by focusing on the essential features and the simplicity of user interface, to ensure a smooth user experience. The categorization system that included food diversities, helps to explore and discover various culinary choices within the region in Perak, Malaysia. The implementation of facial recognition helps to identify potential fake reviews while allowing genuine feedback and data sharing.

### **1.5 Report Organization**

This report is organized into six chapters to provide a structured and comprehensive overview of the project. Chapter 1, Introduction, outlines the background of the project, the problem statement, objectives, scope, and the contribution of the system being developed. Chapter 2, Literature Review, presents an analysis of existing research. It highlights the strengths and limitations of existing application. Chapter 3, System Requirement and Design, details both the functional and non-functional requirements of the system. It also includes various design diagrams such as use case diagrams and descriptions, activity diagrams, and sequence diagrams to illustrate the system's structure and behavior. Chapter 4, System Implementation, describes the development process, including the database design and the implementation of the user interface. Chapter 5, System Testing, explains the testing process carried out to ensure the reliability and correctness of the system, including test cases and results. Finally, Chapter 6, Conclusion, summarizes the achievements of the project, discusses implementation issues and challenges encountered, and suggests possible future enhancements.

### Chapter 2 Literature Review

#### 2.1 Existing Travel App

##### 2.1.1 TripAdvisor

TripAdvisor offers various features to support trip planning such as accommodations, culinary and activities recommendations as well as booking services. It supports nearby searching by implementing geolocation services and establishing forums, covering various cities and thousands of topics for the user to explore. It provides itinerary tools for the user to arrange the itinerary or the user can let the AI build the trip by filtering the user request, such as the time spent for travelling, going alone or with a partner, sightseeing or food exploration and so forth. However, when more features are added in, the system becomes more complicated, and errors occur. Therefore, TripAdvisor has regularly released new versions to enhance the feature.

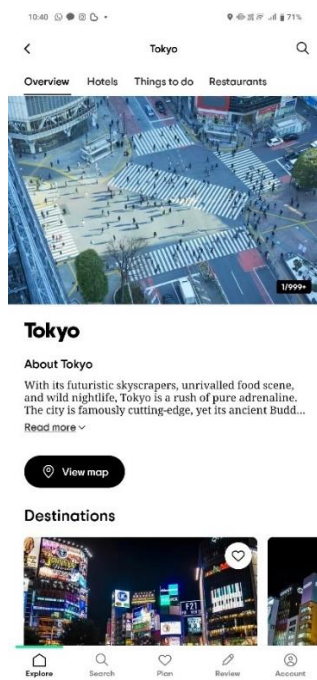


Figure 2.1.1.1 Example of Exploration in Tokyo



Figure 2.1.1.2 Example of Reservation Services

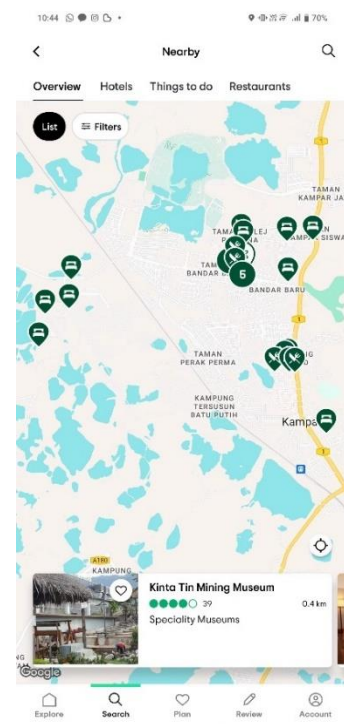


Figure 2.1.1.3 Example of Geolocation Services

## CHAPTER 2 LITERATURE REVIEW

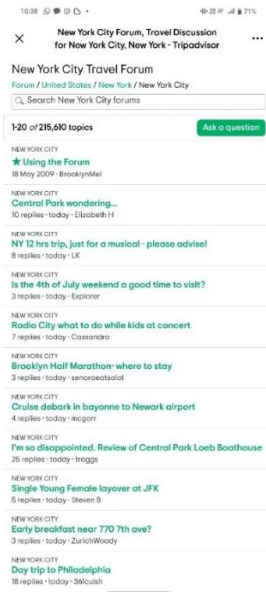


Figure 2.1.1.4 Example of New York City Travel Forum

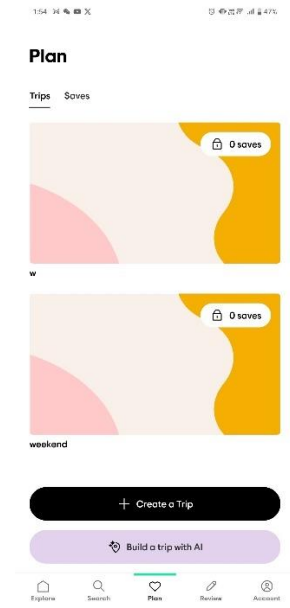


Figure 2.1.1.5 Example of Itinerary Planning Feature

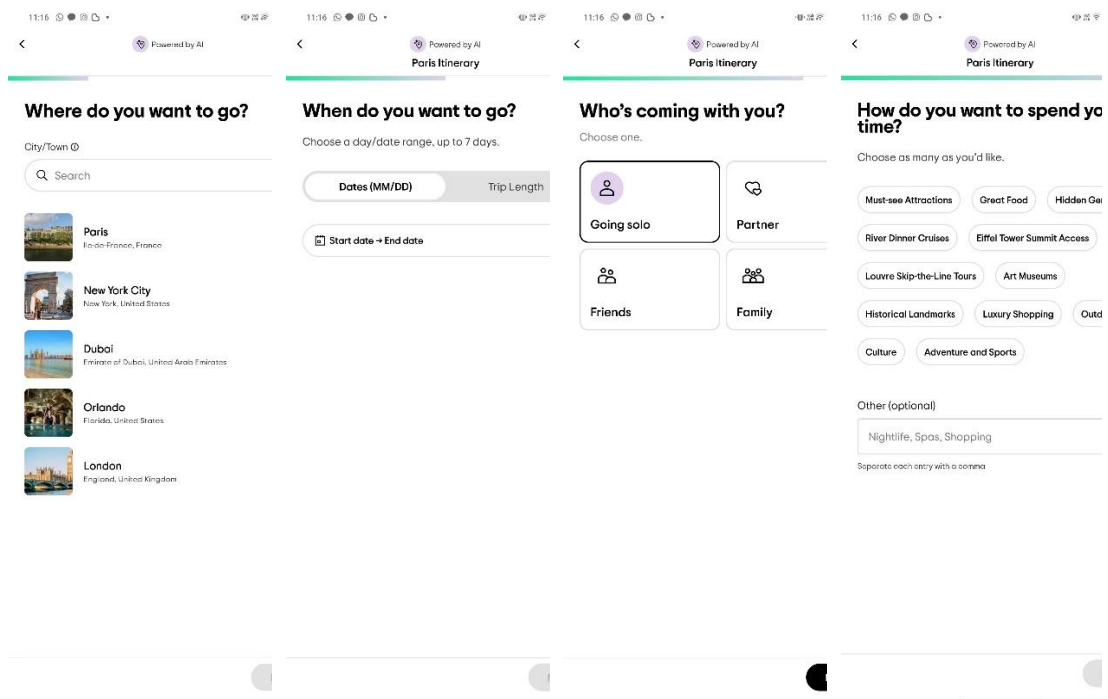


Figure 2.1.1.6 Examples of AI Building A Trip

## CHAPTER 2 LITERATURE REVIEW

TripAdvisor includes culinary experience for the user. Nevertheless, the selections of TripAdvisor are always the least compared to other features. To solve these problems, a new feature called “Add A Missing Place” is introduced. It allows the business owner to add in their business or user to add in the hidden gem on the map. An “Improve This Listing” option is provided on the bottom of the business information for the user to update the current status of the business, in order to ensure all the information is up to date.

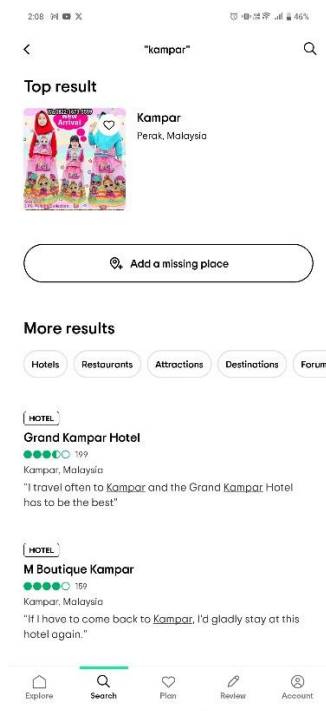


Figure 2.1.1.7 Example of  
“Add A Missing Place”  
Feature

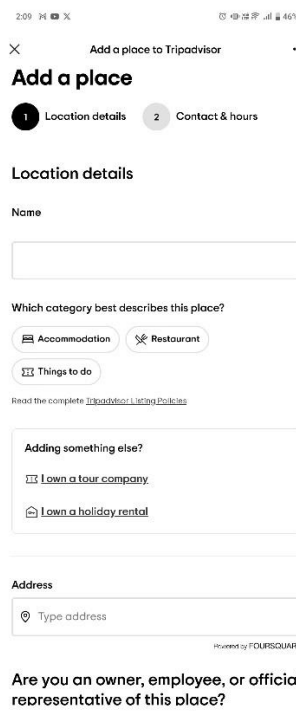


Figure 2.1.1.8 Details of  
“Add A Missing Place”  
Feature

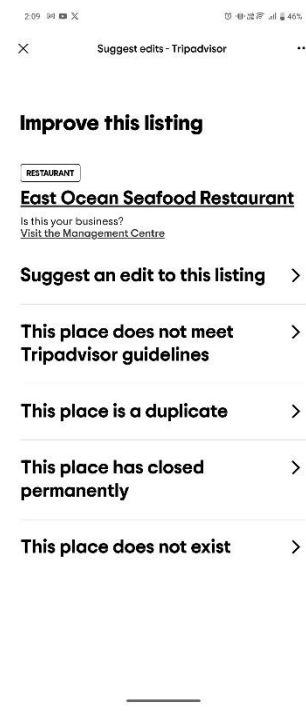


Figure 2.1.1.9 Exampe  
of “Improve This  
Listing” Feature

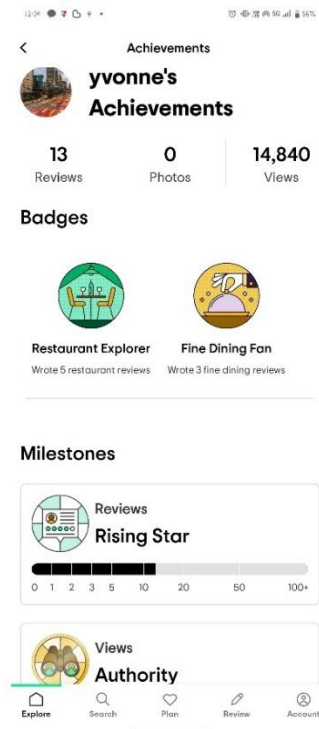


Figure 2.1.1.9 Exampe of User Review’s Achievement System

In addition, TripAdvisor introduces an user achievement system to track the amount of reviews posted on the platform. The user will be awarded badges such as “Restaurant Explorer”, “Review Rising Star” based on the number of reviews. As the user build theirs reputation through the uploaded reviews, it helps the user to differentiate between credible reviewers and those who may be posting fake or low quality reviews.

### Strengths:

Tripadvisor continually releases new versions to address the errors or bugs. At the same time, it maintains the available functionalities and features, allowing the user to effortlessly navigate, explore and plan the itineraries.

“Add A Missing Place” feature could encourage more discovery of culinary on the map by allowing the user to upload the hidden gems, but not restricting to the business owner only. “Improve This Listing” feature also ensures the latest business’s information.

## CHAPTER 2 LITERATURE REVIEW

TripAdvisor implements user achievement system to encourage user to share the feedback and maintain the credibility of the user.

### 2.1.2 Traveloka

Traveloka offers even more features as compared to TripAdvisor, since it covers not only accommodation, culinary and activities recommendation, but it also includes various transportation services such as flight booking and car rental service. It even grabs every chance to make profit such as selling travel essentials, tickets or vouchers for places in searchable countries. It is also surprising to find that the listed places are not only categorized into restaurants, activities and attractions, but also included playground, spa and relaxation and sport recreation. To remain competitive on the market, it offers various promotions and implements a point membership system to earn points from every payment made in the application. In other words, Traveloka intends to build an application on letting the user to fully utilize the application, from the beginning of flight and hotel selections to the detailed planning of destinations. Therefore, Traveloka has a tough time ensuring all functionalities and features are well behaved by regularly releasing new versions to address and fix the bugs.

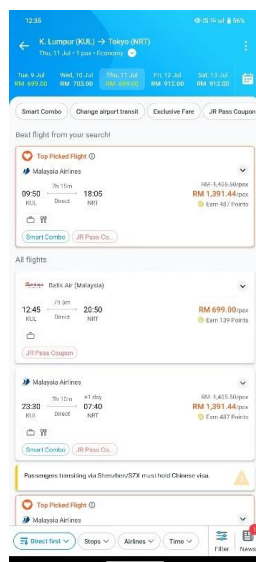


Figure 2.1.2.1 Example  
of Flight Schedule



Figure 2.1.2.2 Example  
of Hotel  
Recommendation

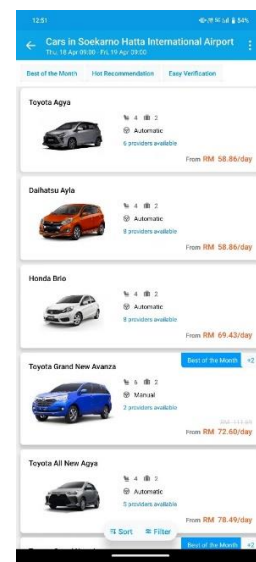


Figure 2.1.2.3 Example of  
Car Rental Service



## CHAPTER 2 LITERATURE REVIEW

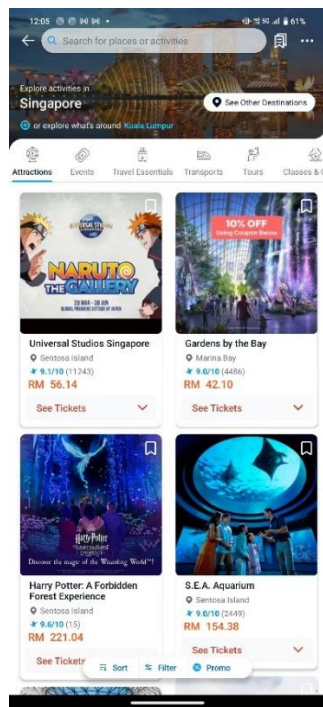


Figure 2.1.2.4 Example of Exploration of a Country:  
Singapore

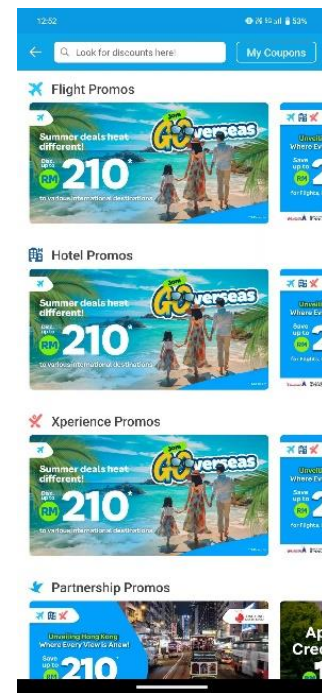
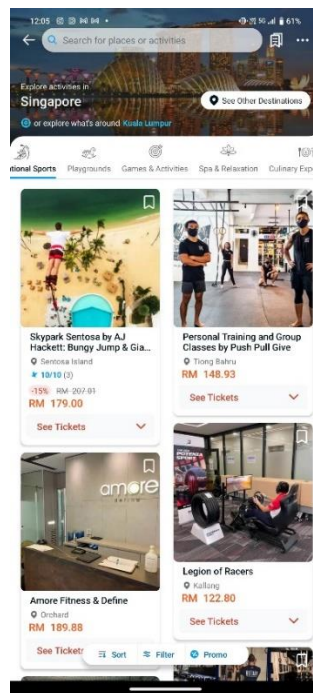


Figure 2.1.2.5 Example  
of Promotions Offered

Traveloka includes culinary experience for the user. The selections of TripAdvisor are always the least compared to other features. Nonetheless, there is no proposed solution on increasing the restaurant numbers currently. Traveloka includes rating and review features. However, only the customer who makes purchases can leave a review.



## CHAPTER 2 LITERATURE REVIEW

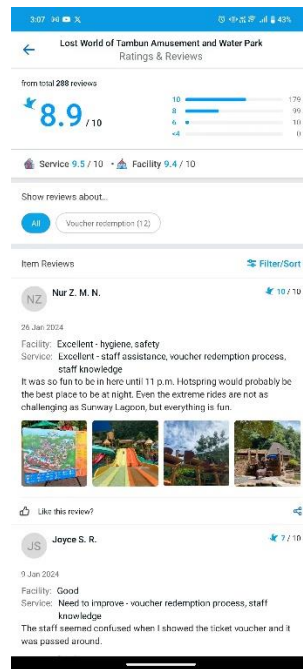


Figure 2.1.2.6 Example of Review & Rating

### Strengths:

Traveloka will constantly release new versions to fix the errors while maintaining the comprehensive functionalities and features. It empowers the user to arrange the itinerary and fulfil all the travel needs, from the beginning until the end of the journey. As a result, it streamlines the planning process, saving user time.

Traveloka only allows the the customer who have completed the payment to leave a comment. It adds credibility to comments.

## CHAPTER 2 LITERATURE REVIEW

### 2.1.3 Visit a City

Visit A City has a more simplified interface compared to TripAdvisor and Traveloka, since it purely works on promoting destinations such as landmarks and attractions of a country. Travel tips will be included in some popular cities. No culinary exploration is included in this application.

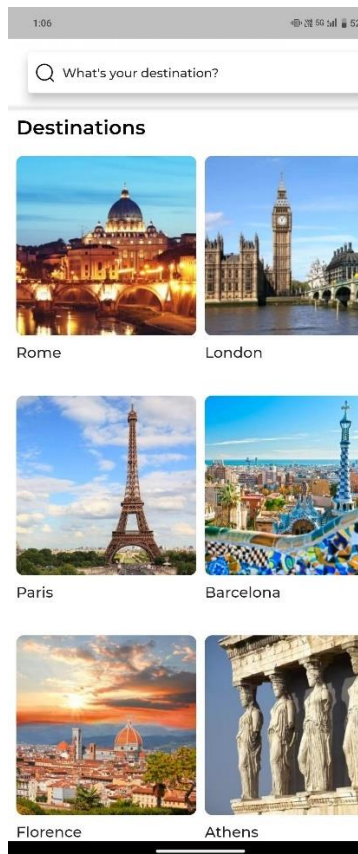


Figure 2.1.3.1 Example of Countries for Travel

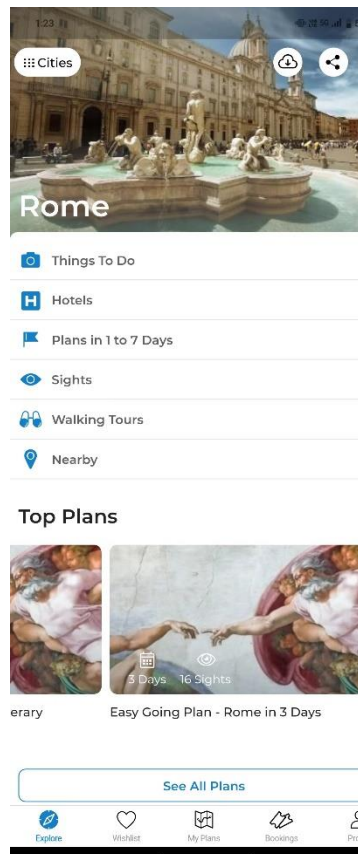


Figure 2.1.3.2 Example of Exploration of A Country: Rome

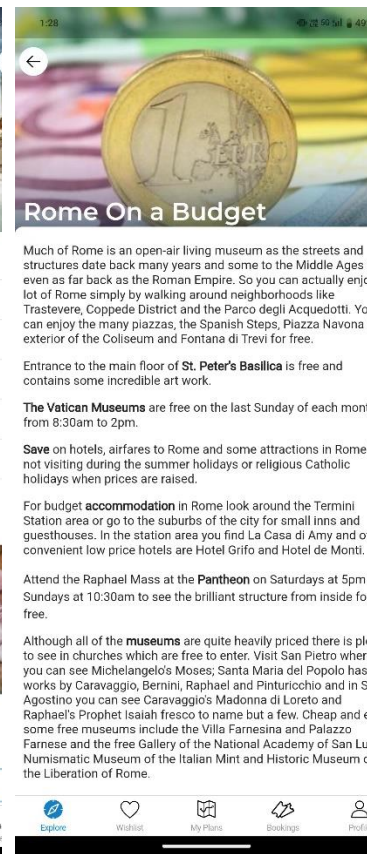


Figure 2.1.3.3 Example of Travel Tips for Rome

Visit A City includes rating and review features. Nevertheless, there is no associated actions to handle the fake reviews. It relies more on user reports to manage the fake review.



Figure 2.1.3.4 Example of Rating and Review

### Strength:

Visit A City has a user-friendly interface even with limited functionalities and features.

## 2.2 Limitations on Previous Applications

TripAdvisor and Traveloka indeed introduce a lot of features to support trip planning. The new release version of the application still could not fix these issues. Based on [8] and [9], the user was forced to quit the application. Some features on the application will malfunction suddenly, such as the user could not save their itinerary, could not retrieve the booking details in the application or could not upload the photo on the platform [8] [9].

Furthermore, both TripAdvisor and Traveloka incorporate a booking feature. According to [8], TripAdvisor will not help the customer to communicate or handle the booking management. The customer has to bear the loss if the payment for booking is not admitted. Nonetheless, Traveloka is much better than TripAdvisor, but the customer service is very slow. Both applications try to add more functions to keep the user

## CHAPTER 2 LITERATURE REVIEW

engaged with the platform, but they could not follow up immediately when the problem arose.

In addition, both TripAdvisor and Traveloka will inject several advertisements in every pages, which may upset the user. For Traveloka, there is no obvious searching icon on the main page, causing the first-time user to try to click on every icon. The main page is crowded with various titles, which may overwhelm the user.

All these three applications face the same challenge where the information of the places are wrong or out-dated. For instance, the user encountered a situation where they planned the places to go, but the shop was permanently closed a long time ago [8]. This will ruin the sense of trust in customers, in turn neglecting the application. For Visit A City, there is a limited listing of destinations. The user may find it hard to plan a whole trip by referencing this application.

Although the feature “Missing A Place” is very useful, the restaurant’s options are still limited. For instance, there are only a few selections when the user searches for “Ipoh” in restaurants. The displayed restaurants are all well-known and there is some hidden unique local food that is not being shared on the application. This is due to the scope and targeted audience of TripAdvisor are global tourists. Therefore, TripAdvisor is not user-friendly for Malaysians or the residents of Ipoh to explore local food.

When it comes to Traveloka, no restaurant is displayed for the Ipoh area. The business owner needs to deal with the partnership to register a business account. Additional costs or fees such as commission fees on bookings or marketing expenses will be charged. These costs may outweigh the potential advantages of registering a business account. Similarly, the business owner could choose to promote the business on costs-free social media platforms.

### 2.3 Proposed Solution

This project seeks to develop a tourism search application that offers solutions to the limitation addressing on the existing applications. In response to the limitations identified, the main reason for bugs occurring and limited support in handling booking management highlighted previously, is due to the overwhelming number of additional functionalities and the publisher fails to follow up when problems arise. For instance, the user could have better user experience and service to book a flight in airline-specific applications compared to Traveloka. So, all these additional functionalities already distract the main core of the tourism search to find suggested attractions, activities or culinary within a city, state or country. It becomes even worse when errors occur and the developers could not solve it, causing the overall user experience to be downgraded

Increasing additional features is not the only way to engage the user. Therefore, this project will neglect all the additional features by focusing only on the basic functions of tourism search, which is to display, search and filter the attractions, activities and cuisine based on the rating given by the users or various tags provided. No booking and payment will be included for this project. However, the user could add the intended visited place to the wishlist to review later. One of the advantages of removing the additional features is less memory space is needed to download the application and can speed up the performance.

To enhance the priority in culinary options, categorization system is introduced. Various tags, such as food type, food flavor and ethnic options are displayed. This helps the people to explore and discover the food diversity easily and filter the result based on the user preferences in a very short time.

To ensure the authenticity of reviews and reduce the risk of fake submissions, the system requires users to verify their identity by scanning their IC and performing facial recognition. The live image of the user will be compared with the photo on the IC to confirm their identity. This verification process ensures that only genuine and traceable users are allowed to post reviews, thereby increasing the credibility and reliability of the feedback shared on the platform.

## Chapter 3 System Requirements and Design

### 3.1 System Use Case Diagram

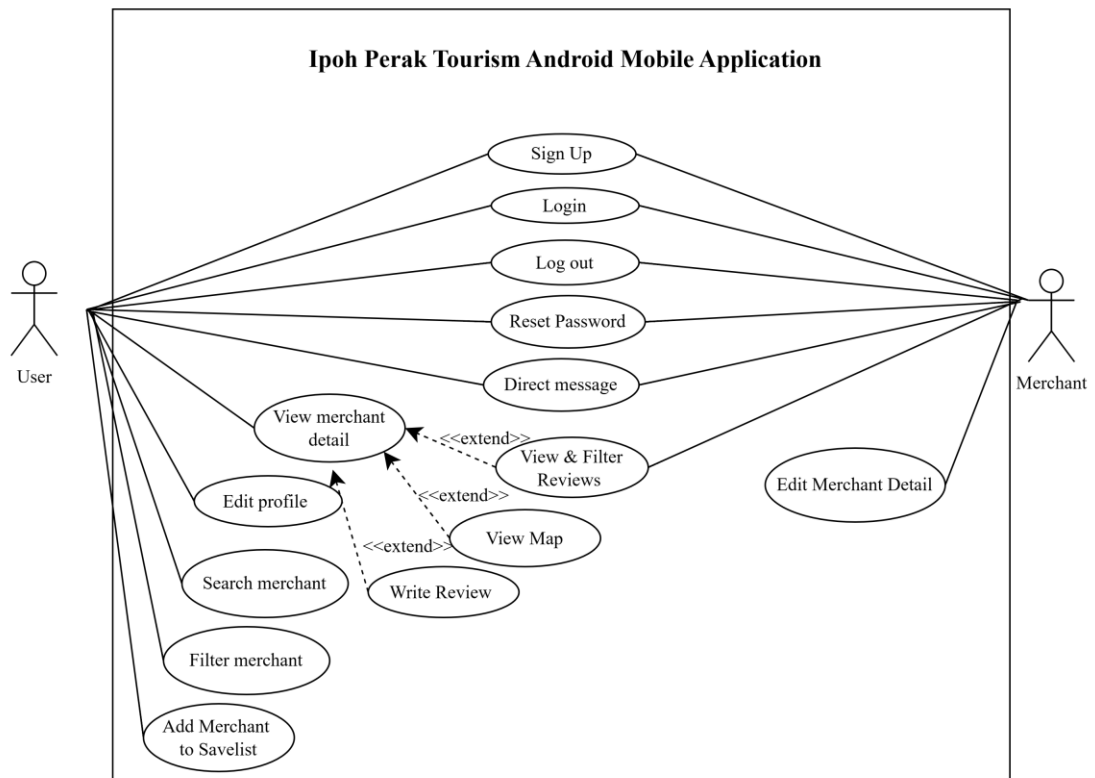


Figure 3.1 Overall Use Case Diagram

### 3.2 Use Case Description

#### 3.2.1 Sign Up

<b>Use Case ID</b>	UC001		<b>Version</b>	1.0
<b>Use Case</b>	Sign Up			
<b>Purpose</b>	To authenticate user before using the system.			
<b>Actor</b>	User, Merchant			
<b>Trigger</b>	User/ Merchant clicks sign up button on Login Page.			
<b>Precondition</b>	User/ Merchant has not created account using their email and ic number.			
<b>Scenario Name</b>	<b>Step</b>	<b>Action</b>		
<b>Main Flow</b>	1	User/ Merchant click sign up button on Login Page.		
	2.	System directs user/ merchant to Select Role Page.		
	3.	User/ Merchant selects role.		
	4.	System directs user/merchant to Scan IC Page.		
	5	User/ Merchant uploads IC from gallery or camera.		
	6.	System retrieves IC number, full name and user photo.		
	7.	System directs user/merchant to Face Capture Page.		
	8.	System opens live camera and verifies if face matches with IC photo by calling API		
	9.	System directs users/ merchants to the Sign Up Page.		
	10.	User fills up account details and click Sign Up button		
	11.	System checks if same email address exists on Firebase Authentication.		
	12.	System adds new user on Firebase Authentication and Firestore.		
	13.	System displays success message and direct user to Login page.		
<b>Alternate Flow:</b>	11.1	System displays error message.		
	11.2	System remains on Sign Up Page.		

<b>User exists</b>		
<b>Alternate</b>	10.1	Merchant fills up account details and click Continue button
<b>Flow:</b>	10.2	System directs merchant to Merchant Detail Page
<b>Role =</b>	10.3	Merchant fills up the business detail.
<b>Merchant</b>	10.4	Back to Main Flow Step 11
	12.1	System adds new user on Firebase Authentication and Firestore.
	12.2	System uploads business photos onto Supabase storage, stores photo urls on firestore
	12.3	Back to Main Flow Step 13

Table 3.2.1 Sign Up Use Case Description

### 3.2.2 Login

<b>Use Case ID</b>	UC002		<b>Version</b>	1.0
<b>Use Case</b>	Login			
<b>Purpose</b>	To authenticate user before using the system.			
<b>Actor</b>	User, Merchant			
<b>Trigger</b>	User/ Merchant opens the app.			
<b>Precondition</b>	User/ Merchant must have an existing account. User/ Merchant must know their correct login credentials. User/ Merchant must currently be logged out.			
<b>Scenario Name</b>	<b>Step</b>	<b>Action</b>		
<b>Main Flow</b>	1	User/ Merchant opens the app.		
	2	System checks if user info is available in local storage.		
	3.	If user info is unavailable, system directs user/ merchant to Login Page.		
	4.	User/ Merchant enters valid username and password		
	5.	User/ Merchant clicks the login button.		
	6.	System verifies the provided credentials.		
	7	System checks the role of the actor on Firestore		



		If user: Redirect to User Home Page If merchant: Redirect to Merchant Home Page
	8.	Systems stores user ID and user role into Local Storage.
	9.	System successfully logs in the user/ merchant and redirects them to the respective home page.
<b>Alternate Flow – User info is available</b>	2.1	Continue to Main Flow Step 7.
<b>Alternate Flow – Invalid Credentials</b>	3.1	User/ Merchant enters invalid username or password.
	3.2	System displays an error message.
	3.3	Back to Main Flow Step 2.

Table 3.2.2 Login Use Case Description

### 3.2.3 Log out

<b>Use Case ID</b>	UC003		<b>Version</b>	1.0
<b>Use Case</b>	Log out			
<b>Purpose</b>	To exit the app when not in use.			
<b>Actor</b>	User, Merchant			
<b>Trigger</b>	User/ Merchant clicks log out on Account Page.			
<b>Precondition</b>	User/ Merchant must be logged into their account.			
<b>Scenario Name</b>	<b>Step</b>	<b>Action</b>		
<b>Main Flow</b>	1	User/ Merchant clicks log out button.		
	2.	System pops up confirmation dialog to confirm log out.		
	3.	System clears user info on Local Storage, closes the dialog and directs user to the Login Page.		
<b>Alternate Flow: User/ Merchant</b>	2.1	System closes the dialog.		

<b>clicks No on confirmation dialog</b>		
---	--	--

Table 3.2.3 Log out Use Case Description

### 3.2.4 Reset Password

<b>Use Case ID</b>	UC004		<b>Version</b>	1.0
<b>Use Case</b>	Reset Password			
<b>Purpose</b>	To retrieve account when users forgot the password			
<b>Actor</b>	User, Merchant			
<b>Trigger</b>	User/ Merchant clicks forgot password on Login Page.			
<b>Precondition</b>	User/ Merchant must have an existing account.			
<b>Scenario Name</b>	<b>Step</b>	<b>Action</b>		
<b>Main Flow</b>	1	User/ Merchant clicks forgot password button on Login Page.		
	2.	System directs user/merchant to Reset Password Page.		
	3.	User/ Merchant enters email.		
	4.	System sends password reset link via gmail.		
	5	User/ Merchant enters new password via link		
	6.	System updates new password on Firebase Authentication		

Table 3.2.4 Reset Password Use Case Description

### 3.2.5 Direct Message

<b>Use Case ID</b>	UC005		<b>Version</b>	1.0
<b>Use Case</b>	Direct Message			
<b>Purpose</b>	To let user communicate with each other or with merchants			
<b>Actor</b>	User, Merchant			
<b>Trigger</b>	User clicks other users' profile image.			

	User clicks ‘Message me here’ on Merchant Detail Page. User enters chat room on Chat List Page.	
<b>Precondition</b>	User/ Merchant must be logged into their account.	
<b>Scenario Name</b>	<b>Step</b>	<b>Action</b>
<b>Main Flow</b>	1	User clicks to start a conversation.
	2.	System checks if a chat room exists between the user and the receiver.
	3.	If exists, system enters chat room and retrieves all the chat history and updates the number of unread messages.
	4.	User sends a message.
	5.	System stores new message into Firestore and update number of unread message of the receiver
	6.	On receiver side, system listens to Firestore change and push new message and displays updated chat in real time.
<b>Alternate Flow: Chat room not exist</b>	3.1	System creates new chat room.
	3.2	Back to Main Flow Step 4.

Table 3.3.5 Direct Message Use Case Description

### 3.2.6 View and Filter Reviews

<b>Use Case ID</b>	UC006	<b>Version</b>	1.0
<b>Use Case</b>	View and Filter Reviews		
<b>Purpose</b>	To let user views others’ reviews		
<b>Actor</b>	User		
<b>Trigger</b>	User clicks view all button on Merchant Detail Page. Merchant swipe to Review Tab on Merchant Home Page.		
<b>Precondition</b>	User/ Merchant must be logged into their account.		
<b>Scenario Name</b>	<b>Step</b>	<b>Action</b>	

<b>Main Flow</b>	1	User clicks view all button
	2.	System directs users to Review List Page
	3.	System displays all reviews in descending order of timestamp.
	4.	User filters the reviews by rating stars
	5.	System shows the filtered result.

Table 3.2.6 View and Filter Reviews Use Case Description

### 3.2.7 View Map

<b>Use Case ID</b>	UC007		<b>Version</b>	1.0
<b>Use Case</b>	View Map			
<b>Purpose</b>	To let user views the route.			
<b>Actor</b>	User			
<b>Trigger</b>	User clicks map on the Merchant Detail Page.			
<b>Precondition</b>	User must be logged into their account.			
<b>Scenario Name</b>	<b>Step</b>	<b>Action</b>		
<b>Main Flow</b>	1	User clicks map on the Merchant Detail Page		
	2.	System directs users to View Map Page		
	3.	System calls API to calculate route from the user's location to the destination, along with the estimated travel time by car and the total distance in kilometers.		
	4.	Systems displays the returned information.		
	5.	User chooses different transportation mode.		
	6.	System updates the route, travel time and total distance accordingly.		

Table 3.2.7 View Map Use Case Description

## 3.2.8 Write Review

<b>Use Case ID</b>	UC008		<b>Version</b>	1.0
<b>Use Case</b>	Write Review			
<b>Purpose</b>	To let user leaves a feedback for reference.			
<b>Actor</b>	User			
<b>Trigger</b>	User clicks write review button on Merchant Detail Page or on Review List Page.			
<b>Precondition</b>	User must be logged into their account.			
<b>Scenario Name</b>	<b>Step</b>	<b>Action</b>		
<b>Main Flow</b>	1	User clicks write review button.		
	2.	System directs users to Write Review Page		
	3.	User provides a star rating, write a comment of more than 10 characters, and optionally upload photos from their gallery or camera.		
	3.	System checks if a star rating is provided and the number of characters entered on comment section.		
	4.	System adds photos into supabase and add new review into the firestore.		
	5.	System recalculates the average rating and increase the rating count by one.		
	6.	System shows success message and direct users back to previous page with refreshed review records.		
<b>Alternate Flow: When star rating is empty or number of characters &lt; 10</b>	3.1	System displays error message		
	3.2	Back to Main flow Step 3.		

Table 3.3.8 Write Review Use Case Description

**3.2.9 Edit Profile**

<b>Use Case ID</b>	UC009		<b>Version</b>	1.0
<b>Use Case</b>	Edit Profile			
<b>Purpose</b>	To let user customizes profile.			
<b>Actor</b>	User			
<b>Trigger</b>	User click Edit Profile option on Profile Page.			
<b>Precondition</b>	User must be logged into their account.			
<b>Scenario Name</b>	<b>Step</b>	<b>Action</b>		
<b>Main Flow</b>	1	User clicks Edit Profile option on Profile Page.		
	2.	System directs user to the Edit Profile Page.		
	3.	System retrieves all user info from Firestore.		
	4.	User selects one user text-form info to be updated.		
	5	System passes the data to Edit Profile Form.		
	6.	User enters new info.		
	7.	System updates new info into firestore.		
	8.	System displays success message and directs user back to Edit Profile Page continue editing another user info.		
<b>Alternate Flow – User update profile image</b>	4.1	User updates profile image		
	4.2	System pops up bottom sheet for user to select image source.		
	4.3	User uploads photo via camera or gallery.		
	4.4	System overwrites image on supabase storage and update the image url on firestore.		
	4.5	System displays success message and refresh to display new profile picture.		

Table 3.2.9 Edit Profile Use Case Description

**3.2.10 Real-time Search**

<b>Use Case ID</b>	UC010	<b>Version</b>	1.0
--------------------	-------	----------------	-----

<b>Use Case</b>	Search Merchant	
<b>Purpose</b>	To let user search intended merchant easily	
<b>Actor</b>	User	
<b>Trigger</b>	User clicks search icon on Home Page.	
<b>Precondition</b>	User must be logged into their account.	
<b>Scenario Name</b>	<b>Step</b>	<b>Action</b>
<b>Main Flow</b>	1	User clicks search icon on Home Page.
	2.	System directs user to the Search Page.
	3.	User enters keywords.
	4.	System starts searching as the user types and update search result dynamically.

Table 3.2.10 Real-time Search Use Case Description

### 3.2.11 Filter Merchant

<b>Use Case ID</b>	UC011	<b>Version</b>	1.0
<b>Use Case</b>	Filter Merchant		
<b>Purpose</b>	To let user filters the list easily.		
<b>Actor</b>	User		
<b>Trigger</b>	User opens filter drawer on Merchant List Page.		
<b>Precondition</b>	User must be logged into their account.		
<b>Scenario Name</b>	<b>Step</b>	<b>Action</b>	
<b>Main Flow</b>	1	User opens filter drawer on Merchant List Page.	
	2	System pops filter drawer.	
	3.	User can filter by rating stars, multiple business tags, sort the results by review count or rating stars, or apply any combination of these filters	
	4.	System filters the list according to the filter options.	

Table 3.2.11 Filter Merchant Use Case Description

**3.2.12 Add/Remove Merchant from Savelist**

<b>Use Case ID</b>	UC012		<b>Version</b>	1.0
<b>Use Case</b>	Add/Remove Merchant from Savelist			
<b>Purpose</b>	To let user adds the intended visited place into savelist			
<b>Actor</b>	User			
<b>Trigger</b>	User clicks favourite icon on Merchant List/ Merchant Detail Page			
<b>Precondition</b>	User must be logged into their account.			
<b>Scenario Name</b>	<b>Step</b>	<b>Action</b>		
<b>Main Flow</b>	1	System checks if the places already exist in the user's save list in Firestore.		
	2.	System displays the favorite icon: red if already added, or grey if not.		
	3.	User clicks on the favourite icon.		
	4.	System updates Firestore: if the place was already saved, it is removed; if not, it is added.		
	5.	System updates the favorite icon color immediately to reflect the change.		
	6.	User navigates to the Save List Page to view all saved places.		
	7.	User can click the favourite icon to remove a place from the save list.		

Table 3.2.12 Add/Remove Merchant from Savelist Use Case Description



**3.2.13 Edit Business Detail**

<b>Use Case ID</b>	UC013	<b>Version</b>	1.0
<b>Use Case</b>	Edit Business Detail		
<b>Purpose</b>	To let merchant updates business detail.		
<b>Actor</b>	Merchant		
<b>Trigger</b>	Merchant clicks edit business detail option on Setting Page		
<b>Precondition</b>	Merchant must be logged into their account.		
<b>Scenario Name</b>	<b>Step</b>	<b>Action</b>	
<b>Main Flow</b>	1	Merchant clicks edit business detail option on Account Page.	
	2.	System directs user to the Edit Business Page.	
	3.	System retrieves all business info from Firestore.	
	4.	Merchant selects one user text-form info to be updated.	
	5	System passes the data to Edit Form.	
	6.	User enters new info.	
	7.	System updates new info into firestore.	
	8.	System displays success message and directs user back to Edit Profile Page continue editing another user info.	
<b>Alternate Flow – Merchant updates image</b>	4.1	User updates image	
	4.2	System pop up bottom sheet for user to select image source.	
	4.3	User uploads photo via camera or gallery.	
	4.4	System overwrites image on supabase storage and update the image url on firestore.	
	4.5	System displays success message and refresh to display new photos	

Table 3.2.13 Edit Business Detail Use Case Description

### 3.3 General Activity Diagram

#### 3.3.1 Sign Up

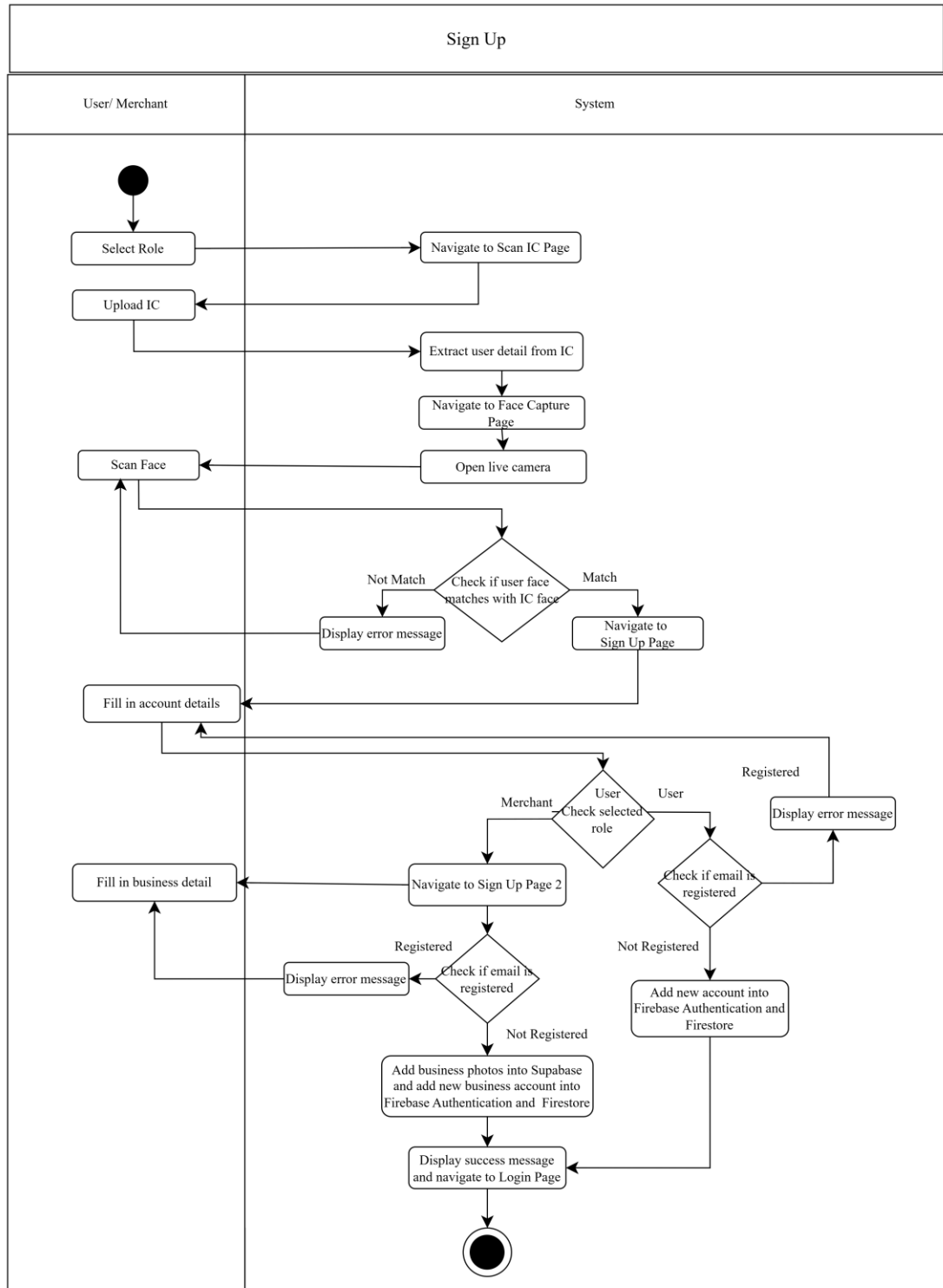


Figure 3.3.1 Sign Up Activity Diagram

### 3.3.2 Login

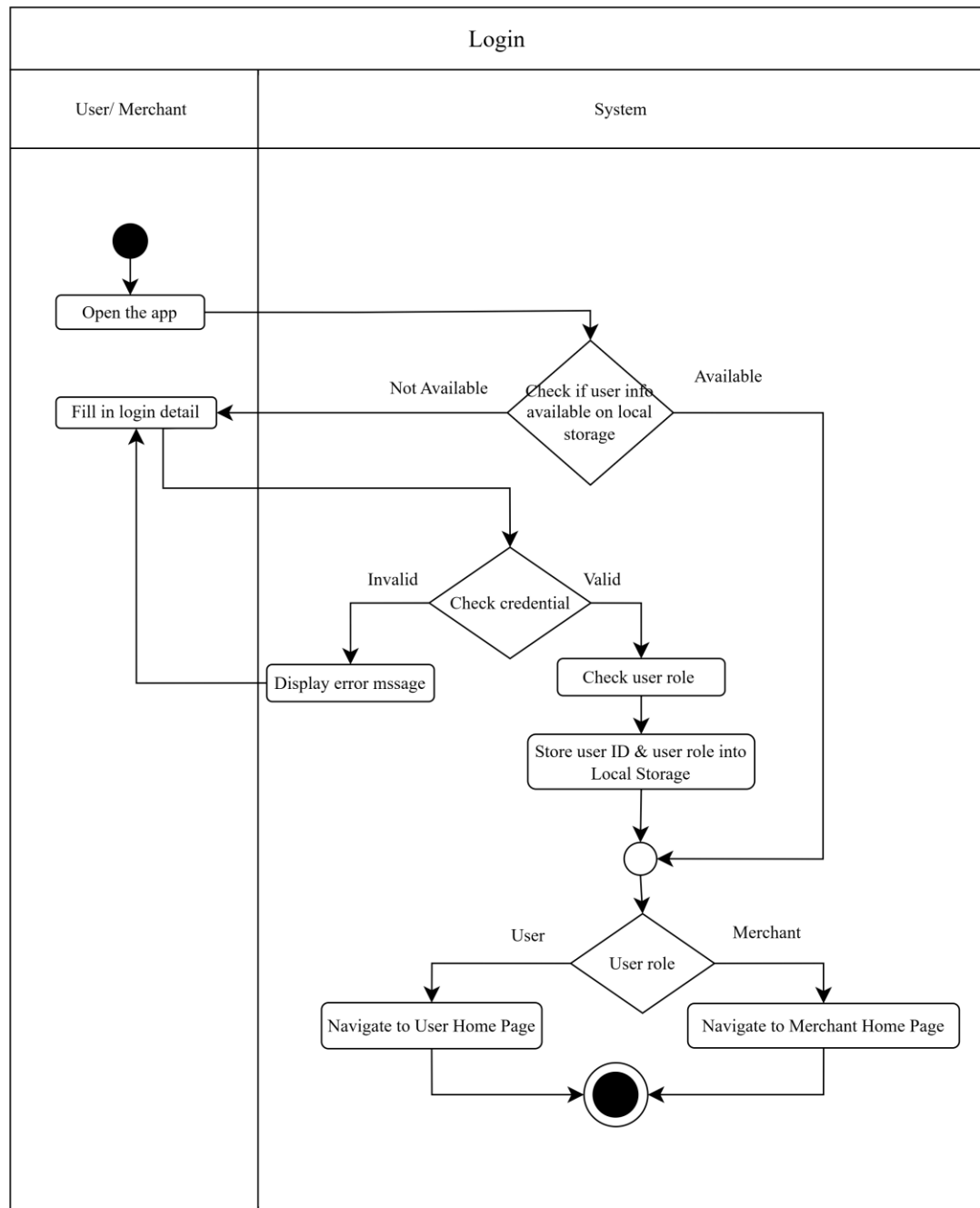


Figure 3.3.2 Login Activity Diagram

### 3.3.3 Log out

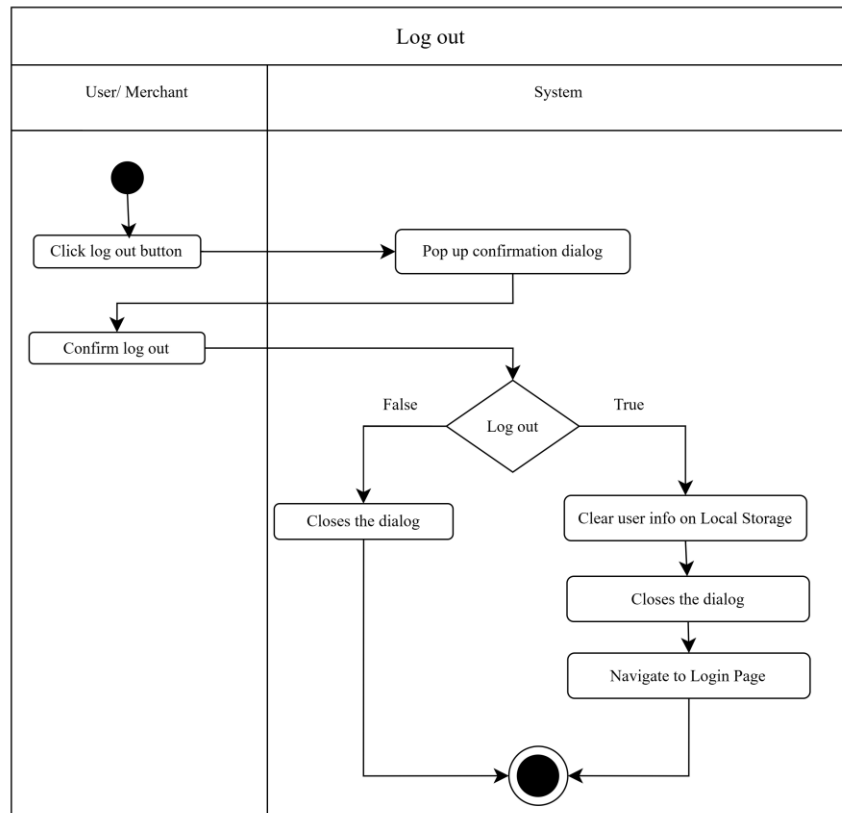


Figure 3.3.3 Log out Activity Diagram

### 3.3.4 Reset Password

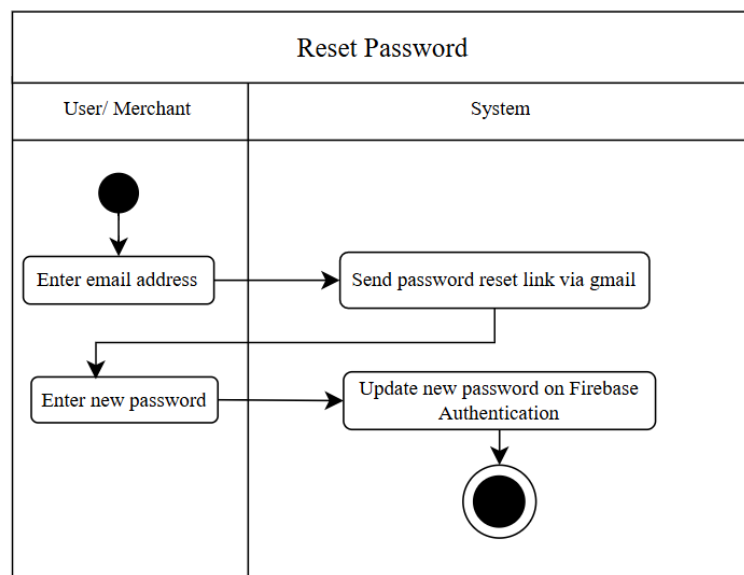


Figure 3.3.4 Reset Password Activity Diagram

### 3.3.5 Direct Message

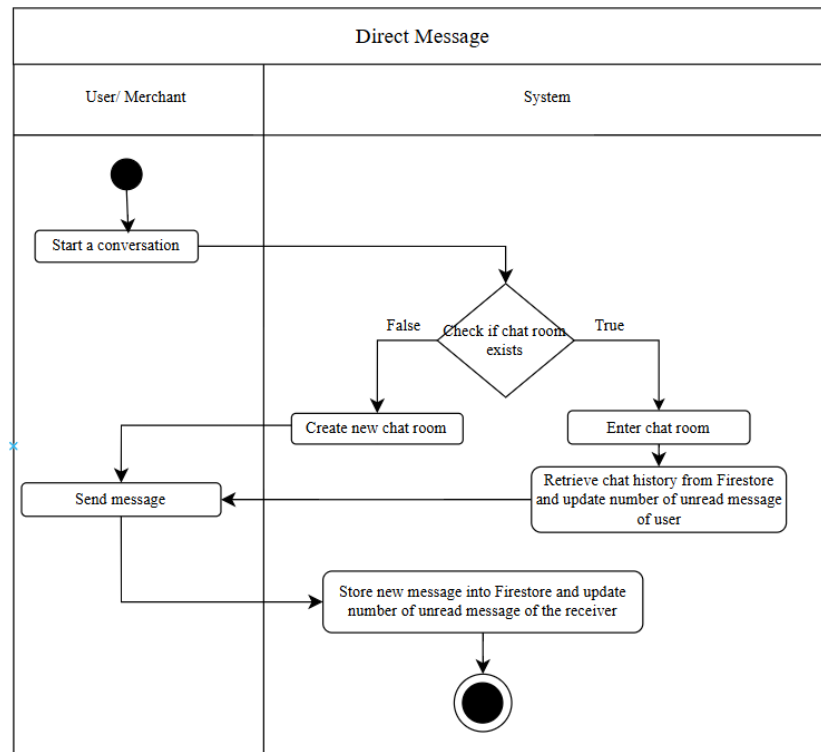


Figure 3.3.5 Direct Message Activity Diagram

### 3.3.6 View and Filter Reviews

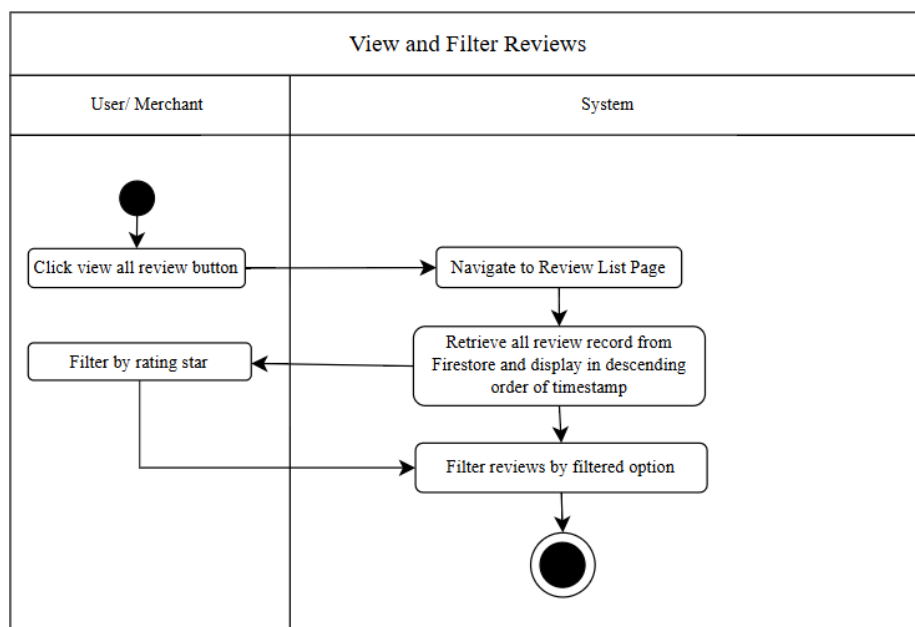


Figure 3.3.6 View and Filter Review Activity Diagram

### 3.3.7 View Map

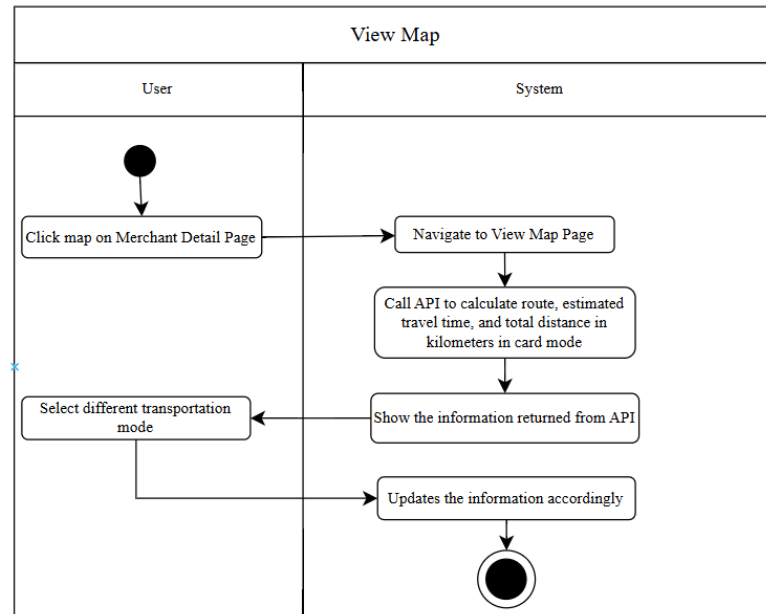


Figure 3.3.7 View Map Activity Diagram

### 3.3.8 Write Review

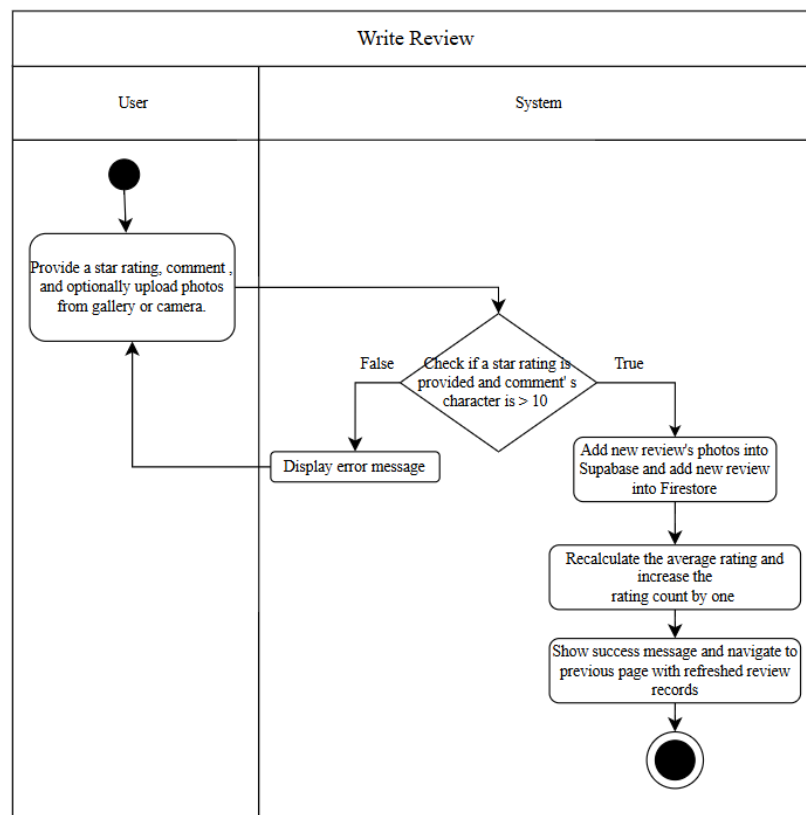


Figure 3.3.8 Write Review Activity Diagram

## 3.3.9 Edit Profile

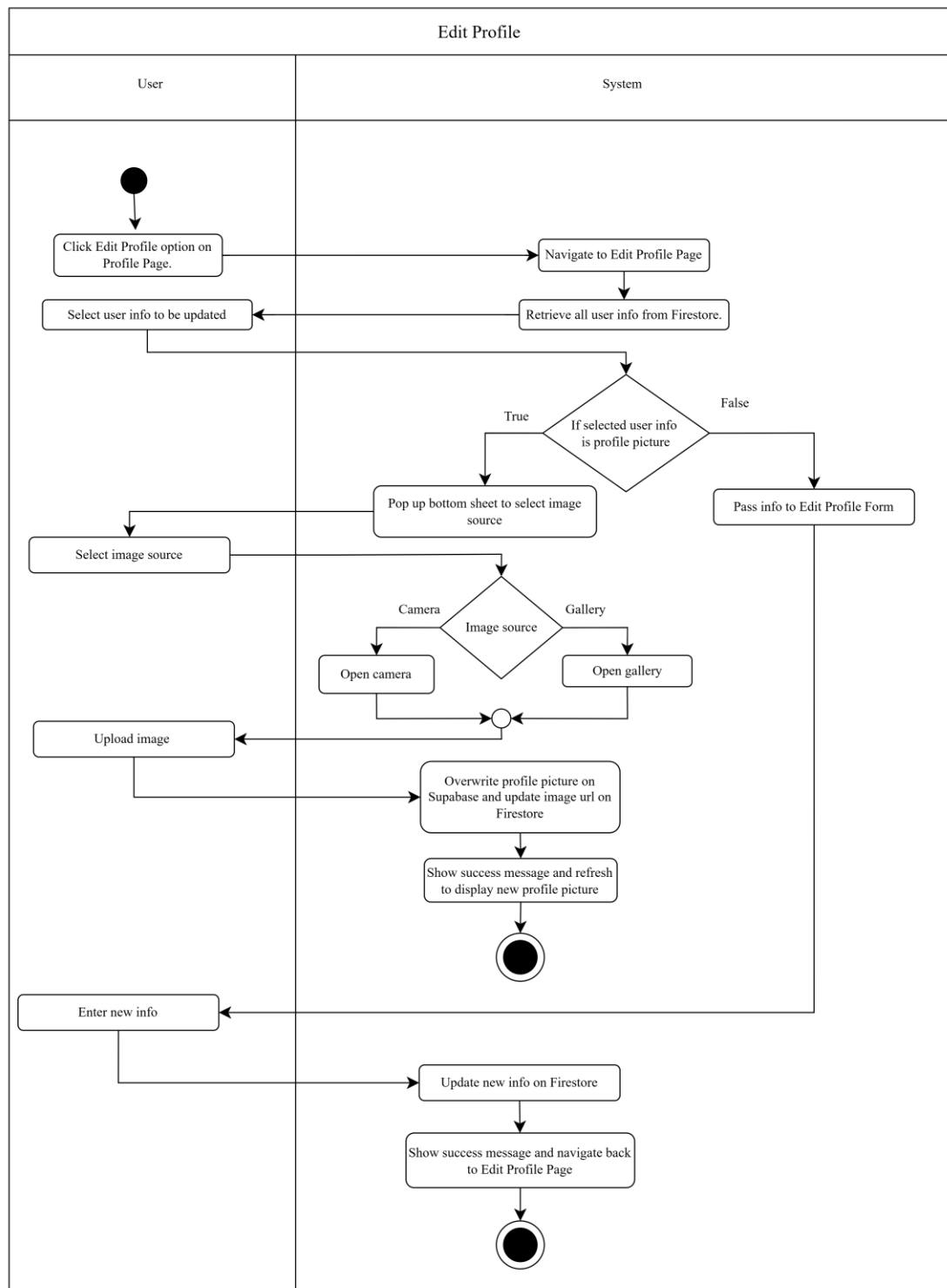


Figure 3.3.9 Edit Profile Activity Diagram

### 3.3.10 Real-time Search Merchant

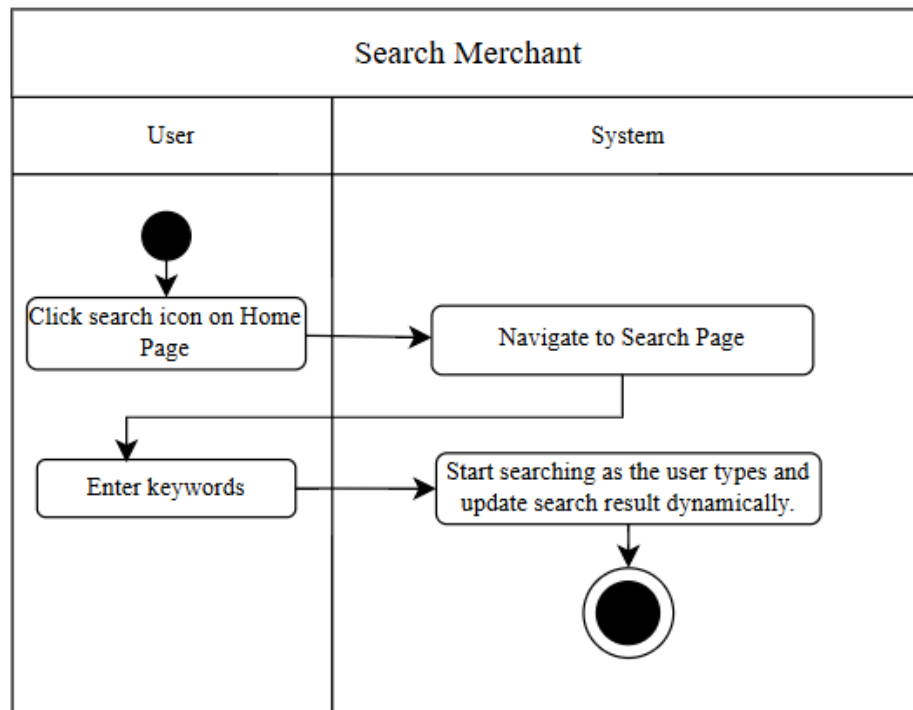


Figure 3.3.10 Real-time Search Merchant Activity Diagram

### 3.3.11 Filter Merchant

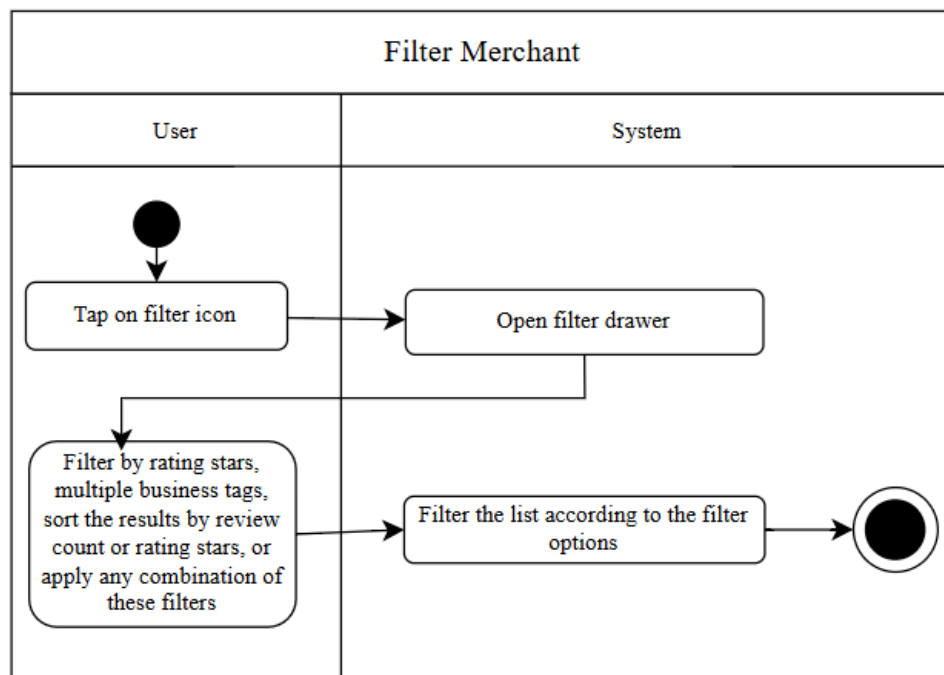


Figure 3.3.11 Filter Merchant Activity Diagram



### 3.3.12 Add/ Remove Merchant from Savelist

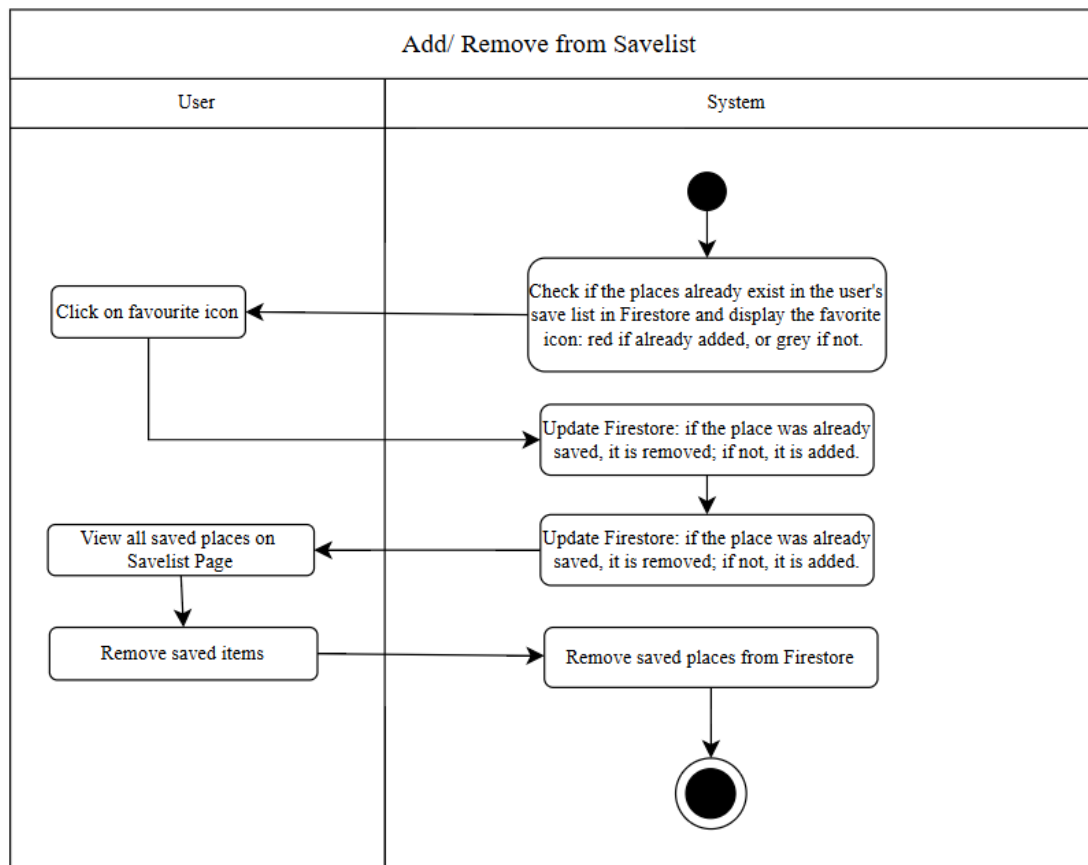


Figure 3.3.12 Add/ Remove Merchant from Savelist Activity Diagram

### 3.3.13 Edit Business Detail

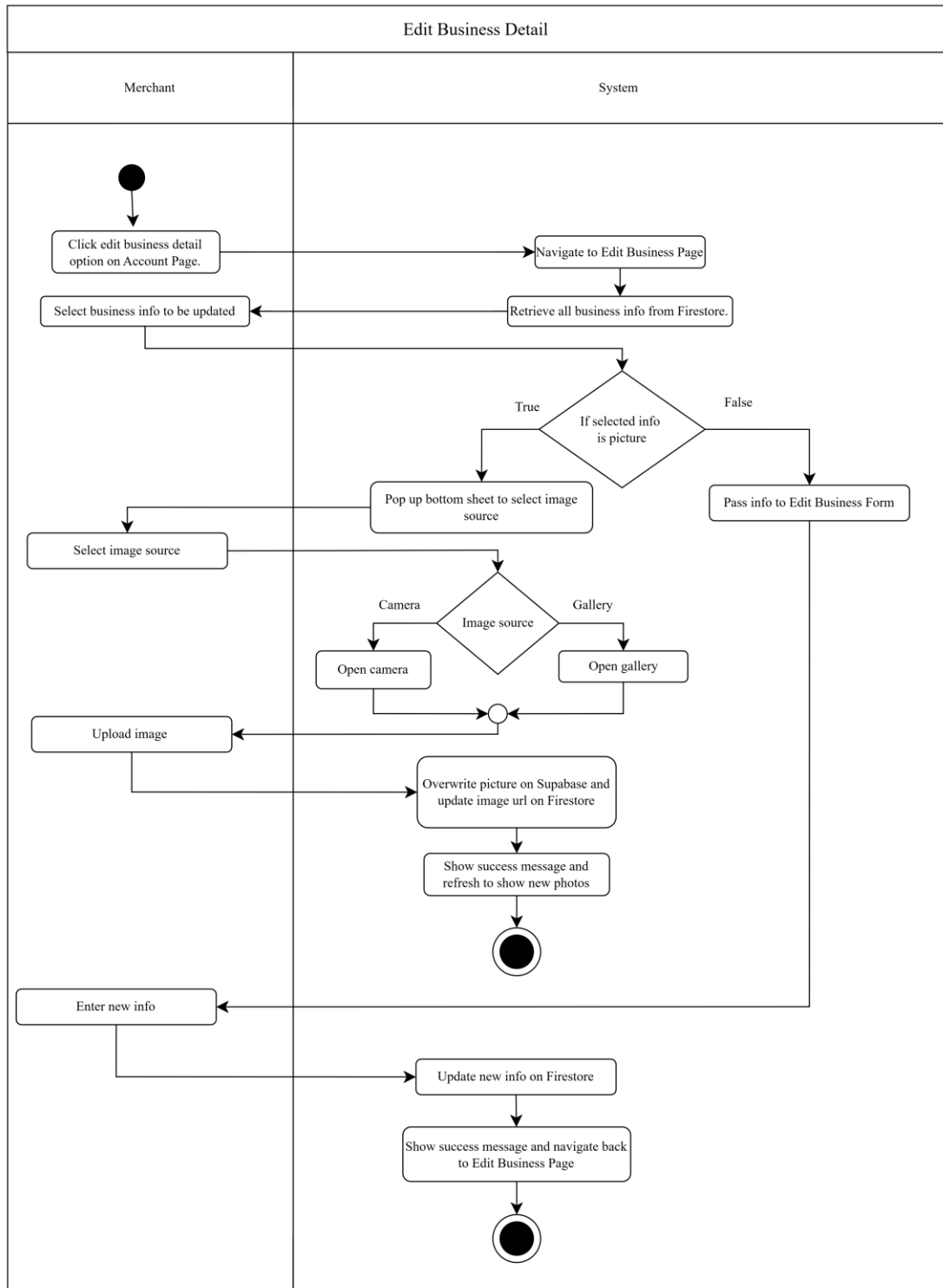


Figure 3.3.13 Edit Business Detail Activity Diagram

## 3.4 Sequence Diagram

### 3.4.1 Sign Up

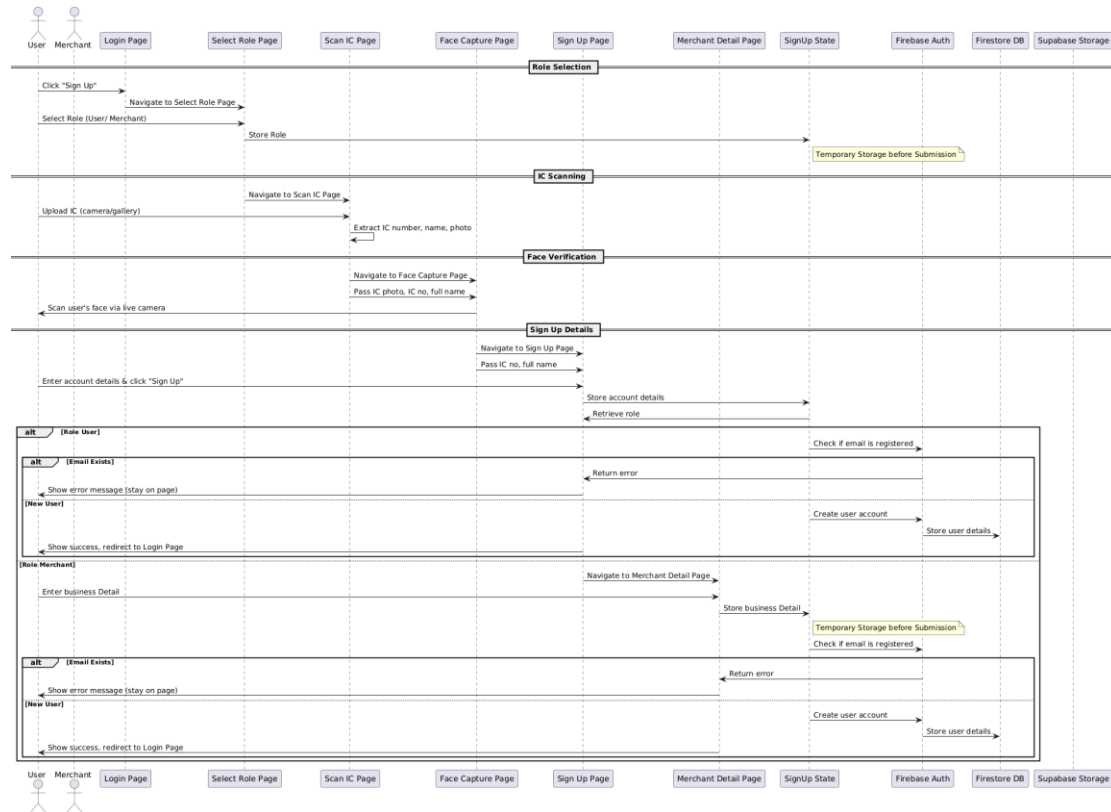


Figure 3.4.1 Sign Up Sequence Diagram

## 3.4.2 Login

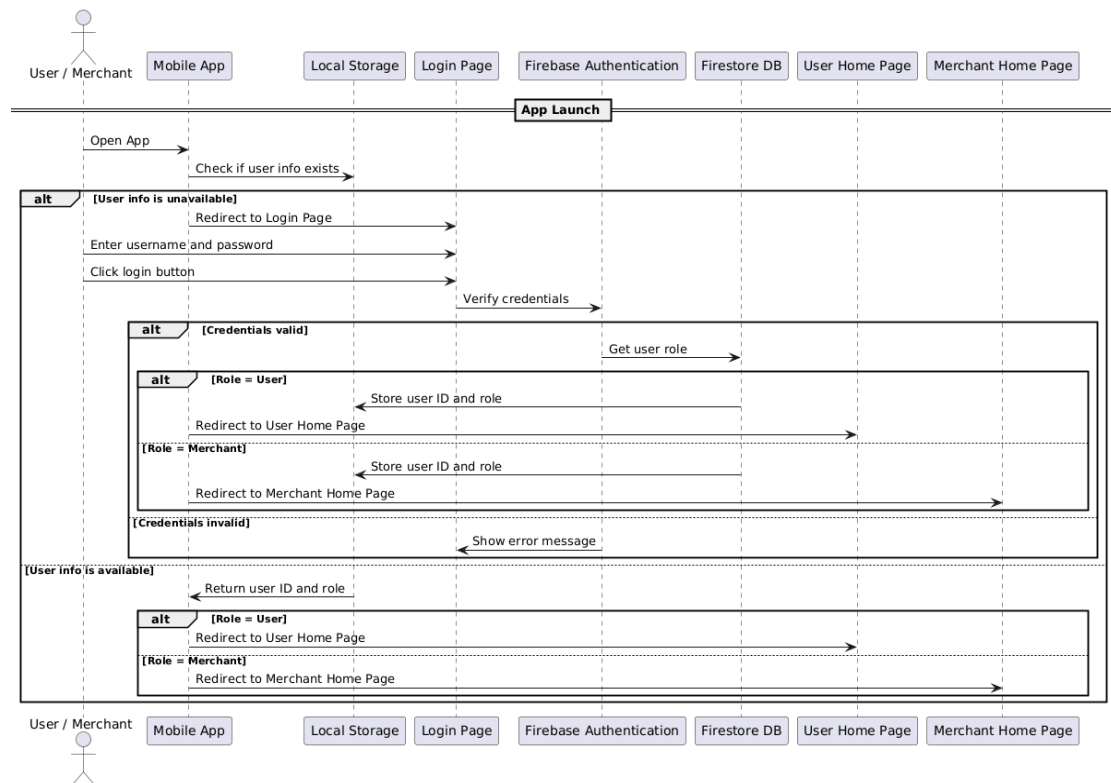


Figure 3.4.2 Login Sequence Diagram

## 3.4.3 Reset Password

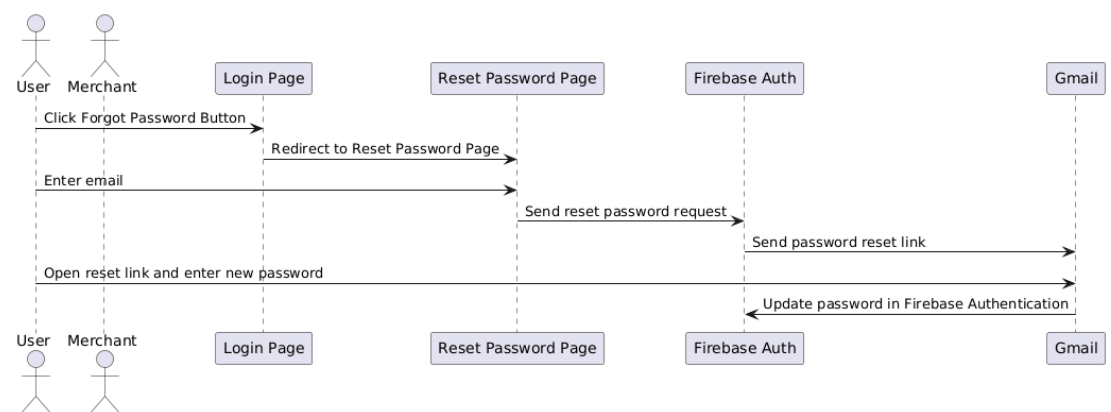


Figure 3.4.3 Reset Password Sequence Diagram

### 3.4.4 Log out

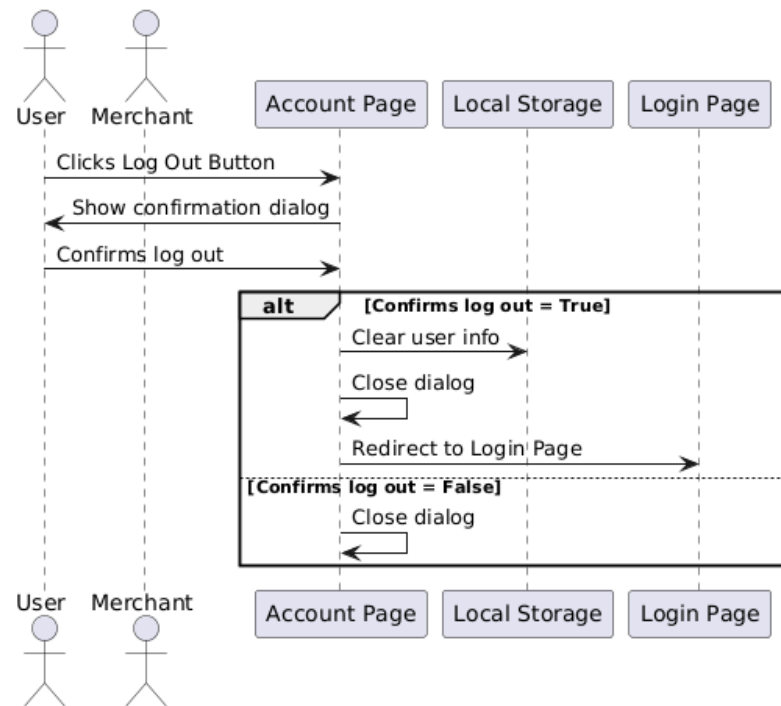


Figure 3.4.4 Log out Sequence Diagram

### 3.4.5 Direct Message

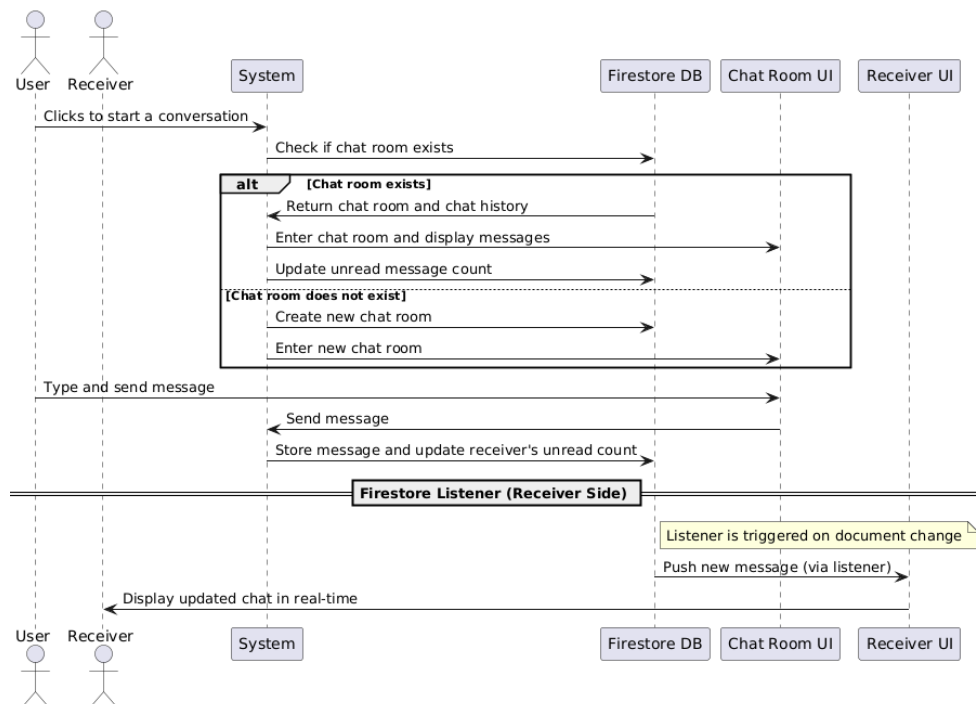


Figure 3.4.5 Direct Message Sequence Diagram

### 3.4.6 View and Filter Reviews

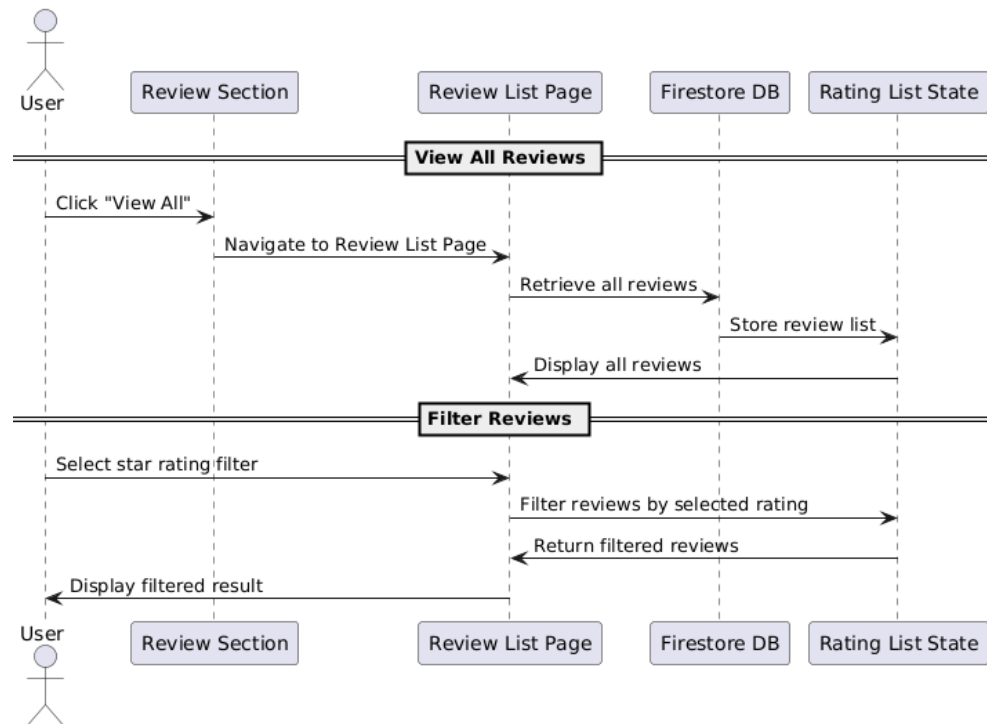


Figure 3.4.6 View and Filter Reviews Sequence Diagram

### 3.4.7 View Map

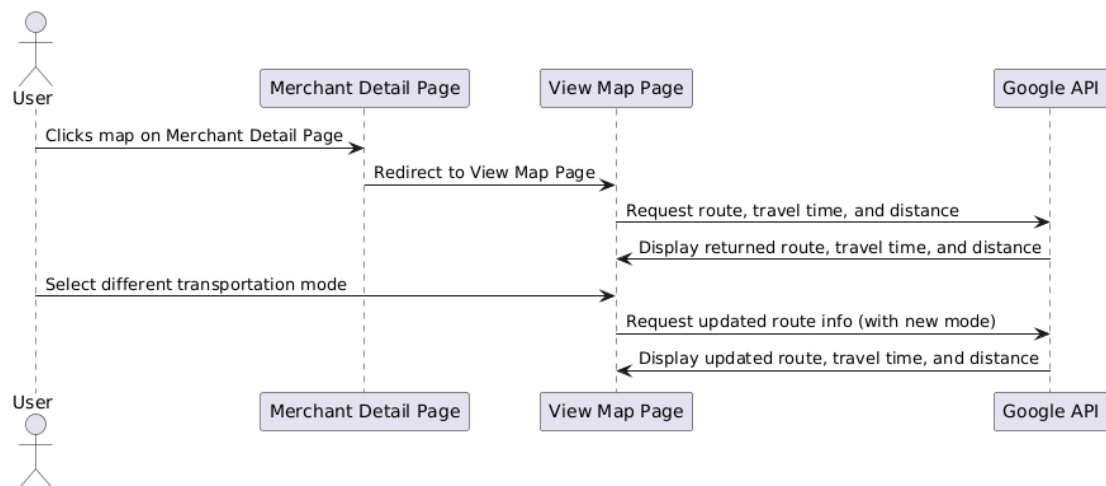


Figure 3.4.7 View Map Sequence Diagram

## 3.4.8 Write Review

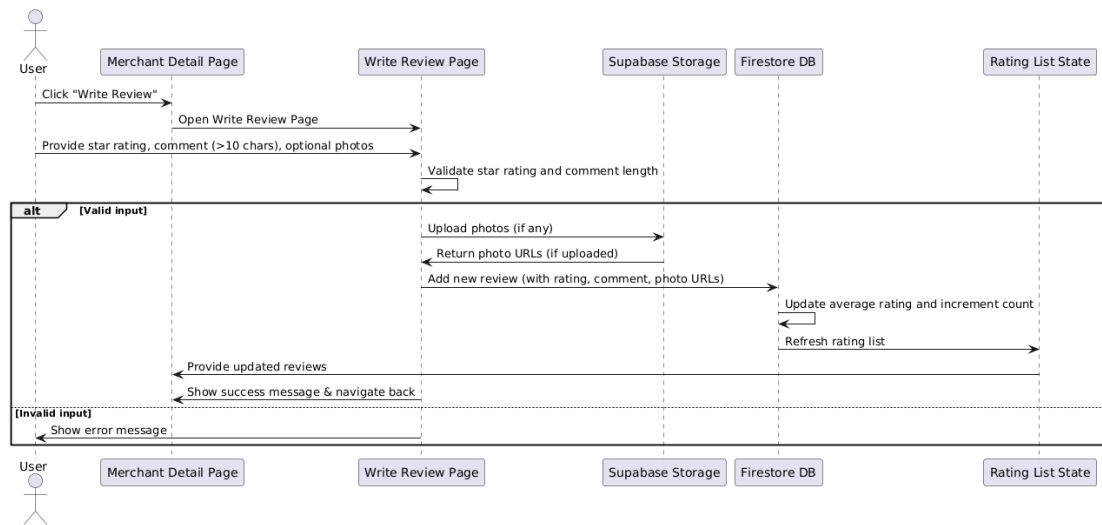


Figure 3.4.8 Write Review Sequence Diagram

## 3.4.9 Edit Profile

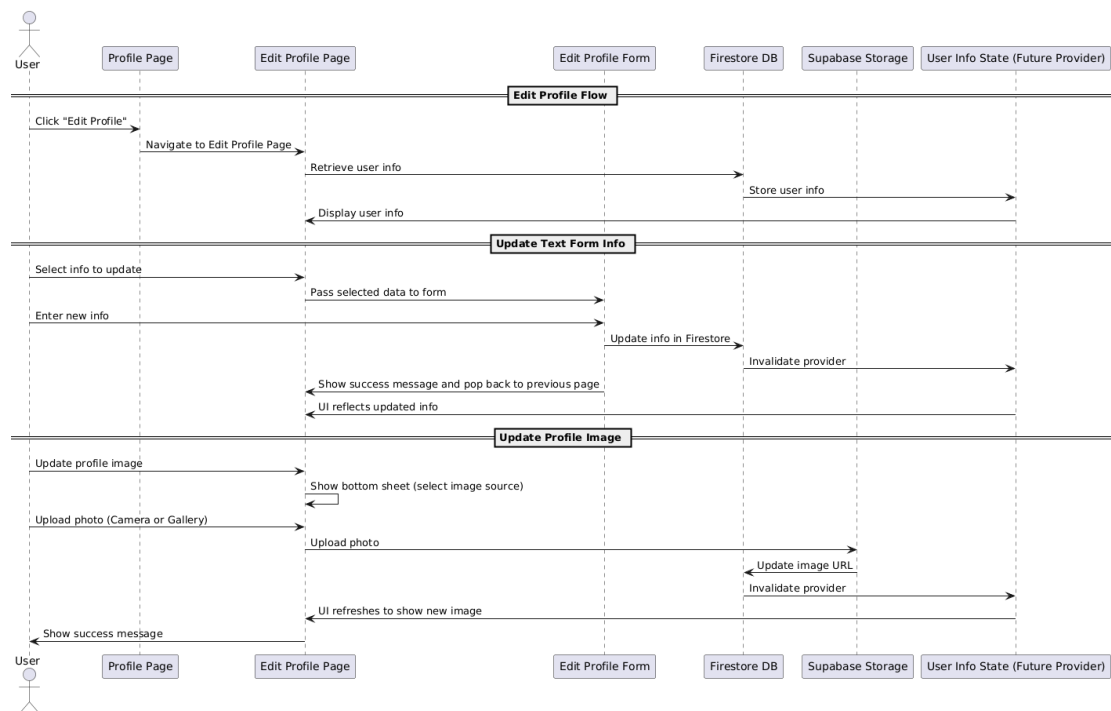


Figure 3.4.9 Edit Profile Sequence Diagram

### 3.4.10 Real-time Search Merchants

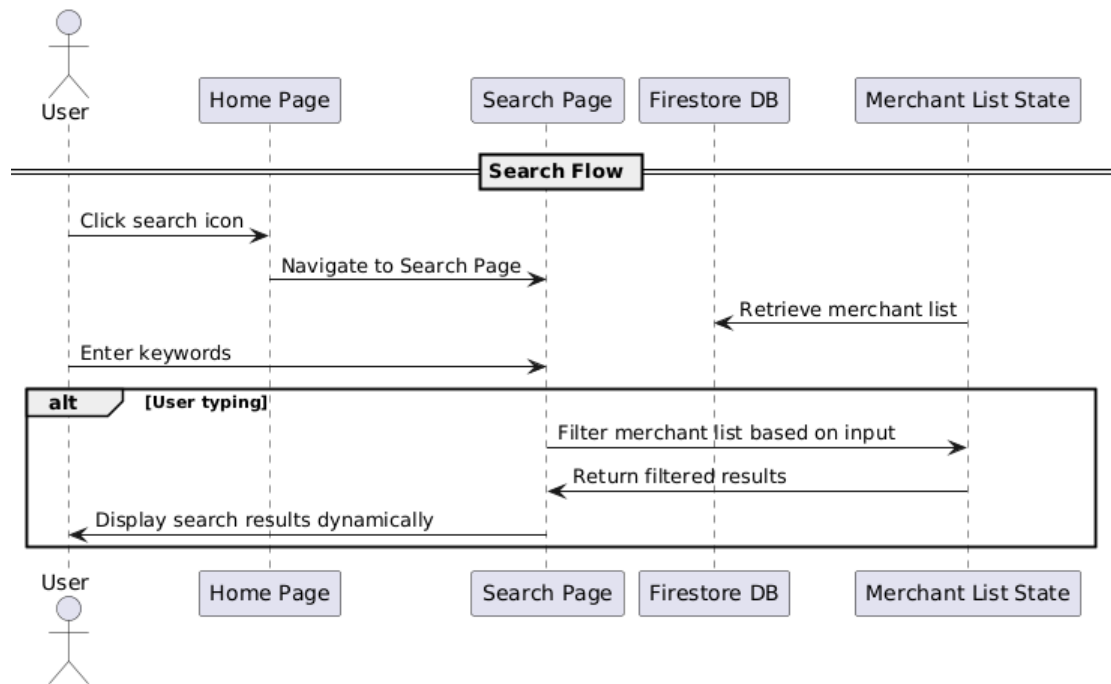


Figure 3.4.10 Real-time Search Merchants Sequence Diagram

### 3.4.11 Filter Merchants

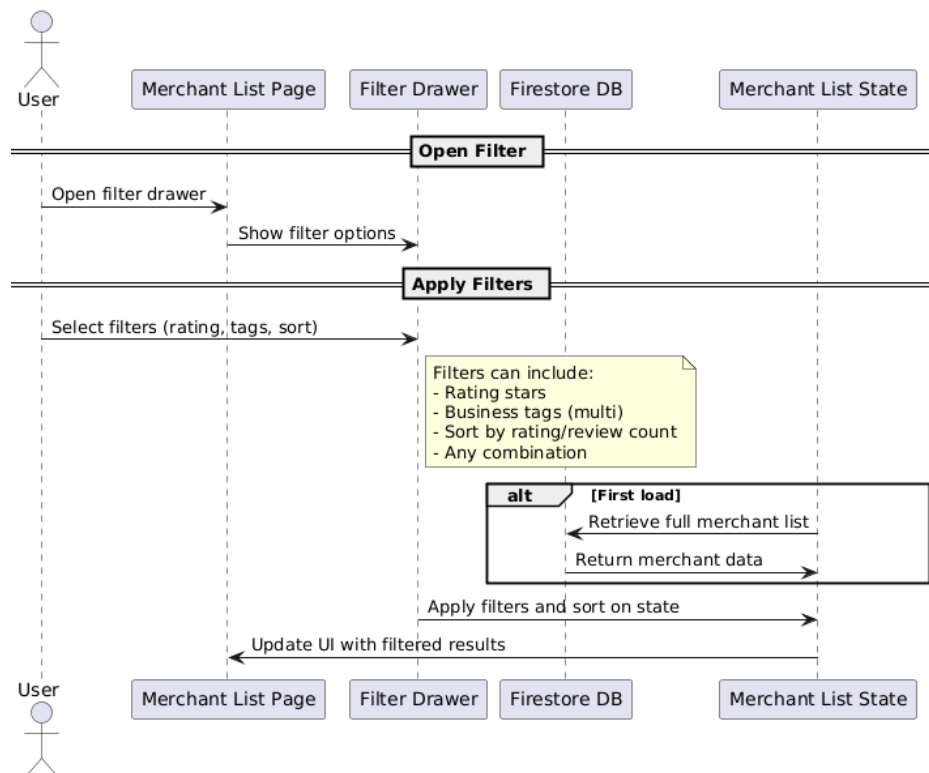


Figure 3.4.11 Filter Merchants Sequence Diagram



## 3.4.12 Add/Remove Merchant from Savelist

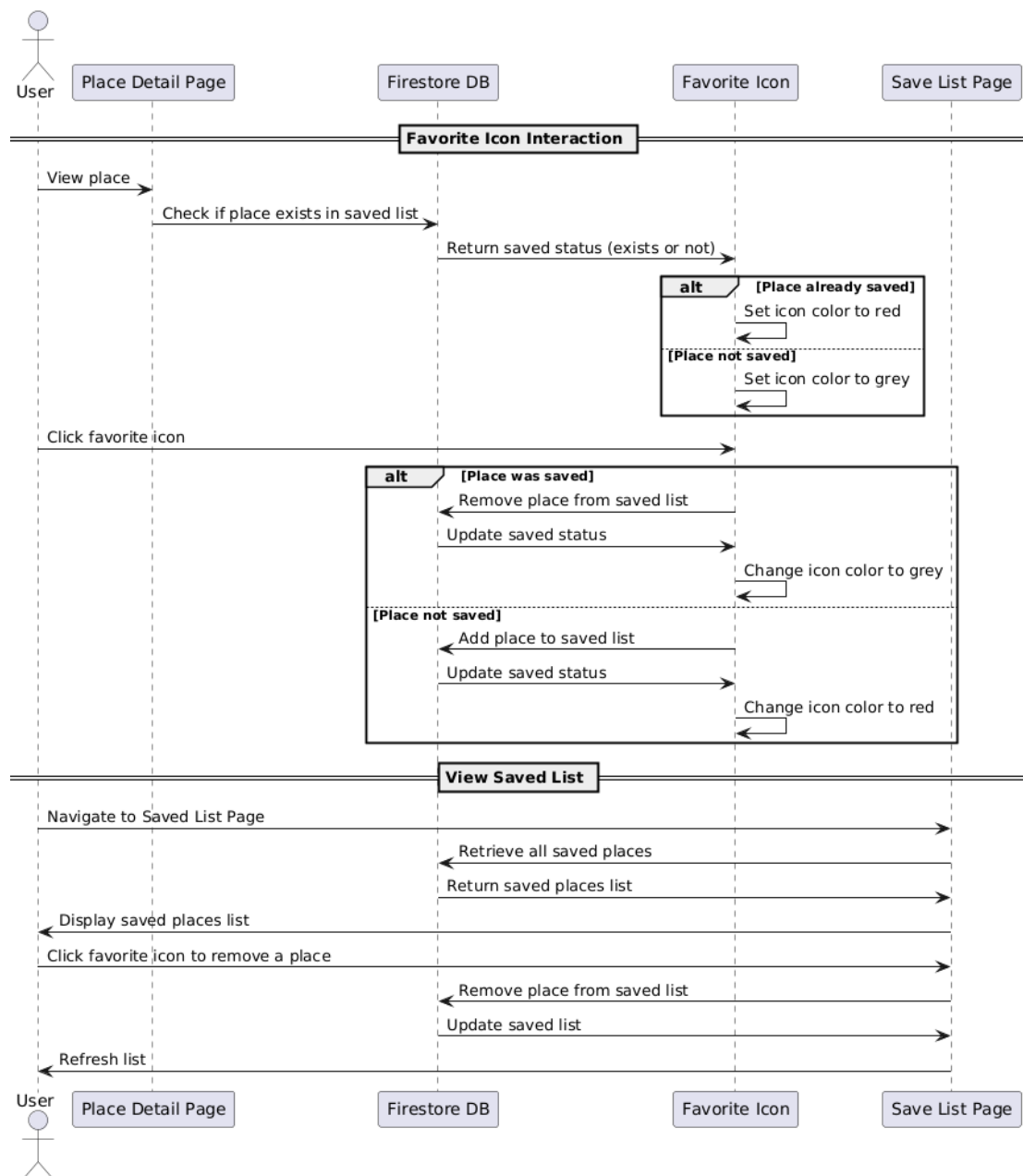


Figure 3.4.12 Add/ Remove Merchants from Savelist Sequence Diagram

## 3.4.13 Edit business detail

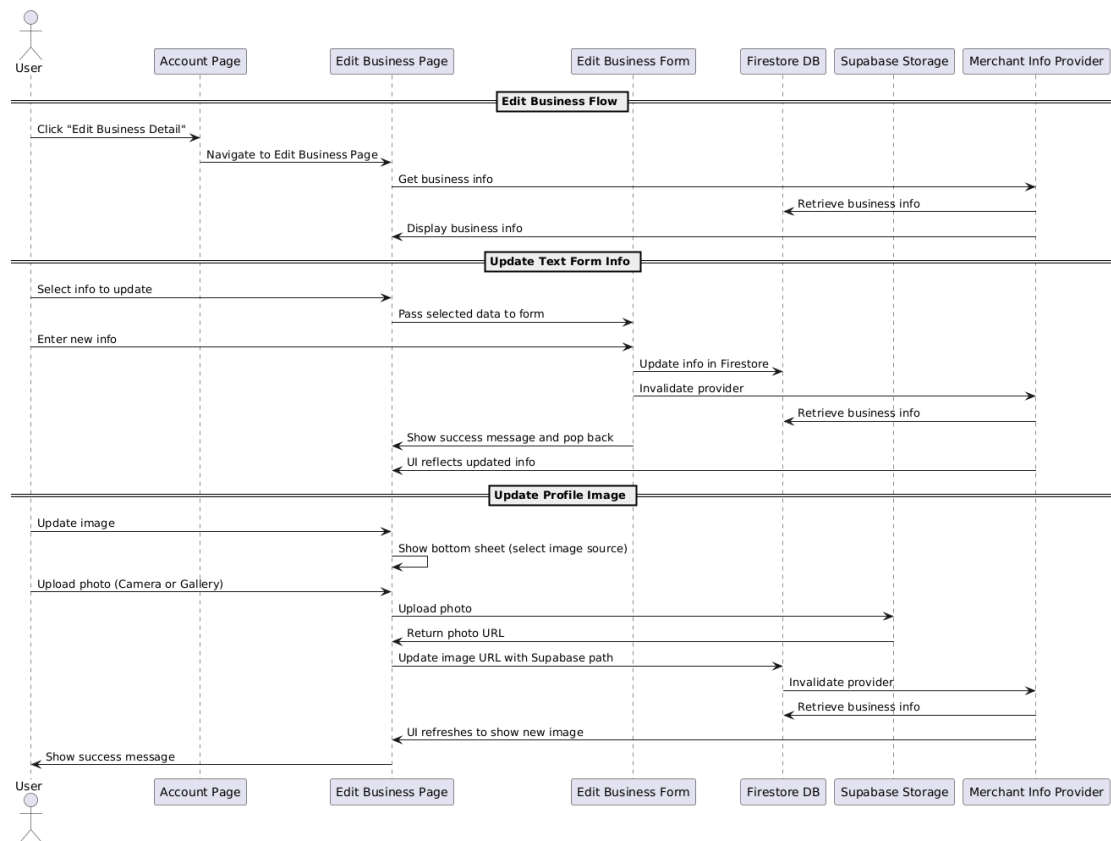


Figure 3.4.13 Edit Business Detail Sequence Diagram

### 3.5 Block Diagram

#### 3.5.1 User Block Diagram

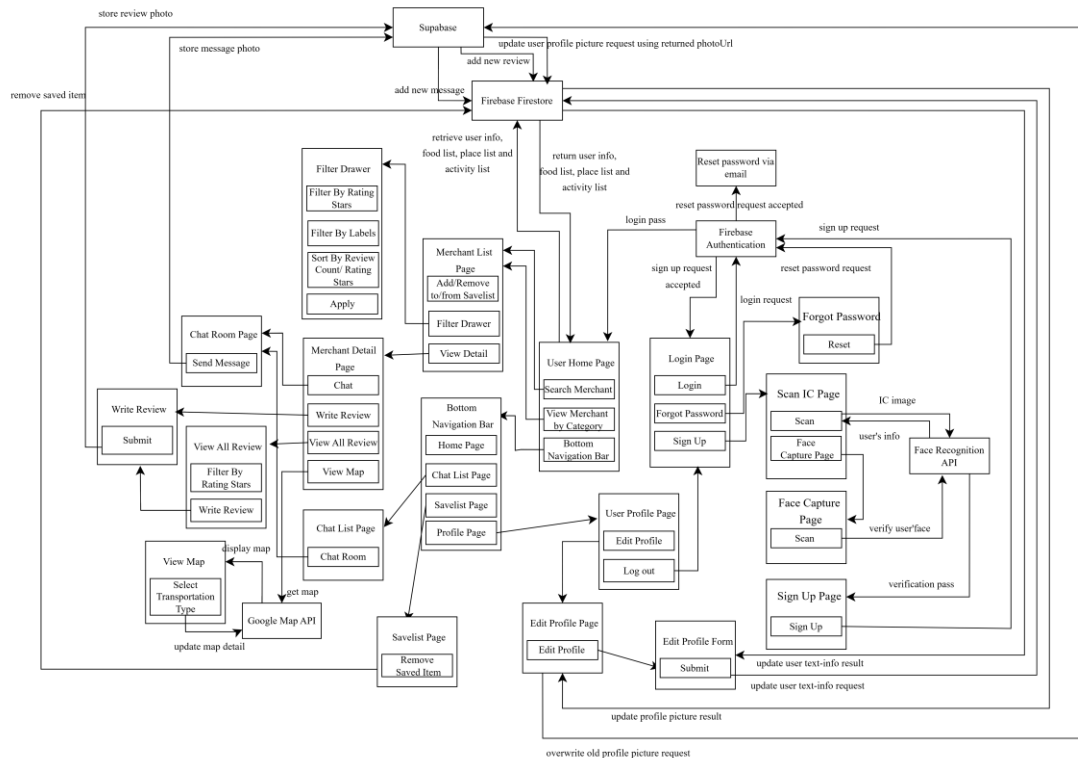


Figure 3.5.1 User Block Diagram

When a new user wants to join this application, they first visit the Sign Up Page. The user must undergo a real identity verification process where the user scans IC to capture the official ID information. After scanning the IC, the user must capture a live photo of the face. The system will compare the face from the IC and the live photo to verify that the user is genuine. After successfully scanning and capturing, the user completes registration at the Sign Up Page by submitting the extracted user info from IC as well as email and password. The system will check if the user's IC number or email address is registered. If same information is found on the firebase authentication and firestore, the system will throw error message, else it will store the user info and direct user to login page to log into the app. If an existing user forgets the password, the user can go to the Forgot Password Page to request a reset link via email.

## CHAPTER 3 SYSTEM REQUIREMENTS AND DESIGN

From the Home Page, the user can go to profile page to edit username and profile image. The user can explore the Merchant List Page, where they can add or remove merchants from the personal Save List, open the filter drawer to filter merchants by rating stars or multiple tags or sort by review count or average rating stars or any combinations of these three filter options. The user can search the intended visited places by typing keyword. The system will update the merchant list automatically upon user typing.

After the user selects a merchant, the system will lead the user to the merchant detail page, where they can chat with the merchant and other user directly, write a review based on their experience, view all reviews left by other users and view the location on a map. For the chat feature, the user can send messages. The system will store the message into the firestore and update the number of unread message on receiver side. This system supports real-time communication, where the system will listen to the firestore change and update the chat room instantly. For the write review feature, the user needs to provide a star rating, at least 10 characters of comment and optionally photos from gallery or camera. The system will store new review into the firestore, and update merchant detail by recalculating the average rating and incrementing rating count by one. The user can view all the reviews given by other users and filtered by rating star. For the map page, the system will show the route, total distance in kilometre, and total time taken to reach the destination in car mode. The user can select other transportation mode such as cycling and walking.

### 3.5.2 Merchant Block Diagram

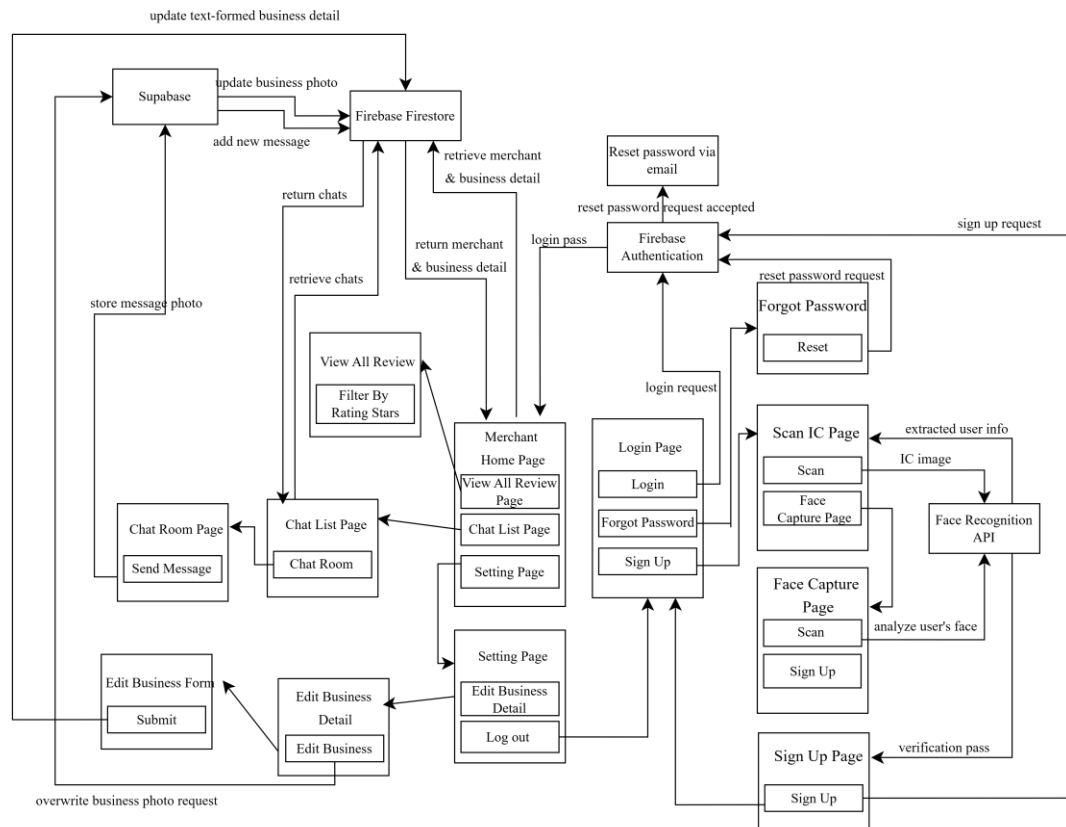


Figure 3.5.2 Merchant Block Diagram

The merchant shares the similar login, sign up and forgot password procedures as user. On merchant home page, the system will show all the reviews given by users and the merchant can look through the reviews by filtering via rating stars. By switching to the Chat List tab, the merchant can view a list of active conversations with users. Within each chat room, the merchant can respond to the direct messages from users, enabling real-time communication between users and merchants. The merchant can edit business detail.

## Chapter 4 Methodology, Tools And System Set up

### 4.1 Methodologu



Figure 4.1 Agile Development Cycle [12]

Agile development is implemented in this application since it is flexible to the change of user requirements. It focuses on coding and testing part rather than planing and designing. This allows quickly deliver of software application.

### 4.2 Tools to use

#### 4.2.1 Hardware

The hardware involved in this project is laptop and android mobile device. Laptop is used for different phases in the development of tourism search application. It supports comprehensive software development environments. While an android mobile device is used to test the application.

Description	Specifications
Model	Ideapad 3-14ITL6 Laptop – Type 82H7
Processor	Intel Core i5-1135G7
Operating System	Windows 10
Graphic	NVIDIA GeForce MX350 2GB
Memory	4GB DDR4-3200
Storage	512 GB SSD PCIe

Table 4.2.1.1 Specifications of laptop

Description	Specifications
Model	Vivo V29 5G
Processor	Qualcomm Snapdragon 778G SM7325 2.40 GHz Number of cores: 8
Operating System	Android 13
GPU	Adreno 642L
Memory	8GB + 256GB
Standard battery	Li-Po 4600 mAh

Table 4.2.1.2 Specifications of Android Mobile Device

## 4.2.2 Software

### 4.2.2.1 Flutter

Flutter is an open source UI toolkit developed by Google to develop a cross-platform application for android, ios, web, linux, macos and windows. Flutter is selected for this application development due to its powerful and customizable widgets, in creating visually appealing and interactive user interfaces. It supports extensive libraries and plugins for easier features. It also provides a hot reload feature, allowing the developer to track the changes to the user interface and system flow, without restarting the application.

### 4.2.2.2 Firebase

Firebase is a free open source cloud-based data management service. Firebase is integrated due to its services provided such as authentication and firestore. Authentication is utilized to validate a user's identity during the sign up and login processes. Firestore serves as the primary NoSQL database where user information, application data, and references such as image URLs are stored. In addition, by adding the Firebase SDKs and performing basic configurations, the app can easily integrate Firebase services without manually establishing HTTP connections. The SDKs handle

all underlying communication with Firebase services, such as Authentication and Firestore, through secure and optimized API call.

### **4.2.2.3 Supabase**

Supabase is utilized as the primary storage solution for user-uploaded photos. Instead of using Firebase Storage, which incurs costs based on usage, Supabase Storage offers a more cost-effective alternative, especially during development and for projects with limited budgets. When a user uploads a photo, the image is stored in a Supabase bucket, and the publicly accessible URL generated by Supabase is then stored in Firebase Firestore. This approach maintains the application's integration with Firestore for data management while reducing expenses by offloading media storage to Supabase. By separating data and file storage between Firestore and Supabase respectively, the system remains scalable and efficient without compromising on performance or usability.

### **4.2.2.4 Google Map API**

Google Maps API is integrated into the application to enable a range of location-based services that enhance user experience and functionality. Through this integration, the app can access the device's real-time location using GPS, allowing features such as displaying the current location on the map. Additionally, the API provides capabilities to generate directions between two points, visually display the route path on the map, and calculate important travel details such as estimated time of arrival (ETA) and distance between locations. The API also supports dynamic map rendering, markers, and customization options, enabling the app to deliver interactive and user-friendly mapping experiences. The seamless integration of Google Maps ensures reliable and accurate geolocation services while leveraging Google's robust infrastructure.



### 4.2.2.5 Face Detection and Face Recognition API

The application integrates Face++ API to implement face detection and face recognition functionalities, particularly for identity verification purposes. Face detection is used to scan and analyze the user's facial features from a live camera. In this system, when a user uploads their IC via gallery or camera, the application extracts the facial image from the IC using Face++'s detection capabilities. Then, the app captures a live photo of the user's face through the camera. The face recognition feature of the API then compares the facial data from the IC image with the live-captured photo to verify if they belong to the same person. Face++ provides a confidence score based on similarity, which helps determine whether the identity is valid. This process enhances the security and reliability of identity verification in the application by ensuring that the person presenting the IC is indeed its rightful owner. It eliminates manual checks and automates the comparison process with high accuracy.

## 4.3 System Set Up

### 4.3.1 Flutter

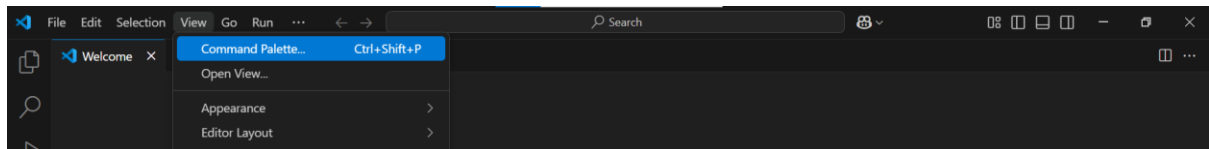


Figure 4.3.1.1 Open command palette on Visual Studio Code

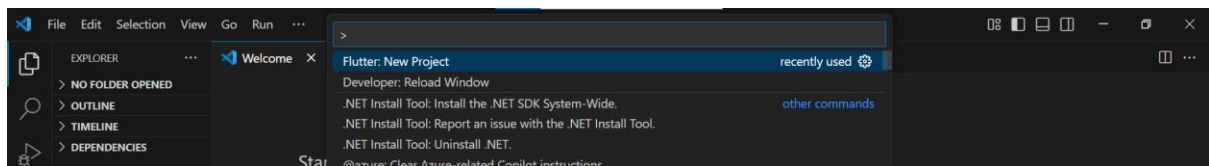


Figure 4.3.1.2 Create New Flutter Project

## CHAPTER 4 METHODOLOGY, TOOLS & SYSTEM SET UP

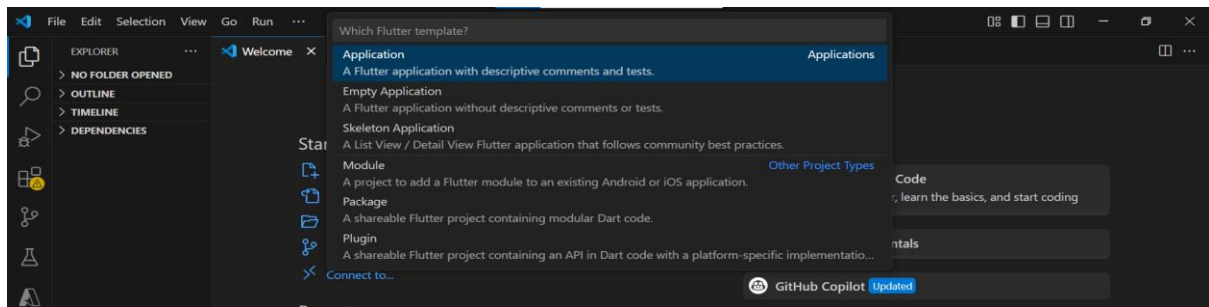


Figure 4.3.1.3 Create New Application

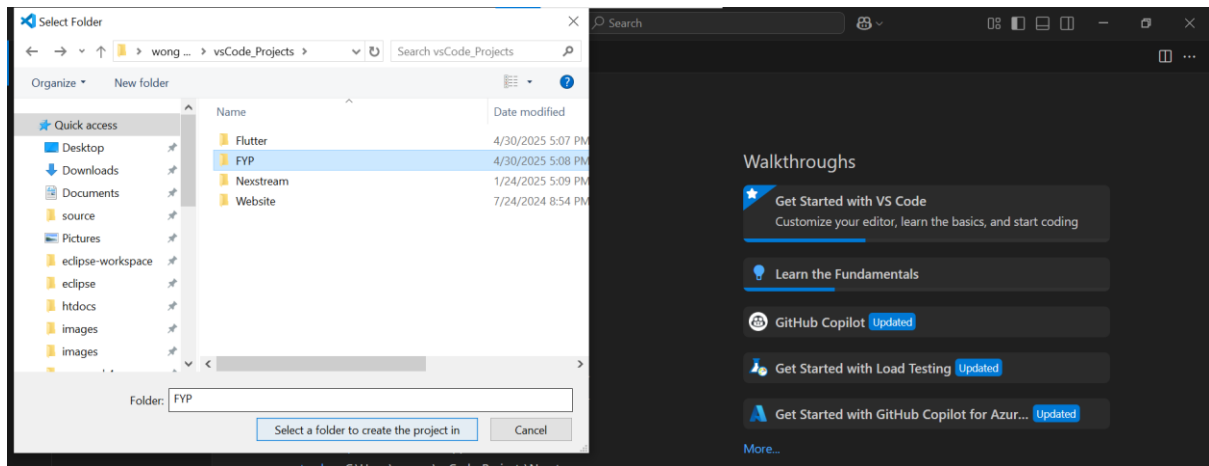


Figure 4.3.1.4 Select a folder where this application is located

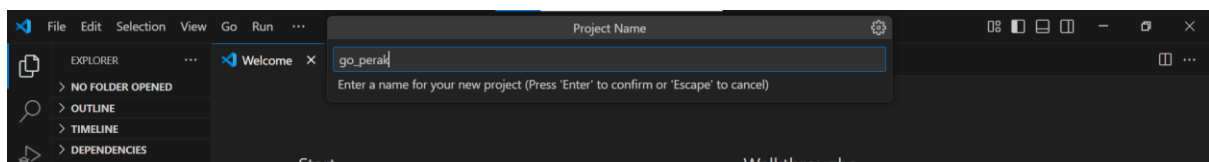


Figure 4.3.1.5 Enter project name

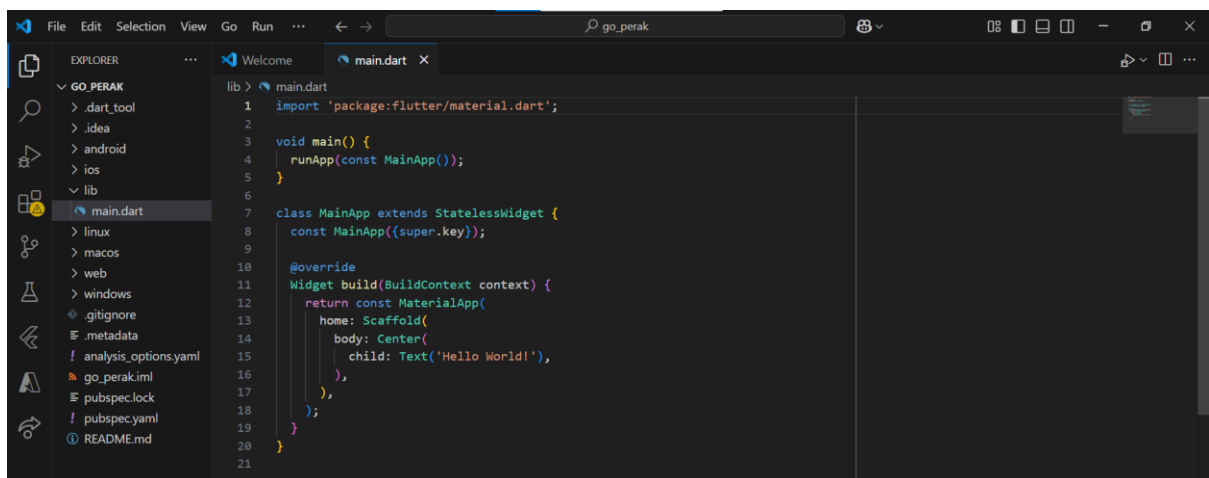


Figure 4.3.1.6 Main.dart is generated, confirming successful creation of the application

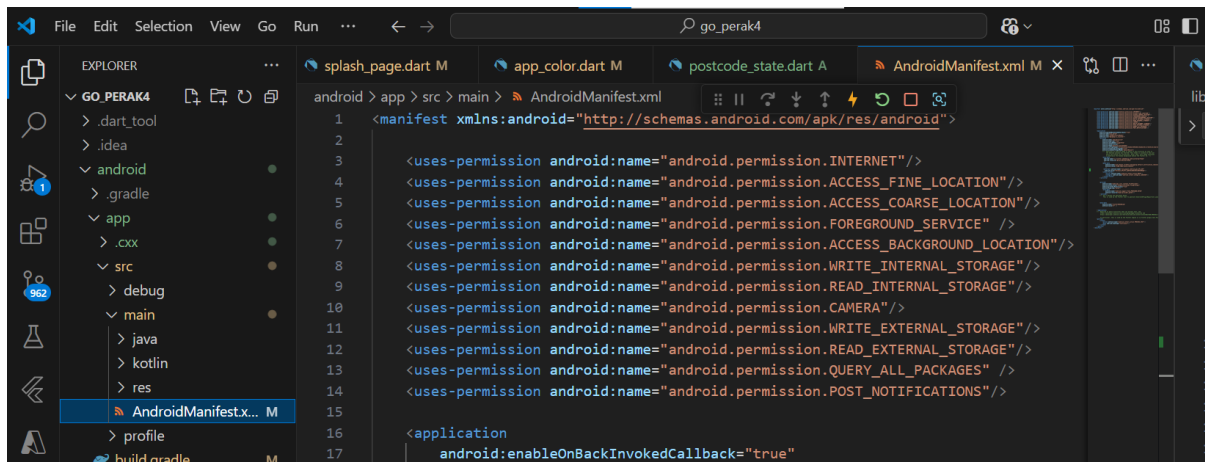


Figure 4.3.1.7 Set up user permissions on Android Manifest.xml

### 4.3.2 Firebase

#### (A) Set up Firebase Project

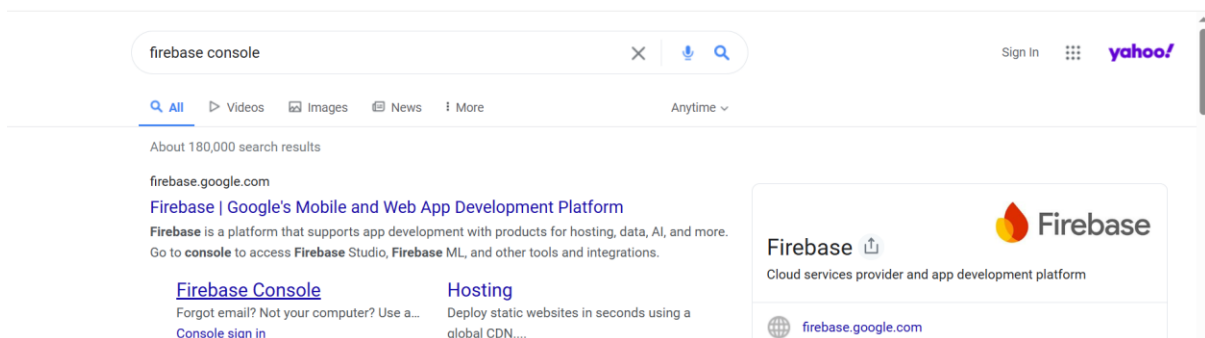


Figure 4.3.2.(A).1 Go to Firebase Console

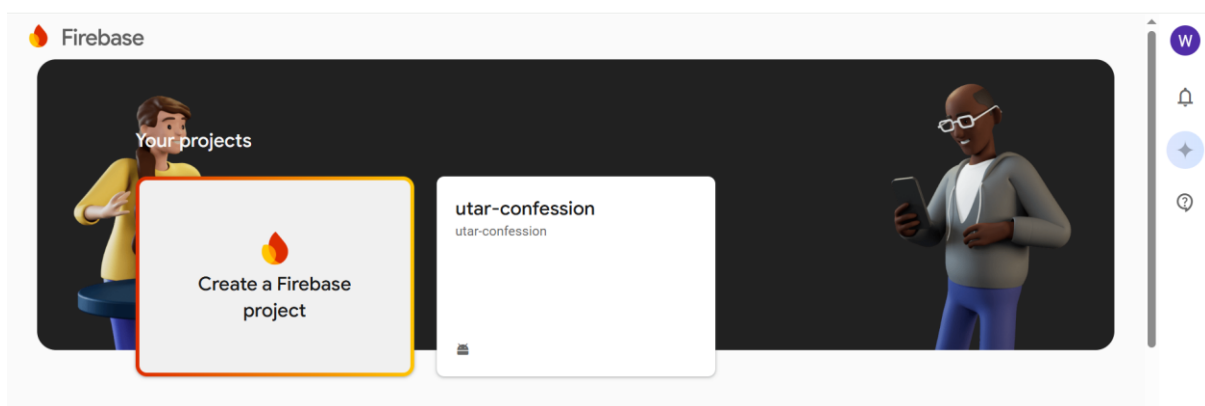
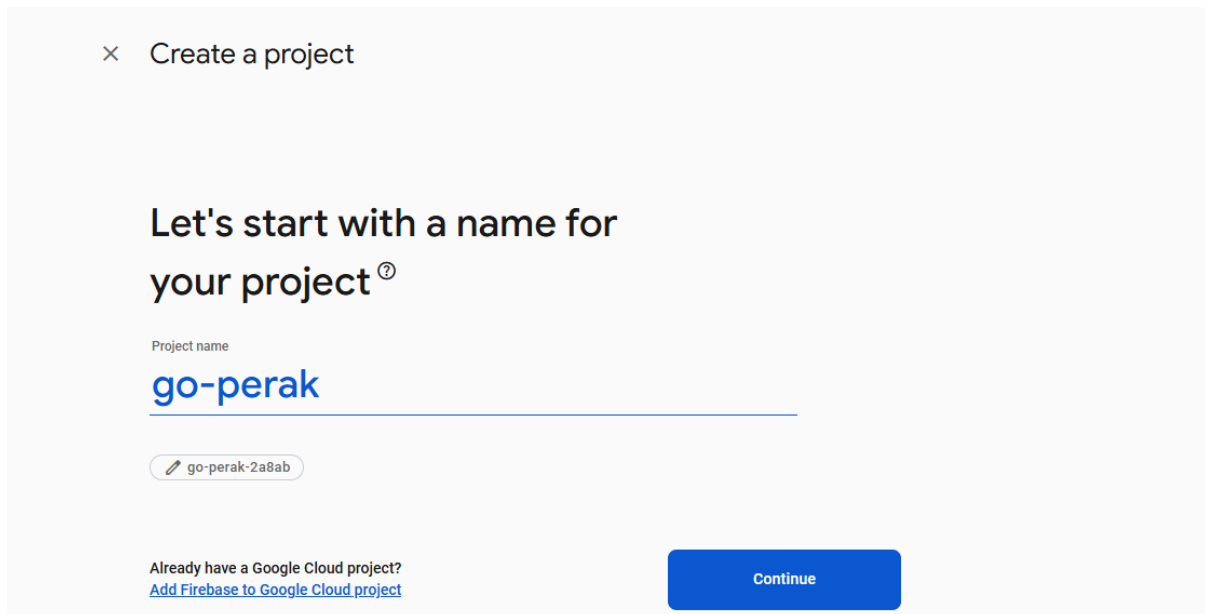


Figure 4.3.2.(A).2 Create a Firebase Project



× Create a project

## Let's start with a name for your project<sup>?</sup>

Project name

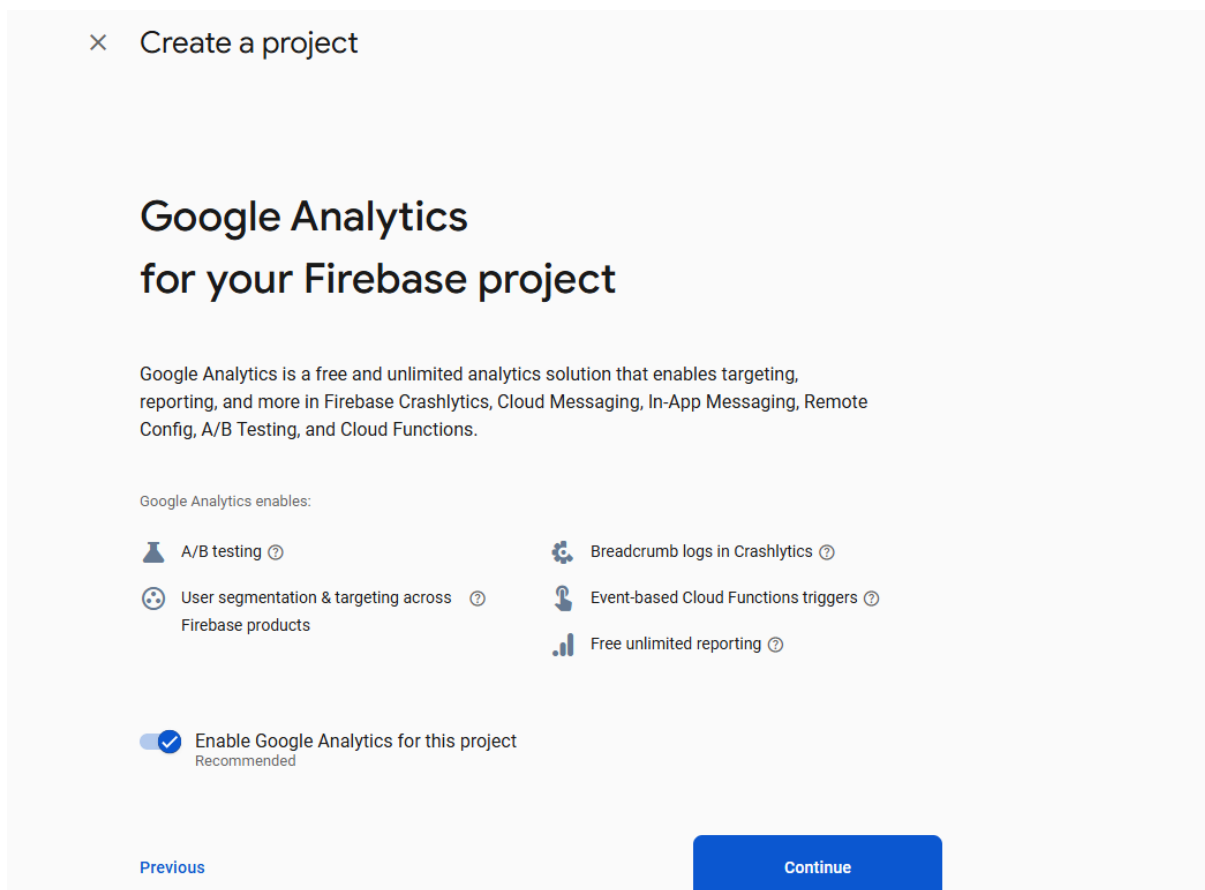
go-perak

go-perak-2a8ab

Already have a Google Cloud project?  
[Add Firebase to Google Cloud project](#)

Continue

Figure 4.3.2.(A).3 Enter Project Name



× Create a project

## Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.

Google Analytics enables:

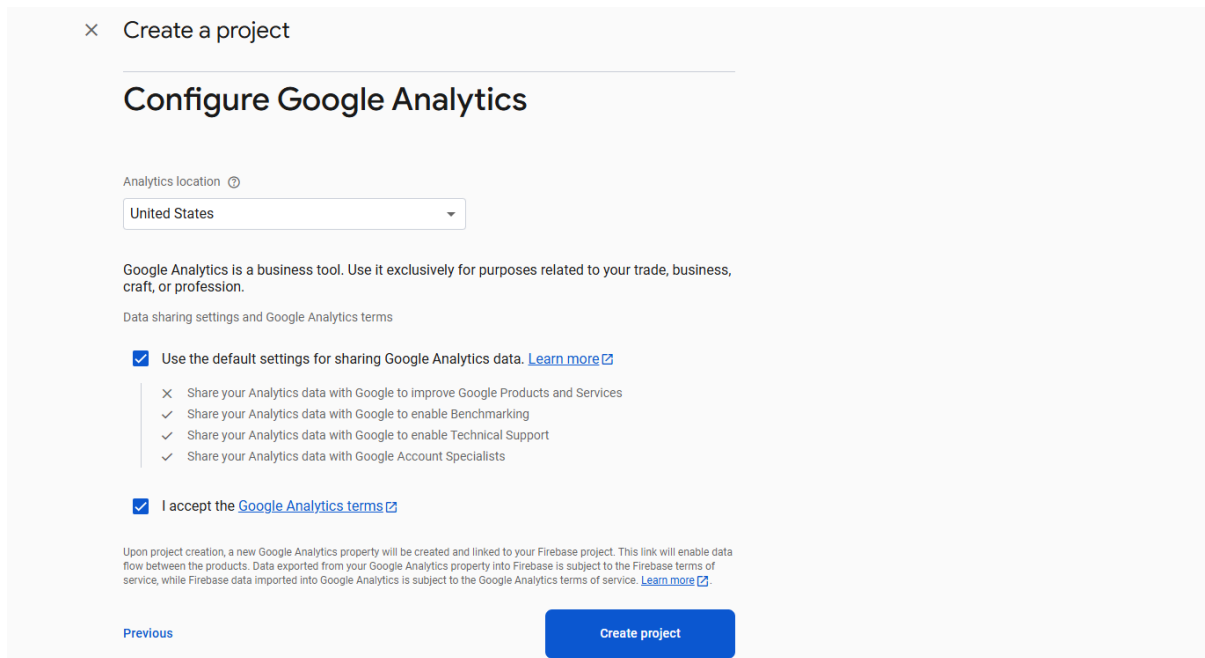
- A/B testing<sup>?</sup>
- User segmentation & targeting across Firebase products<sup>?</sup>
- Breadcrumb logs in Crashlytics<sup>?</sup>
- Event-based Cloud Functions triggers<sup>?</sup>
- Free unlimited reporting<sup>?</sup>

☒ Enable Google Analytics for this project  
Recommended

Previous

Continue

Figure 4.3.2.(A).4 Enable Google Analytics



× Create a project

### Configure Google Analytics

Analytics location ⓘ  
United States ▼

Google Analytics is a business tool. Use it exclusively for purposes related to your trade, business, craft, or profession.

Data sharing settings and Google Analytics terms

☒ Use the default settings for sharing Google Analytics data. [Learn more](#)

- × Share your Analytics data with Google to improve Google Products and Services
- ✓ Share your Analytics data with Google to enable Benchmarking
- ✓ Share your Analytics data with Google to enable Technical Support
- ✓ Share your Analytics data with Google Account Specialists

☒ I accept the [Google Analytics terms](#)

Upon project creation, a new Google Analytics property will be created and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#)

[Previous](#) [Create project](#)

Figure 4.3.2.(A).5 Select firebase location

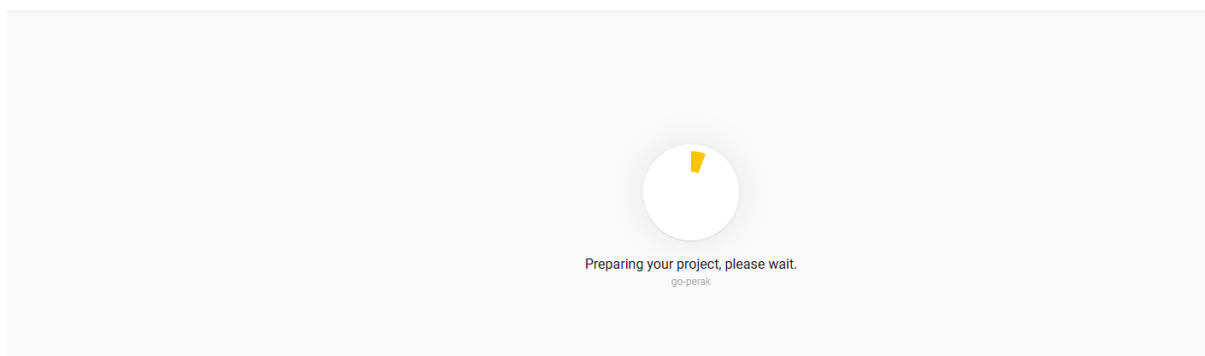


Figure 4.3.2.(A).6 Wait for the firebase to be prepared

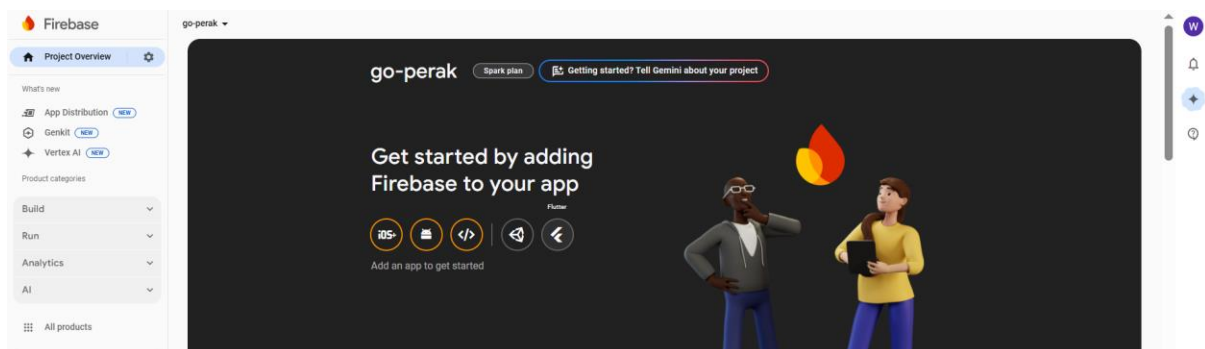


Figure 4.3.2.(A).7 Firebase Dashboard displayed, confirming successful setup

### (B) Link Firebase to Flutter Project

```
C:\Users\wongw>cd C:\Users\wongw\vsCode_Projects\FYP\go_perak\
C:\Users\wongw\vsCode_Projects\FYP\go_perak>flutterfire configure --project=go-perak-2a8ab
Found 7 Firebase projects. Selecting project go-perak-2a8ab.
Which platforms should your configuration support (use arrow keys & space to select)? · android, ios, macos, web, windows
Registered a new Firebase android app on Firebase project go-perak-2a8ab.
Firebase android app com.example.go_perak is not registered on Firebase project go-perak-2a8ab.
Registered a new Firebase ios app on Firebase project go-perak-2a8ab.
Firebase ios app com.example.goPerak is not registered on Firebase project go-perak-2a8ab.
Registered a new Firebase ios app on Firebase project go-perak-2a8ab.
Firebase macos app com.example.goPerak registered.
Firebase web app go_perak (web) is not registered on Firebase project go-perak-2a8ab.
Registered a new Firebase web app on Firebase project go-perak-2a8ab.
Firebase windows app go_perak (windows) is not registered on Firebase project go-perak-2a8ab.
Registered a new Firebase windows app on Firebase project go-perak-2a8ab.

Firebase configuration file lib\firebase_options.dart generated successfully with the following Firebase apps:

Platform  Firebase App Id
web        1:11167618887:web:c1b3c0bb26d441d1ebde62
android    1:11167618887:android:6dabc1d57506f19aebde62
ios        1:11167618887:ios:778b9d522a3df3bebd62
macos      1:11167618887:ios:778b9d522a3df3bebd62
windows    1:11167618887:web:86ed6c38c3307ceebde62

Learn more about using this file and next steps from the documentation:
> https://firebase.google.com/docs/flutter/setup
```

Figure 4.3.2.(B).1 Set up Firebase into Flutter project using command prompt

```
lib > firebase_options.dart > ...
1 // File generated by FlutterFire CLI.
2 // ignore_for_file: type=lint
3 import 'package:firebase_core/firebase_core.dart' show FirebaseOptions;
4 import 'package:flutter/foundation.dart'
5   show defaultTargetPlatform, kIsWeb, TargetPlatform;
6
7 /// Default [FirebaseOptions] for use with your Firebase apps.
8 ///
9 /// Example:
```

Figure 4.3.2.(B).2 A firebase\_option.dart is generated, confirming successful setup on application

```
9 dependencies:
10   flutter:
11     sdk: flutter
12   firebase_core: ^3.13.0
13   firebase_auth: ^5.5.2
14   cloud_firestore: ^5.6.6
15
PROBLEMS  AZURE  PORTS  DEBUG CONSOLE  TERMINAL  OUTPUT
Try 'flutter pub outdated' for more information.
PS C:\Users\wongw\vsCode_Projects\FYP\go_perak> flutter pub get
```

Figure 4.3.2.(B).3 Add firebase dependencies on pubspec.yaml using flutter terminal

```
lib > main.dart > main
1 import 'package:firebase_core/firebase_core.dart';
2 import 'package:flutter/material.dart';
3 import 'package:go_perak/firebase_options.dart';
4
5 Run | Debug | Profile
6 void main() async {
7   WidgetsFlutterBinding.ensureInitialized();
8   await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);
9   runApp(const MainApp());
10 }
```

Figure 4.3.2.(B).4 Initialize Firebase on application

### (C) Set up Firebase Authentication

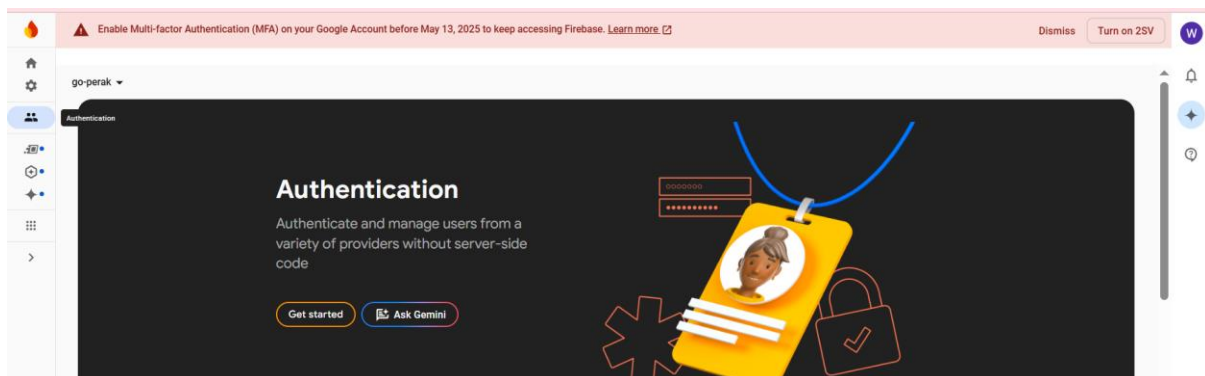


Figure 4.3.2.(C).1 Set up Firebase Authentication

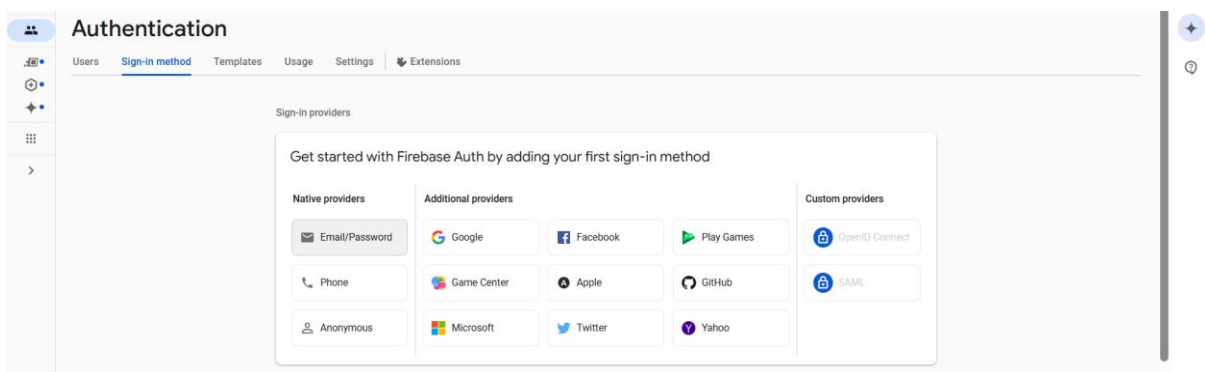


Figure 4.3.2.(C).2 Select sign in method

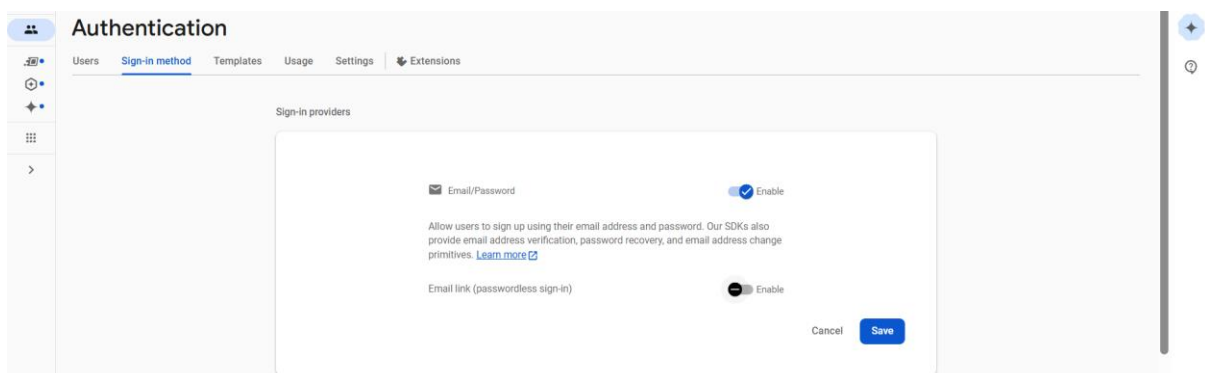


Figure 4.3.2.(C).3 Save sign in method

## CHAPTER 4 METHODOLOGY, TOOLS & SYSTEM SET UP

### (D) Set up Firebase Firestore

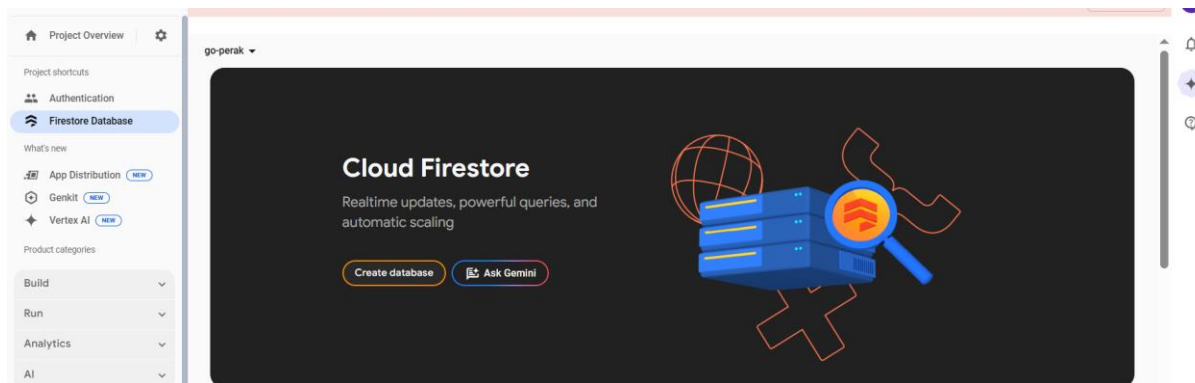


Figure 4.3.2.(D).1 Set up Firestore

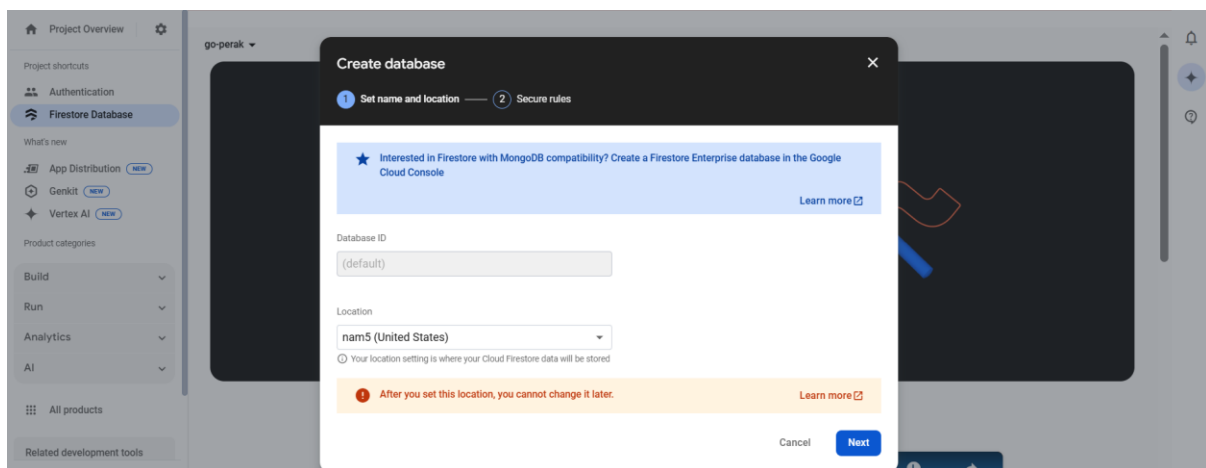


Figure 4.3.2.(D).2 Select Firestore location

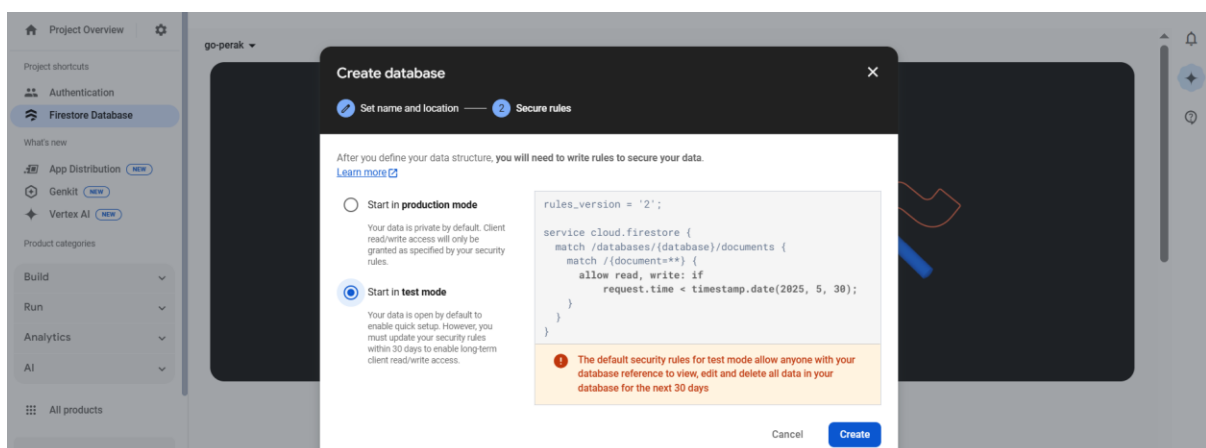


Figure 4.3.2.(D).3 Select test mode for development



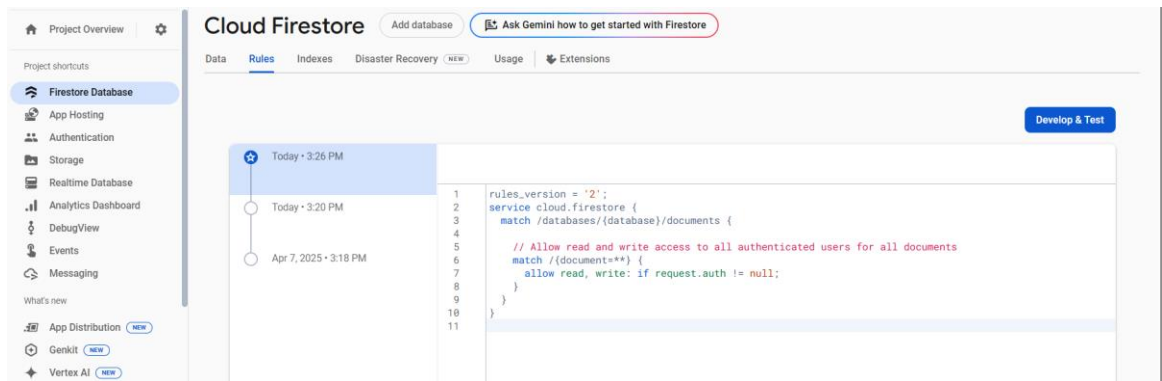


Figure 4.3.2.(D).4 Update security rules to allow only authenticated user to read and write the database

### 4.3.3 Supabase

#### (A) Set up Supabase

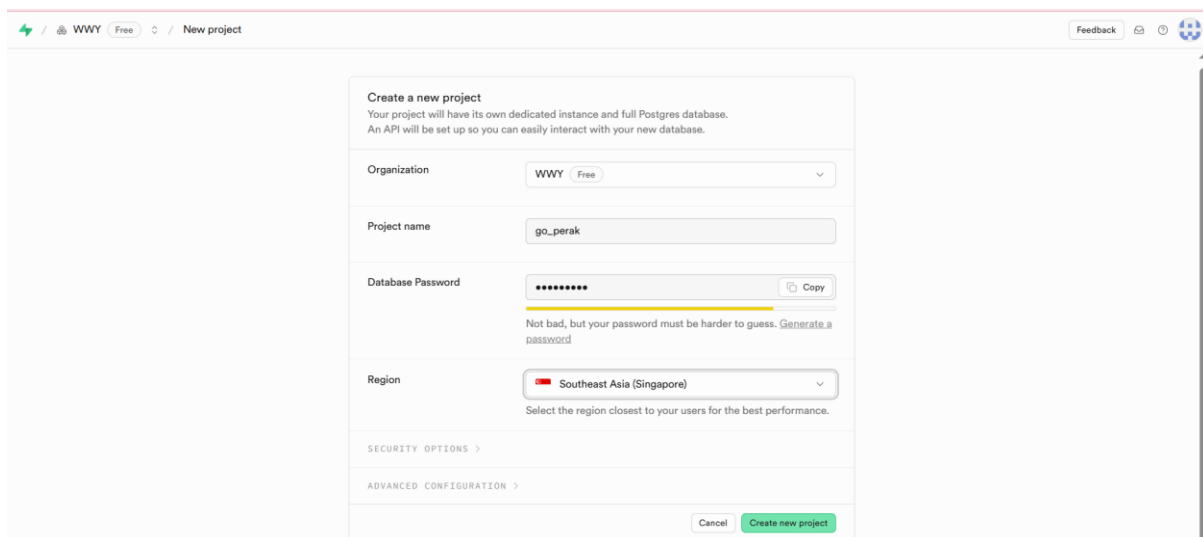


Figure 4.3.3.(A).1 Create new Supabase project

#### (B) Link Supabase to project

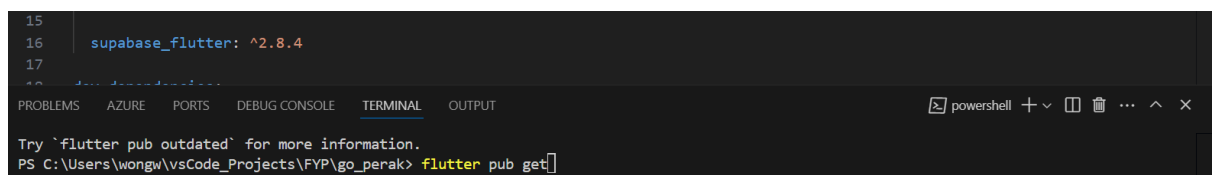
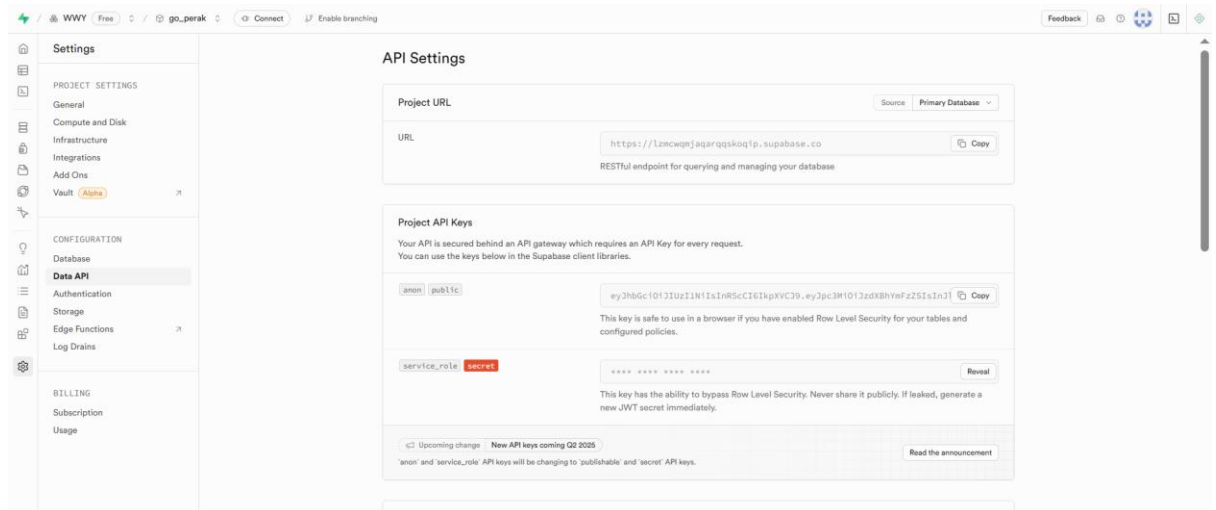


Figure 4.3.3.(B).1 Add Supabase dependency on Flutter

## CHAPTER 4 METHODOLOGY, TOOLS &amp; SYSTEM SET UP



### Figure 4.3.3.(B).2 Get Supabase anonKey

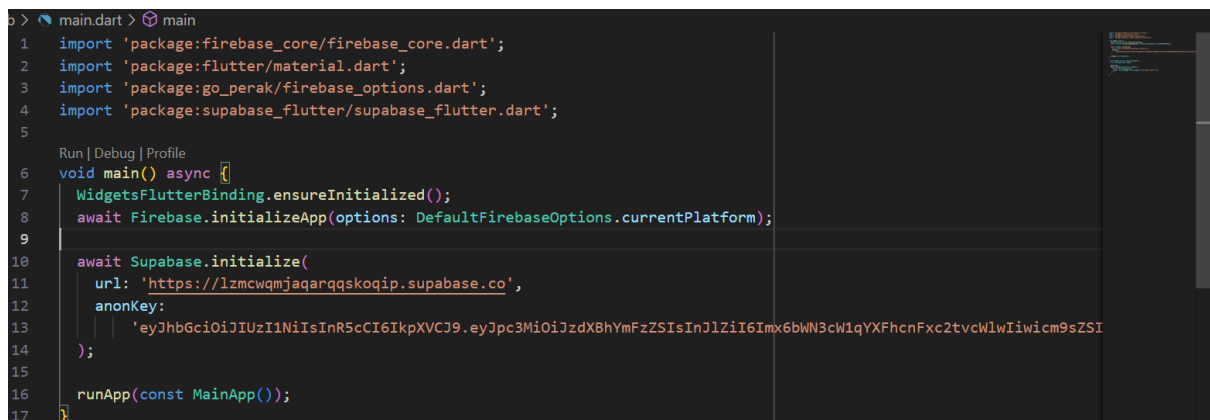


Figure 4.3.3.(B).2 Initialize Supabase on Flutter

### (c) Set up Supabase Storage

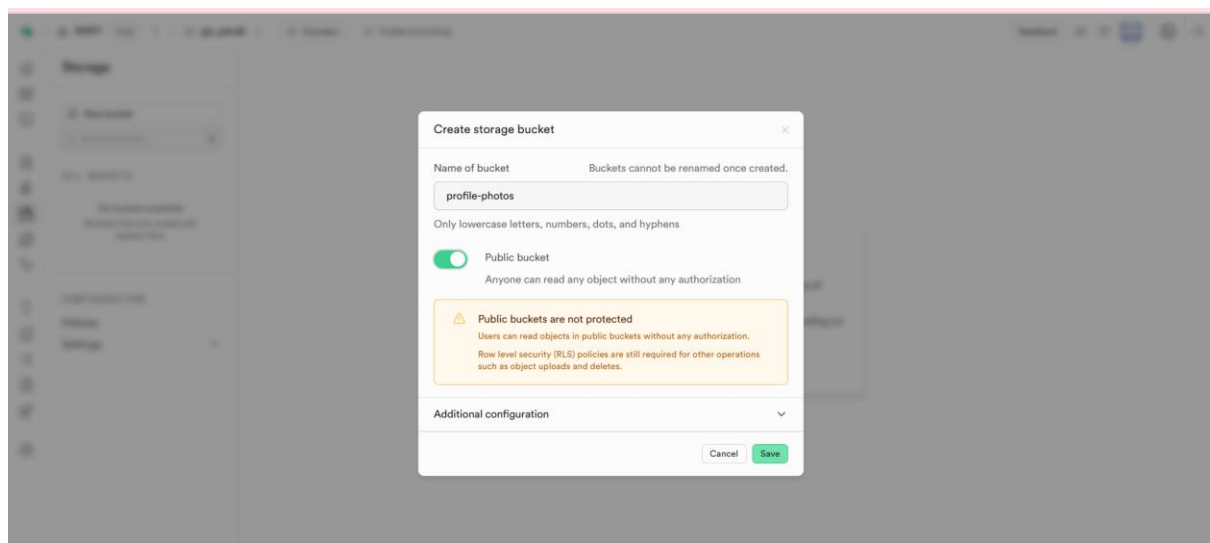


Figure 4.3.3.(C).1 Create storage bucket to store photos

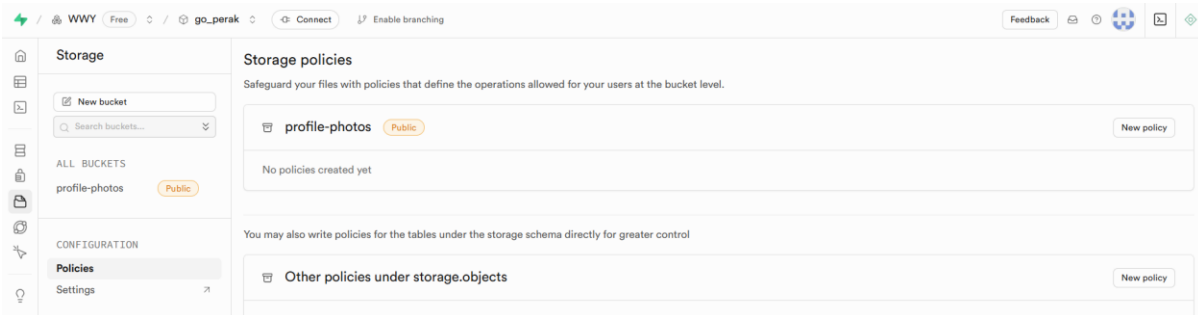


Figure 4.3.3.(C).2 Set up storage bucket's policies for CRUD operation

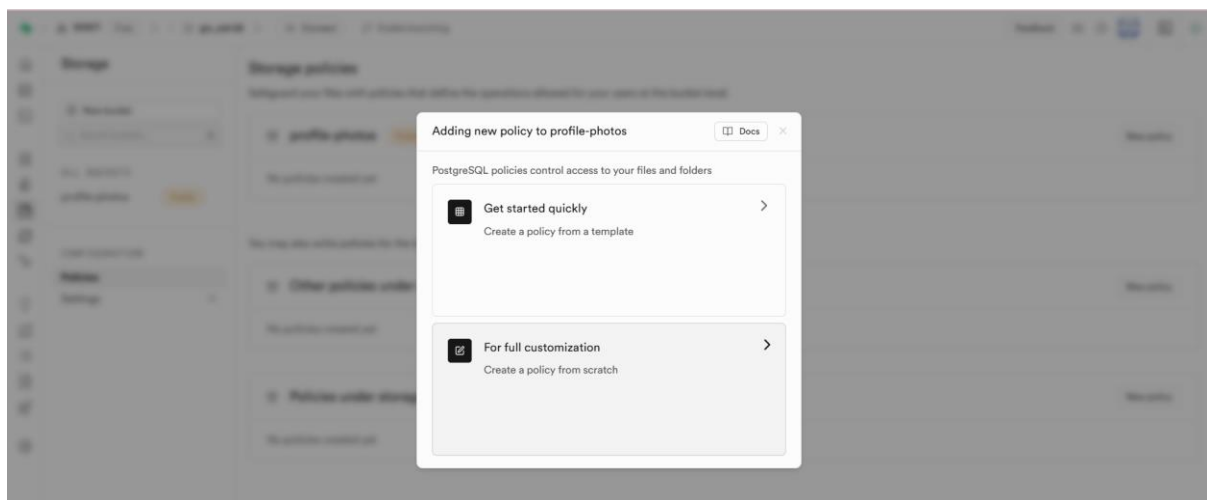


Figure 4.3.3.(C).3 Set up storage bucket's policies using provided customization option

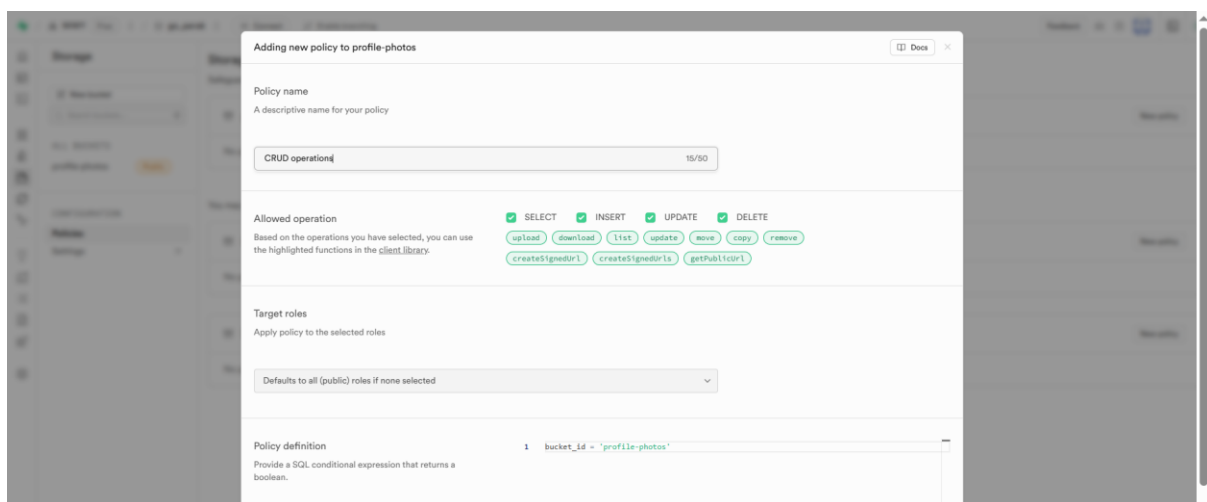


Figure 4.3.3.(C).4 Write new storage bucket policy

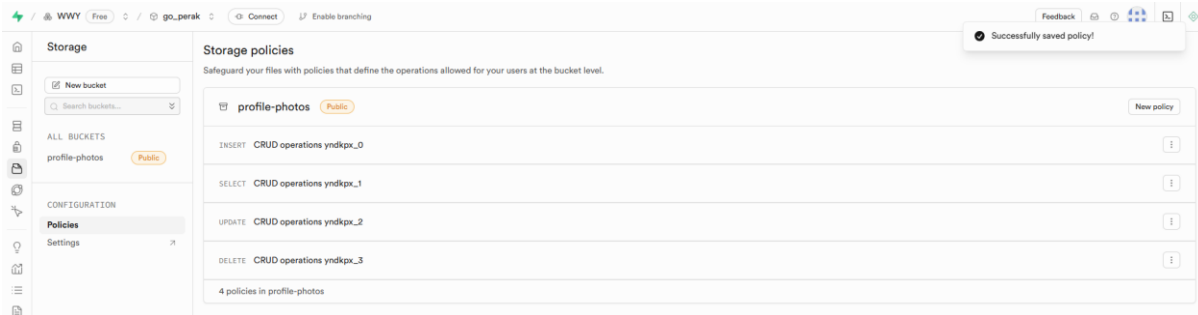


Figure 4.3.3.(C).5 Policy updated

### 4.3.4 Google Map Api

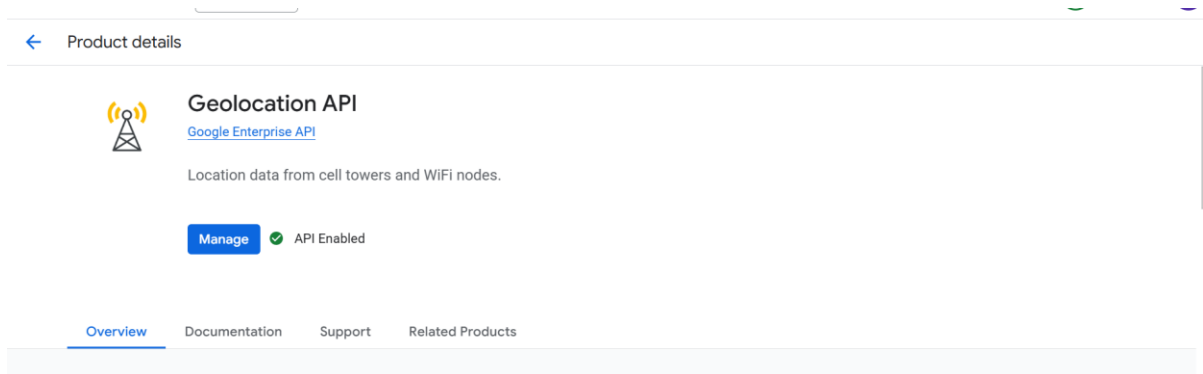


Figure 4.3.4.1 Enable geolocation api on google map platform console

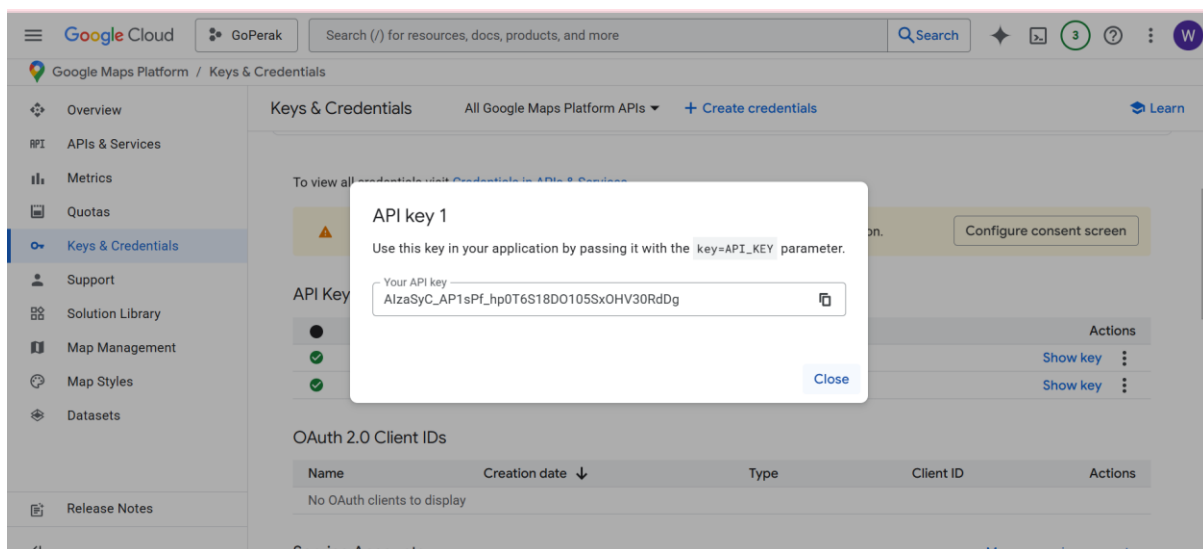


Figure 4.3.4.2 Get Google Map apiKey

```

1  <manifest xmlns:android="http://schemas.android.com/apk/res/android">
16     <application
21         <activity
46     </activity>
47
48     <meta-data android:name="com.google.android.geo.API_KEY"
49               android:value="AIzaSyC_AP1sPf_hp0T6S18D0105Sx0HV30RdDg"/>
50

```

Figure 4.3.4.3 Add Google Map apiKey in xml file

```

44  google_maps_flutter: ^2.12.1

```

Figure 4.3.4.4 Add google map dependency in pubspec.yaml file

```

5  // Helper method to get the destination Latlng from an address
6  Future<LatLng> getDestinationLatLng(String address, String apiKey) async {
7      final url = Uri.parse(
8          'https://maps.googleapis.com/maps/api/geocode/json?address=${Uri.encodeComponent(address)}&key=$apiKey',
9      );
10     final response = await http.get(url);
11     final data = jsonDecode(response.body);
12     final location = data['results'][0]['geometry']['location'];
13     return LatLng(location['lat'], location['lng']);
14 }

```

Figure 4.3.4.5 Sample Code to call API to use google map service

### 4.3.5 Face Comparison API

Face++ 旷视 Cognitive Services

Apps / API Key

+ Get API Key

App	API Key	API Secret	Type	Status	Operation
Emm's first...	bitNrUBohnf2fnKmk8Xvn8rTvNSw...	*****	Free	Active	<a href="#">View</a> <a href="#">Manage</a>

Figure 4.3.5.1 Get Face++ apiKey

```

final url = Uri.parse('https://api-us.faceplusplus.com/facepp/v3/compare');
final request = http.MultipartRequest('POST', url)
  ..fields['api_key'] = 'bitNrUBohnf2fnKmk8Xvn8rTvNSw-zgG'
  ..fields['api_secret'] = 'YN5Ka-3-G1MT7put0LAHhEglWSK0cnpr'
  ..files.add(
    await http.MultipartFile.fromPath('image_file1', icFaceCompressed.path))
  ..files.add(await http.MultipartFile.fromPath(
    'image_file2', liveFaceCompressed.path));

```

Figure 4.3.5.2 Sample code to call API for face comparison using http with apiKey

Chapter 5 Implementation

5.1 Database Design

5.1.1 Firebase Authentication

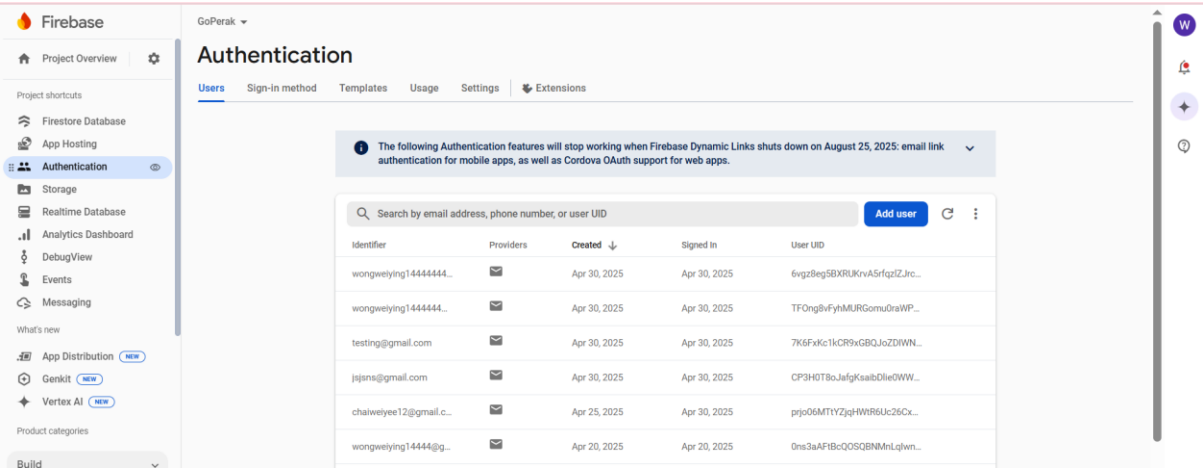


Figure 5.1.1 Firebase Authentication Database

## 5.1.2 Firebase Firestore

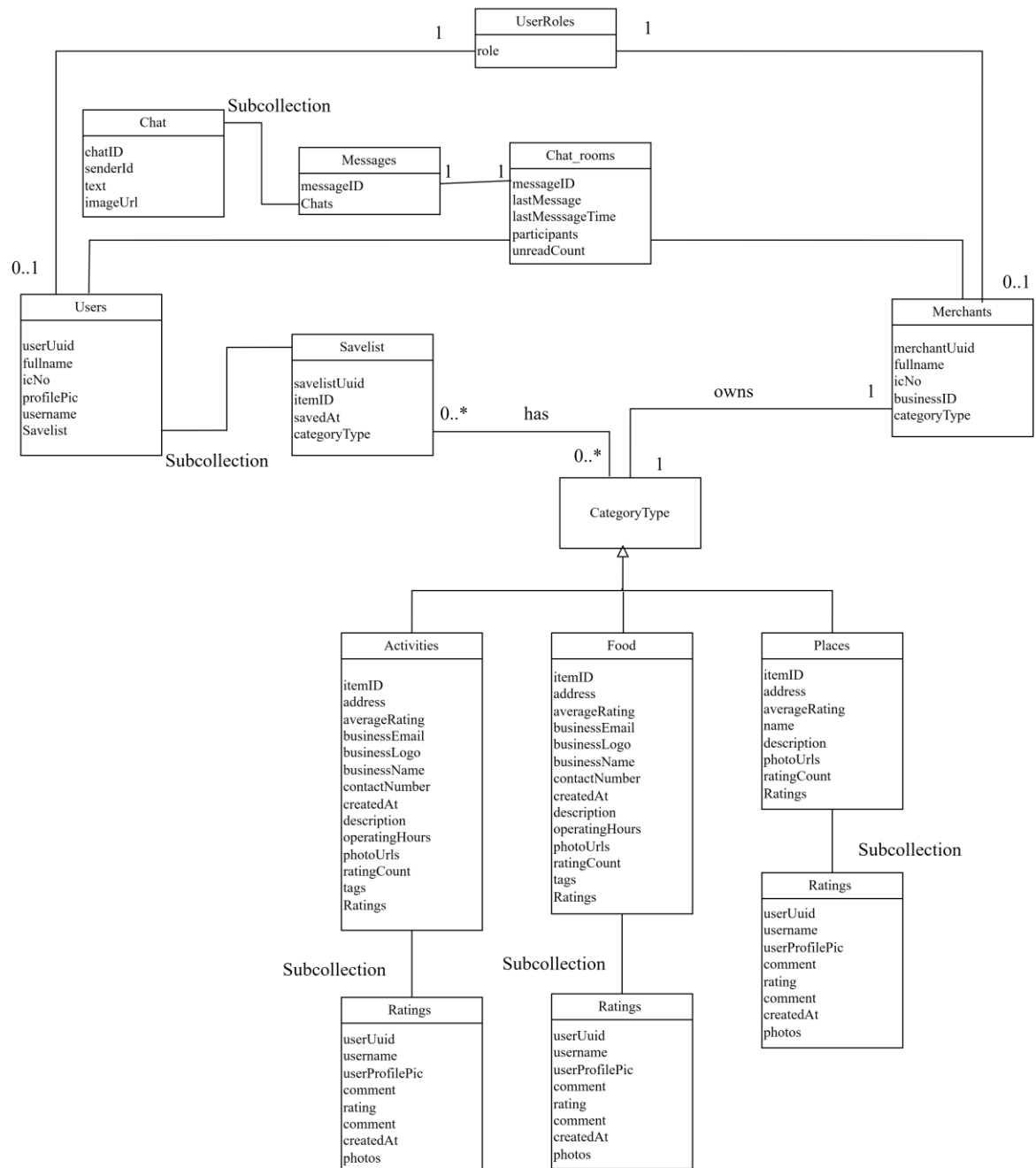


Figure 5.1.2 Overview of Firebase Firestore Database Design

## CHAPTER 5 IMPLEMENTATION

The Firestore database is structured to manage a role-based access system, where users are categorized as either Users or Merchants, determined by the UserRoles table. Upon login, the system queries the UserRoles collection using the unique identifier (UUID). If the role is identified as "Users", the application retrieves personal details from the Users collection, including attributes like profilePic, icNo, fullname, and username. These users are then directed to the user-specific home page. Additionally, each user has a Savelist subcollection that records saved item references. Each entry in the Savelist includes a unique ID, a timestamp (savedAt), and a categoryType, which specifies whether the item belongs to Food, Places, or Activities.

To retrieve saved item details, the system uses the categoryType and itemID stored in the Savelist to fetch the corresponding document from the relevant collection (e.g., Food, Places, or Activities). Each of these category collections maintains business-related information such as address, description, businessEmail, contactNumber, photoUrls, and user-generated metadata like averageRating, ratingCount, and tags. Importantly, every business item includes a Ratings subcollection. This stores user reviews, the review comment, numerical rating, optional review photos, and the creation timestamp. When a new review is created, the system recalculates the average rating, updates the averageRating field, and increments the ratingCount by one.

For merchants, the system follows a similar logic. Upon identifying a user as a "Merchants" role, it fetches data from the Merchants collection. Each merchant record includes attributes such as merchantUuid, fullname, icNo, and critically, the businessID and associated categoryType. These two fields are used to query the correct business collection (e.g., Food, Places, or Activities) and retrieve the corresponding business data. This setup ensures that the system dynamically supports multi-category business listings and enables merchants to manage only their specific business data.

In the chat system, the Messages collection stores each conversation, and the individual messages within a conversation are stored in a Chat subcollection. Each conversation is represented by a document in the Chat\_rooms collection, which tracks the participants, the last message, and the unread message count.



## CHAPTER 5 IMPLEMENTATION

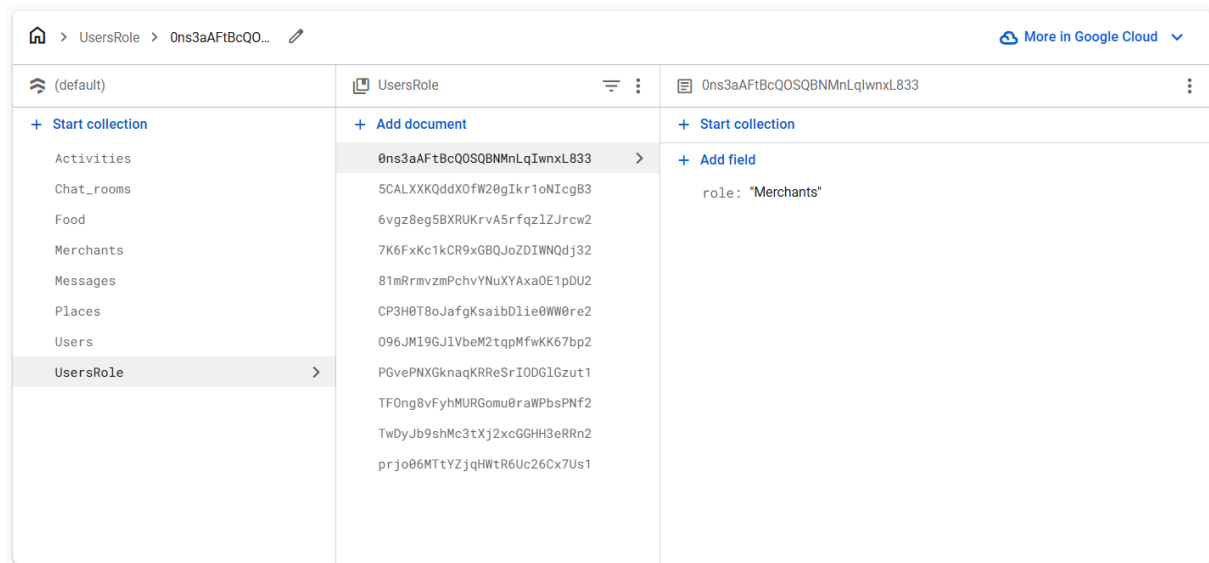


Figure 5.1.2.1 UsersRole Collection

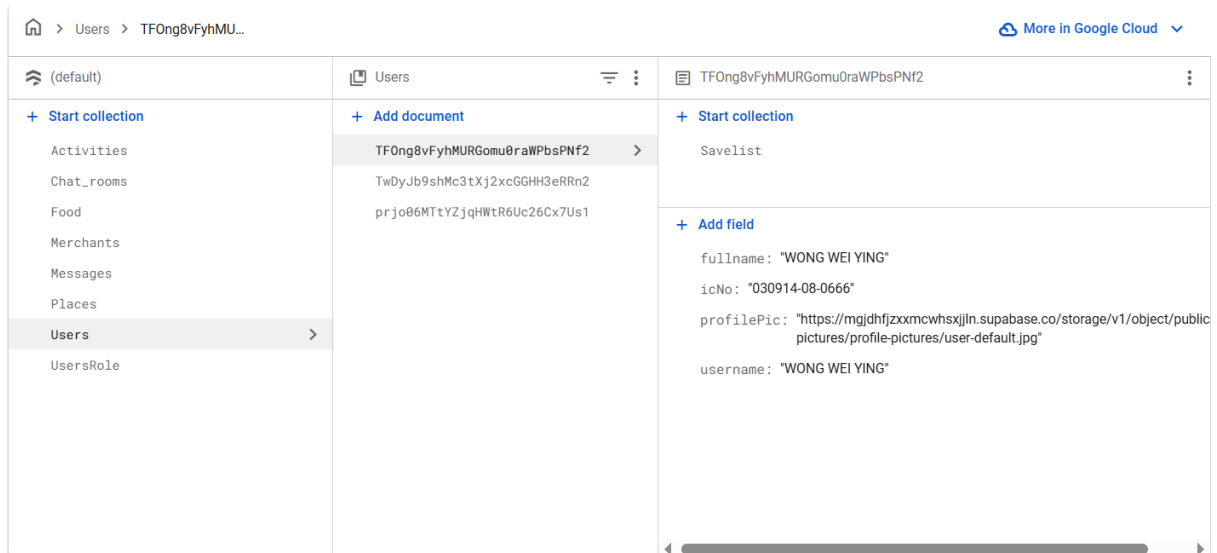


Figure 5.1.2.2 User Collection

## CHAPTER 5 IMPLEMENTATION

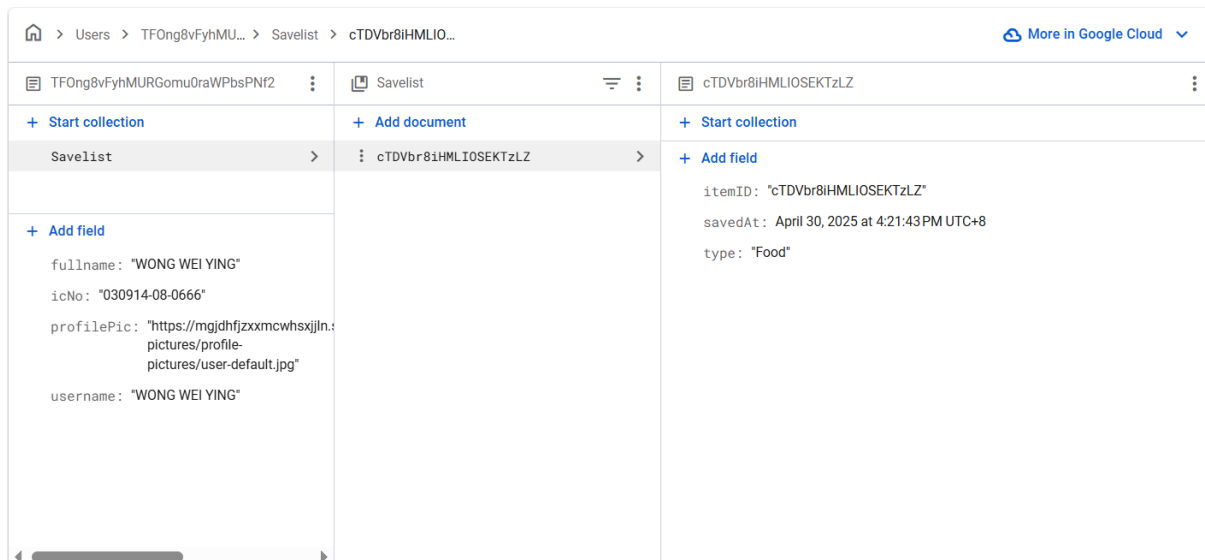


Figure 5.1.2.3 Savelist Sub-Collection from User Collection

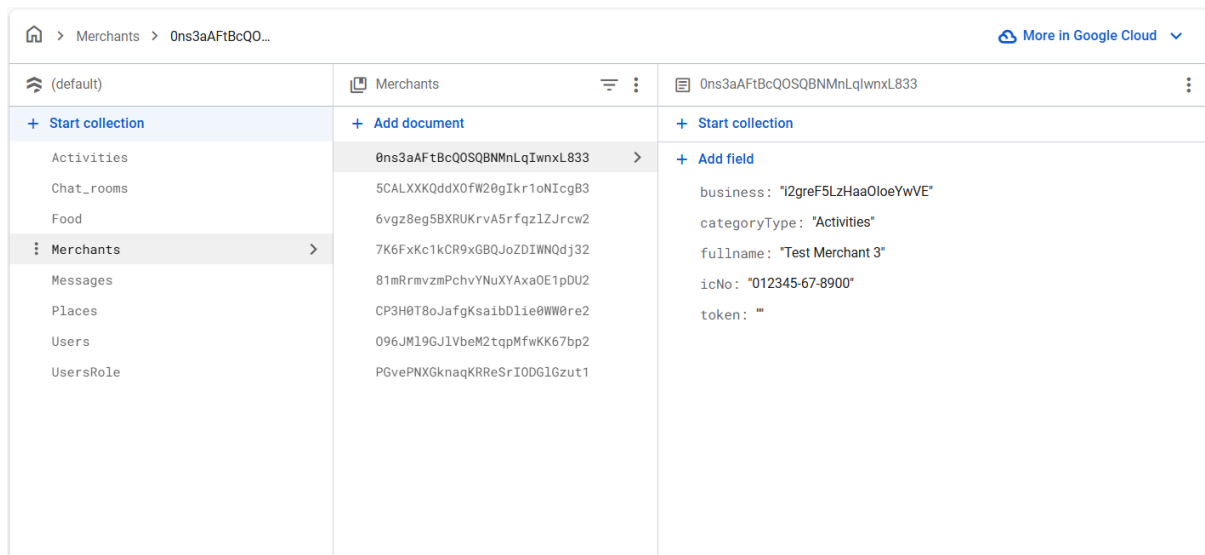


Figure 5.1.2.4 Merchant Collection

## CHAPTER 5 IMPLEMENTATION

🏠 > Places > 51TTyeu1Lr9X...			☁ More in Google Cloud
⚙ (default)	📁 Places	📄 51TTyeu1Lr9XmckHXL4M	⋮
+ Start collection	+ Add document	+ Start collection	
Activities	51TTyeu1Lr9XmckHXL4M >	Ratings	
Chat_rooms	5boU8upZduGNe1ydMs1A		
Food	EmkCugJ1bqxz4N7wkFUz		
Merchants	bSNAmM9rT4HFP2qRpFXX		
Messages	zEWrtQVG1pJc7575DECR		
Places >		+ Add field	
Users		address: "Batu Gajah, Perak"	
UsersRole		averageRating: 5	
		description: "An unfinished mansion built by Scottish planter William Kellie-Smith, offering a glimpse into colonial history and rumored to be haunted."	
		image: "https://3.bp.blogspot.com/-cuv4BL93Ny/UU8tEj-pD2I/AAAAAAAAAHtE/_cQdx0cVpS4/s1600/130317-Kellie's+Castle+001.jpg"	
		name: "Kellie's Castle"	
		ratingCount: 1	

Figure 5.1.2.5 Places Collection

🏠 > Food > 3eNeIFWm92Qi...			☁ More in Google Cloud
⚙ (default)	📁 Food	📄 3eNeIFWm92QiNBP5jv40	⋮
+ Start collection	+ Add document	+ Start collection	
Activities	3eNeIFWm92QiNBP5jv40 >	+ Add field	
Chat_rooms	cTDVbr8iHMLIOSEKTzLZ	address: "No1, Jalan 1, 34500, Batu Kurau, Perak"	
Food >	kvSh8e4QQQR1ZoPJcSz6	averageRating: 0	
Merchants	nYs1N77LBenAZJ0H5uHO	businessEmail: "wongweiyiing1444444@gmail.com"	
Messages	sNBevQAzgdTdfA8hau49	businessLogo: "https://mgjdhfjzxxmcwhsxjln.supabase.co/storage/v1/object/public/logo-photos/business-photos/6vgz8eg5BXRUKrvA5rfqzLJrcw2/1746001763994.webp"	
Places	xUKsuENStHvj8NgFwfVR	businessName: "Kfc"	
Users		contactNumber: "0169958647"	
UsersRole		createdAt: April 30, 2025 at 4:29:27 PM UTC+8	
		description: "Testingg"	
		isApproved: false	
		operatingHours: "Testingg"	
		photoUrls	
		0 "https://mgjdhfjzxxmcwhsxjln.supabase.co/storage/v1/object/public/business-photos/business-photos/6vgz8eg5BXRUKrvA5rfqzLJrcw2/1746001765110.webp"	
		1 "https://mgjdhfjzxxmcwhsxjln.supabase.co/storage/v1/object/public/business-photos/business-photos/6vgz8eg5BXRUKrvA5rfqzLJrcw2/1746001765642.jpg"	
		ratingCount: 0	
		tags	

Figure 5.1.2.6 Food Collection

## CHAPTER 5 IMPLEMENTATION

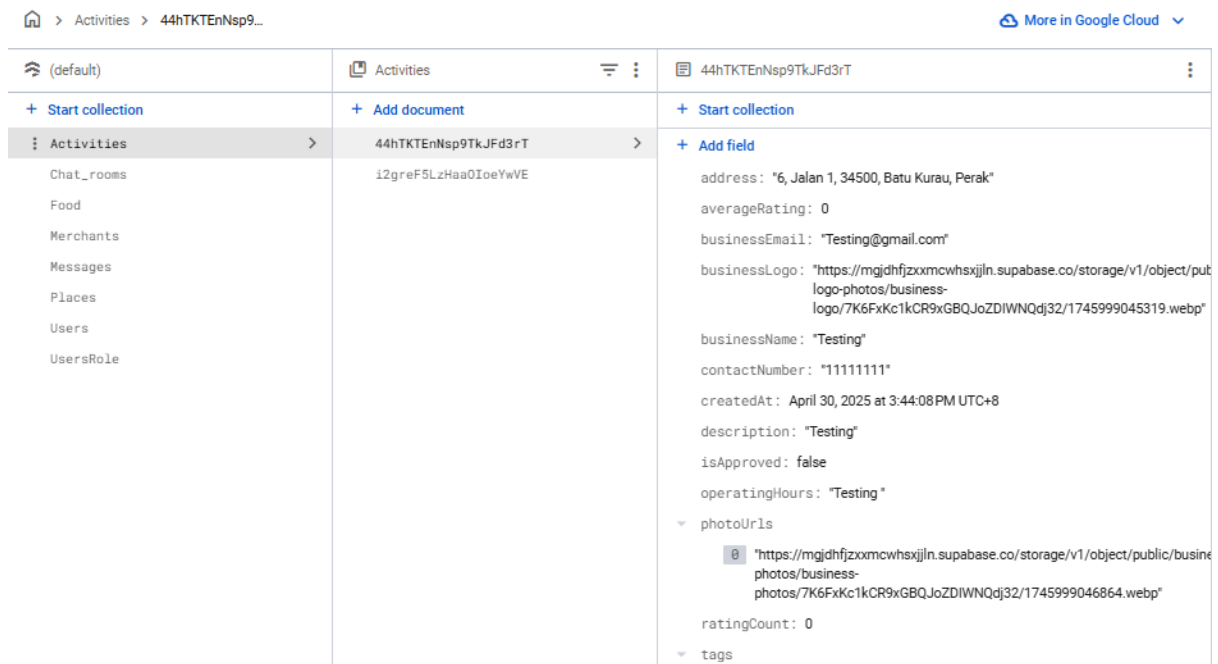


Figure 5.1.2.7 Activities Collection

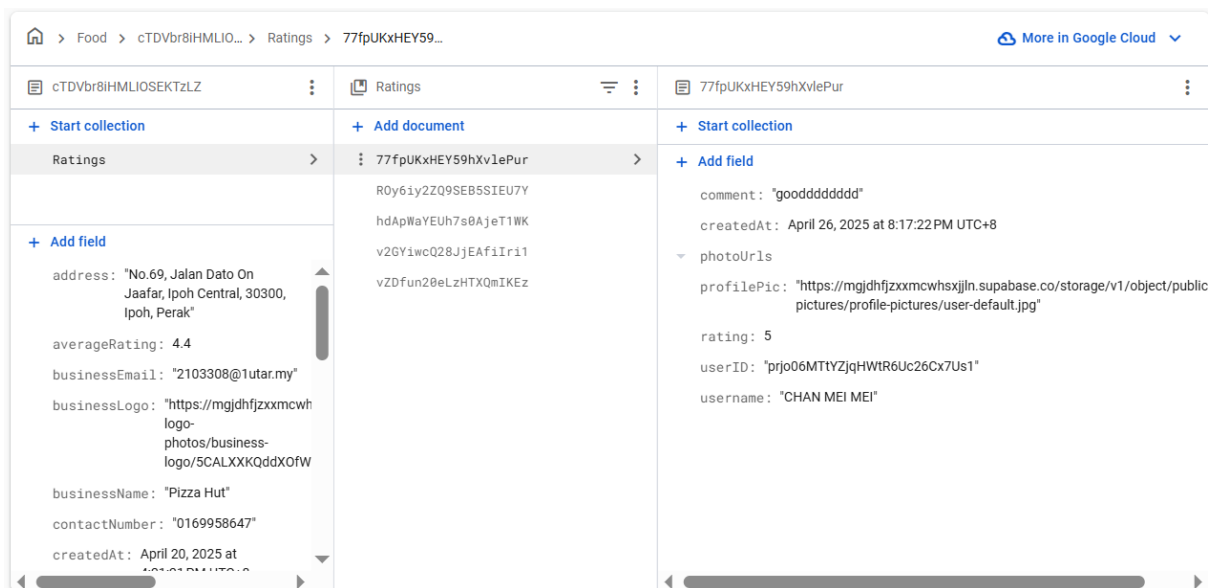


Figure 5.1.2.8 Rating Subcollection from Food Collection

## CHAPTER 5 IMPLEMENTATION

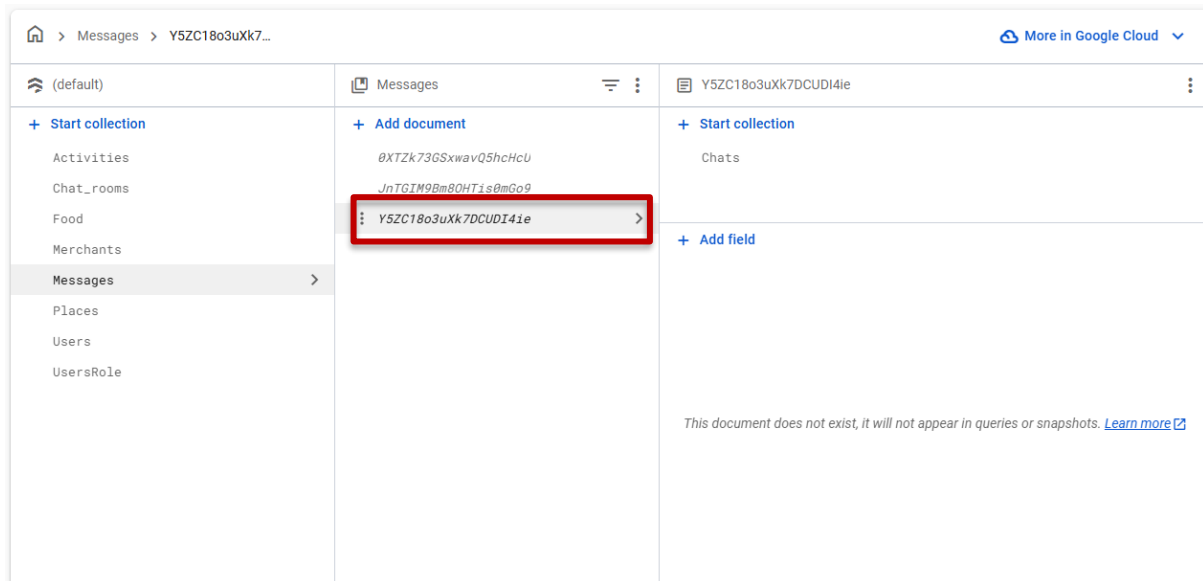


Figure 5.1.2.9 Messages Collection

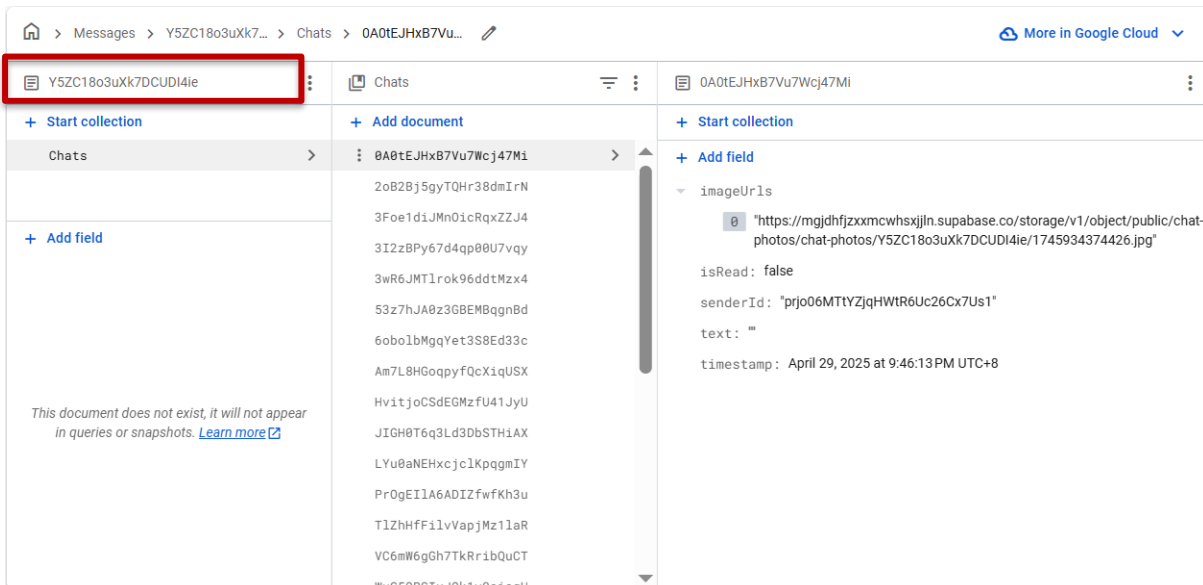


Figure 5.1.2.10 Chats Collection

CHAPTER 5 IMPLEMENTATION

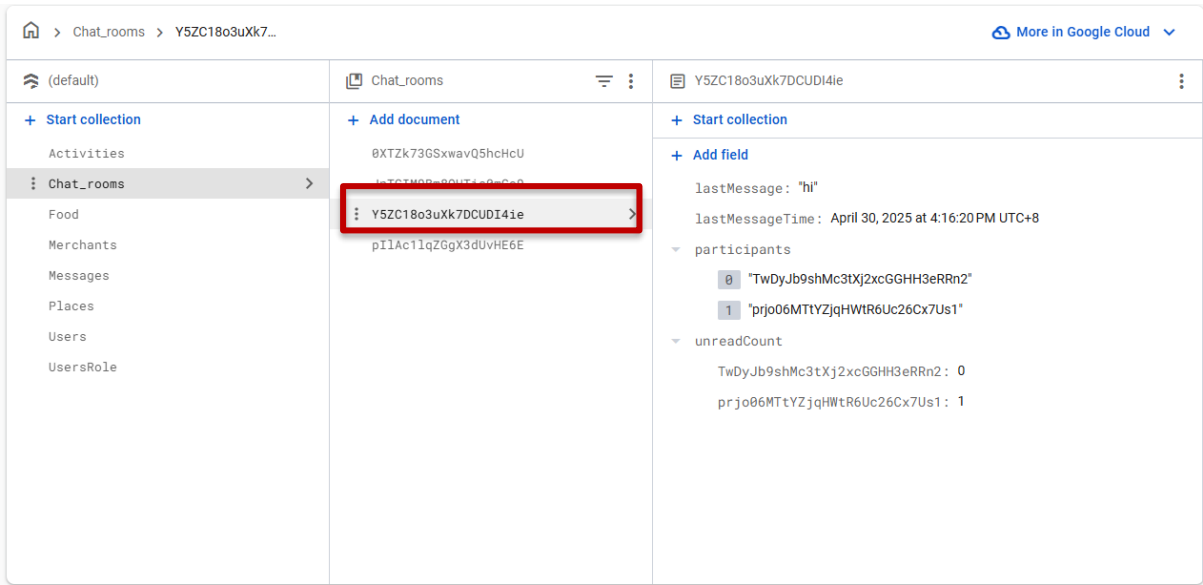


Figure 5.1.2.11 Chat rooms Collection

5.1.3 Supabase Database

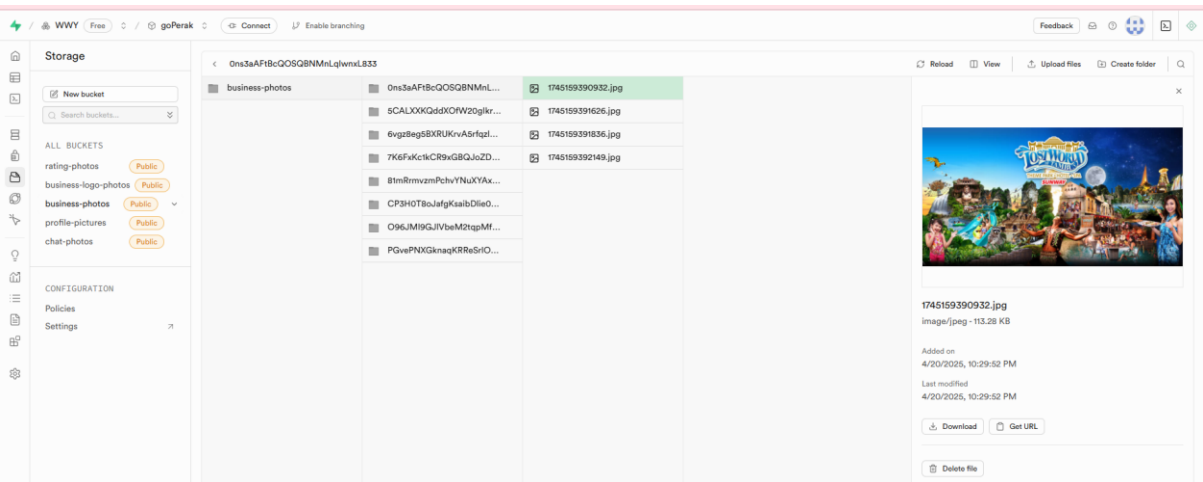


Figure 5.1.3 Supabase Stroage Bucket

### 5.2 User Interface Design

#### 5.2.1 Login Page

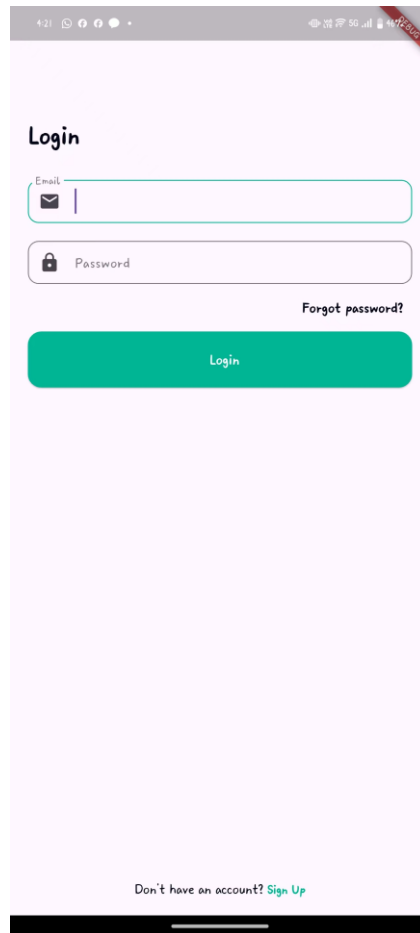


Figure 5.2.1 Login Page

Before sending the authentication request, the system first validates the format of the email address. Once the format is verified, the system proceeds to authenticate the user using Firebase Authentication. If the authentication is successful, it then checks the user's role by querying the UserRoles collection using the user's UUID. If the role is "Users", the user is redirected to the user home page. If the role is "Merchants", the user is redirected to the merchant home page.

### 5.2.2 Select Role Page

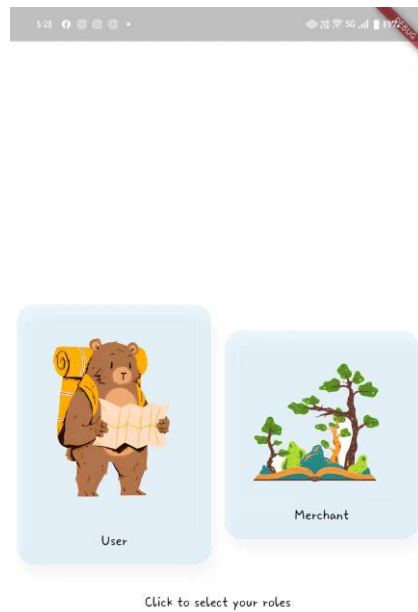


Figure 5.2.2 Select Role Page

Select role is the first step of the sign up process. It is for the user to select their role — either "User" or "Merchant". This selection determines the subsequent registration steps and form requirements.



### 5.2.3 Disclaimer Page

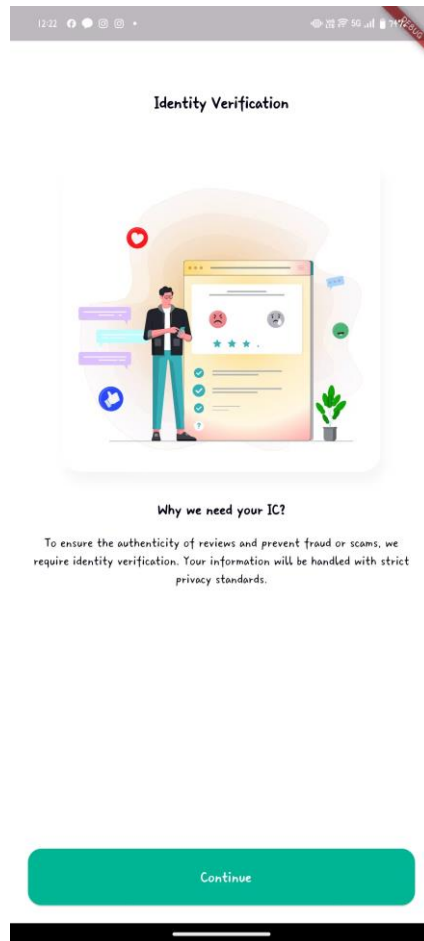


Figure 5.2.3 Disclaimer Page

The user is presented with a disclaimer explaining that the app requires identity verification using an IC. This is necessary to prevent fake accounts and reviews.

### 5.2.4 Scan IC Page

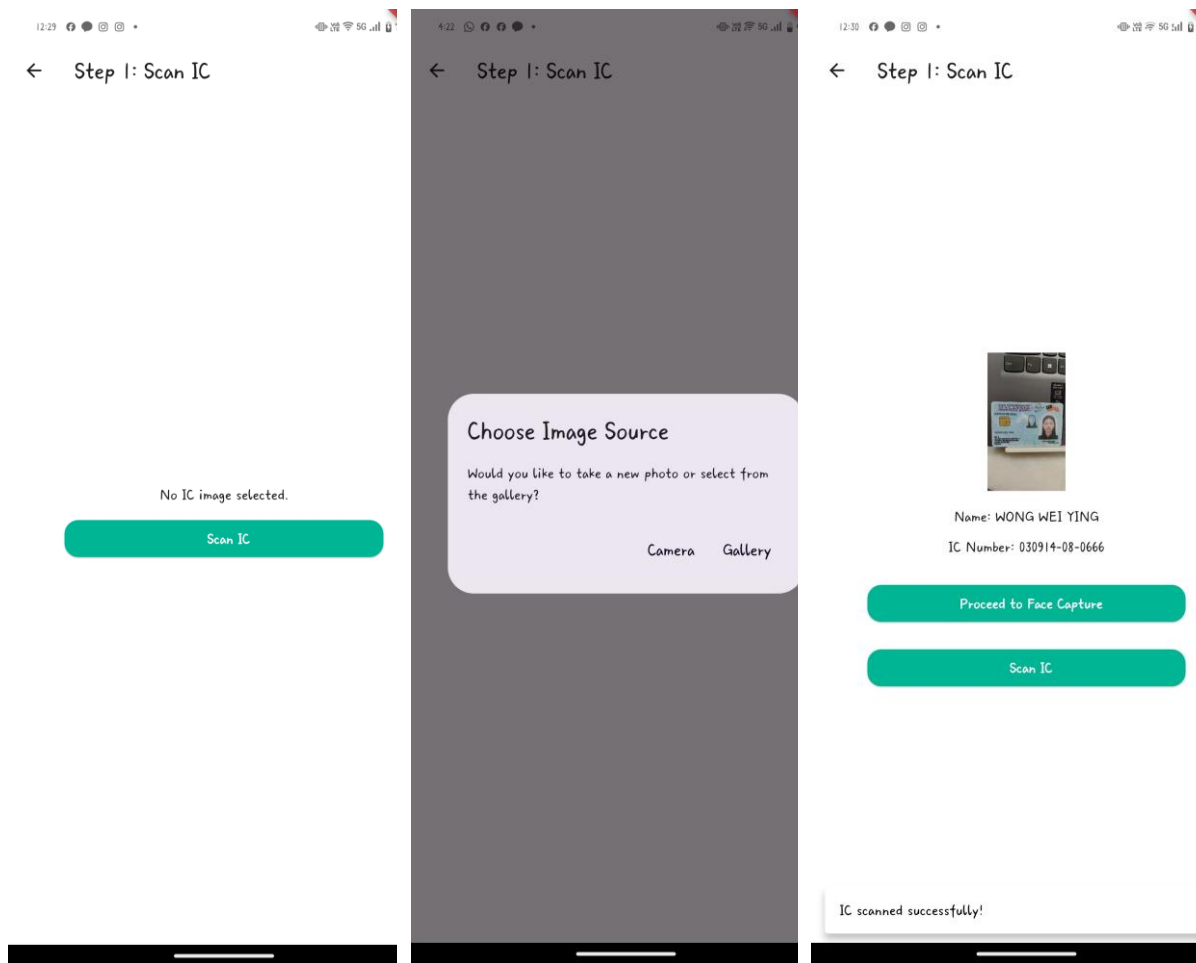


Figure 5.2.4.1 Scan IC Page

Figure 5.2.4.2 Upload IC  
options on the Scan IC Page

Figure 5.2.4.3 Successful IC  
Scan on the Scan IC Page

The user can upload their IC image either from the gallery or by using the camera. The system then extracts the full name and IC number from the image using text recognizer. The extracted details are shown to the user for confirmation. If the information is correct, the user proceeds to the next step.

### 5.2.5 Face Comparison Page



Figure 5.2.5.1 Analyze user's face on the Face Comparison Page

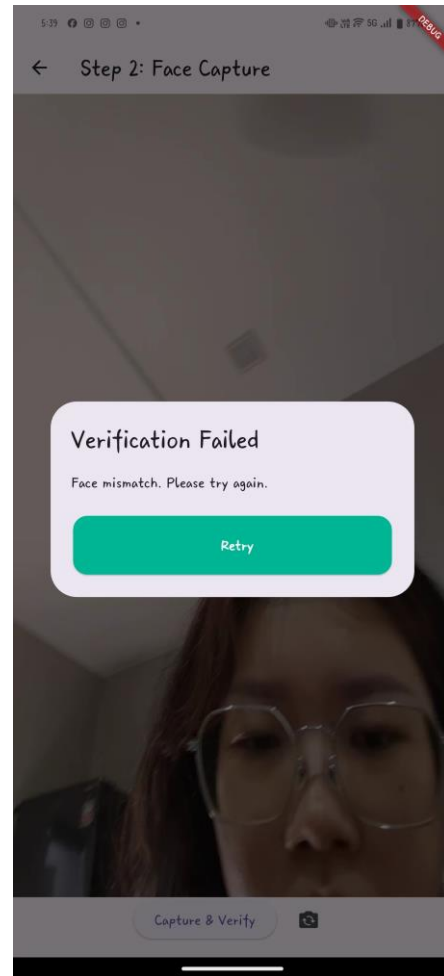
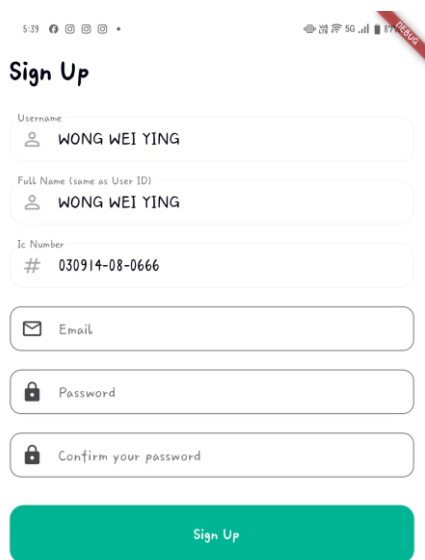


Figure 5.2.5.2 Verification fail on the Face Comparison Page

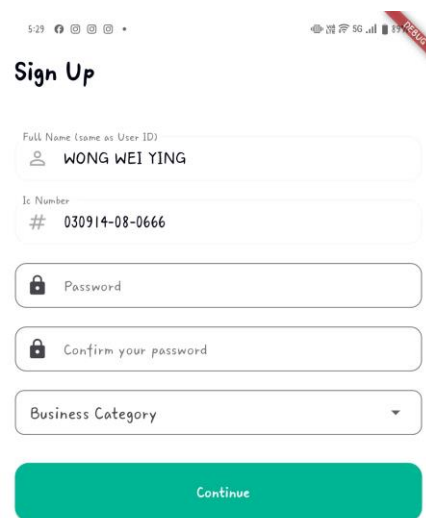
The user is prompted to scan their face using the live camera. If the face matches the IC data, the process continues. If the comparison fails, the user is asked to retry.

## 5.2.6 Sign Up Page



The image shows a mobile app interface for a user sign-up page. At the top, the status bar shows the time as 5:29 and various icons. The page title is "Sign Up". Below the title, there are six input fields: "Username" (containing "WONG WEI YING"), "Full Name (same as User ID)" (containing "WONG WEI YING"), "Id Number" (containing "# 030914-08-0666"), "Email", "Password", and "Confirm your password". Each field has a small icon to its left. At the bottom, there is a large green button labeled "Sign Up".

Figure 5.2.6.1 Sign Up Page for User



The image shows a mobile app interface for a merchant sign-up page. At the top, the status bar shows the time as 5:29 and various icons. The page title is "Sign Up". Below the title, there are five input fields: "Full Name (same as User ID)" (containing "WONG WEI YING"), "Id Number" (containing "# 030914-08-0666"), "Password", "Confirm your password", and "Business Category" (a dropdown menu). Each field has a small icon to its left. At the bottom, there is a large green button labeled "Continue".

Figure 5.2.6.2 Sign Up Page for Merchant

If the selected role was "Merchant" in the initial step, the user is directed to an additional form to provide business details. Else, the registration process is completed at this point. The system first validates the email format and ensures the password is at least six characters long. Once the inputs pass validation, a sign-up request is sent to Firebase Authentication to create the account. Firebase then checks whether the email address is already registered. After completing registration, the user is redirected to Login Page.

## 5.2.7 Merchant Business Detail Page

**Merchant Details**

Business Name: Testing

Registration Number: T001

Business Email: wongweiyng14@gmail.com

Contact Number: 111111

Description: Testing

Operating Hours: Testing

Address: No 1, Jalan I

Postcode: 34900

City: Pantai Remis

State: Perak

Country: Malaysia

Choose Tags: Western, Malay, Chinese

Upload your business logo

Upload your business photos

Sign Up

Figure 5.2.7 Merchant Business Detail Page

The merchant fills in their business details. A sign up request is sent to Firebase Authentication to check if the email address is registered. The business logo and business photos are then uploaded to Supabase storage. A new user account is created on Firebase Authentication and Firestore. The photo urls returned from Supabase is inserted into the Firestore. Once the sign up done, it will redirect user to Login Page.

## 5.2.8 Reset Password Page

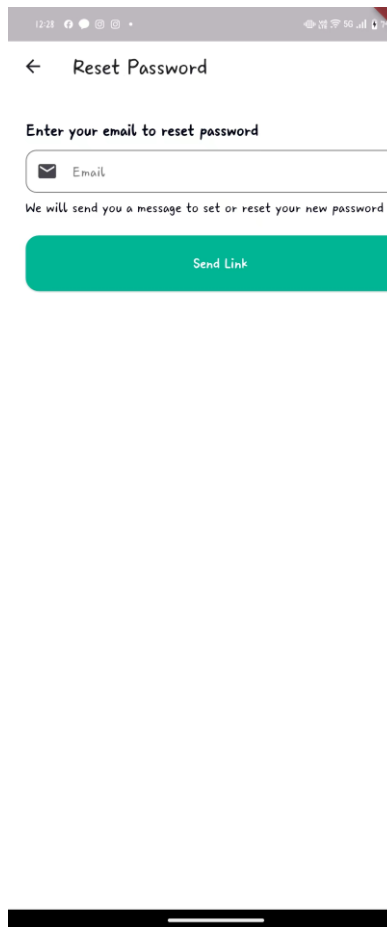


Figure 5.2.8.1 Reset Password Page

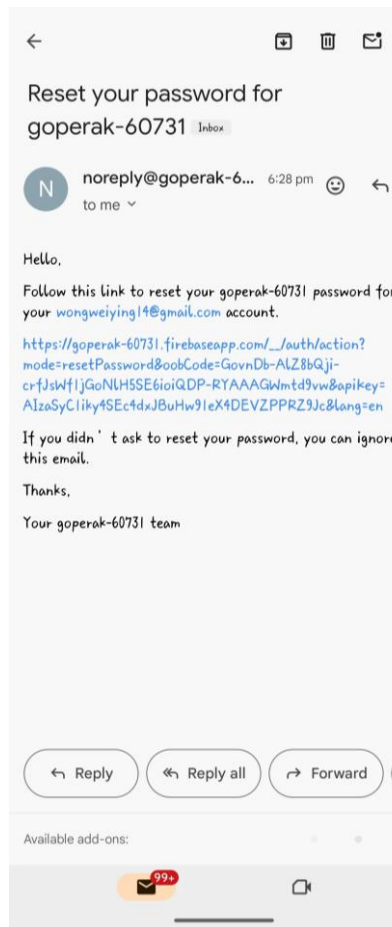


Figure 5.2.8.2 Reset Password Link Sent Via email

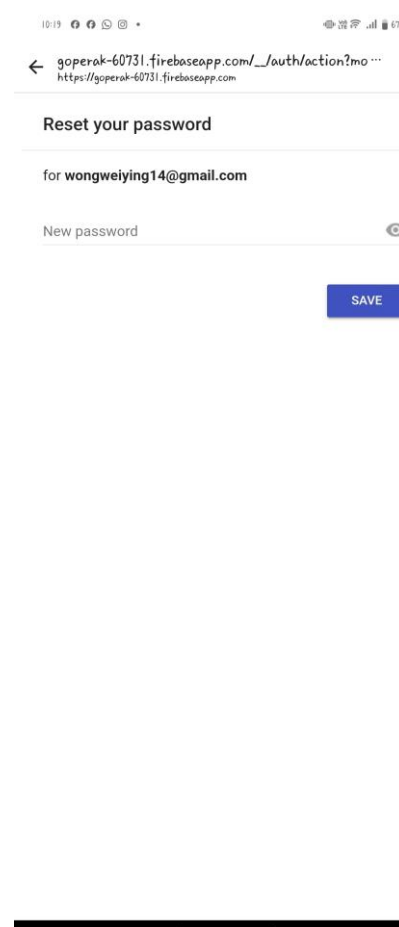


Figure 5.2.8.3 Reset Password with Reset Password Link

The user enters their email address on the Reset Password page. The system sends a password reset link to that email. The user can then click the link to open a secure page where they can set a new password.

## 5.2.9 User Home Page

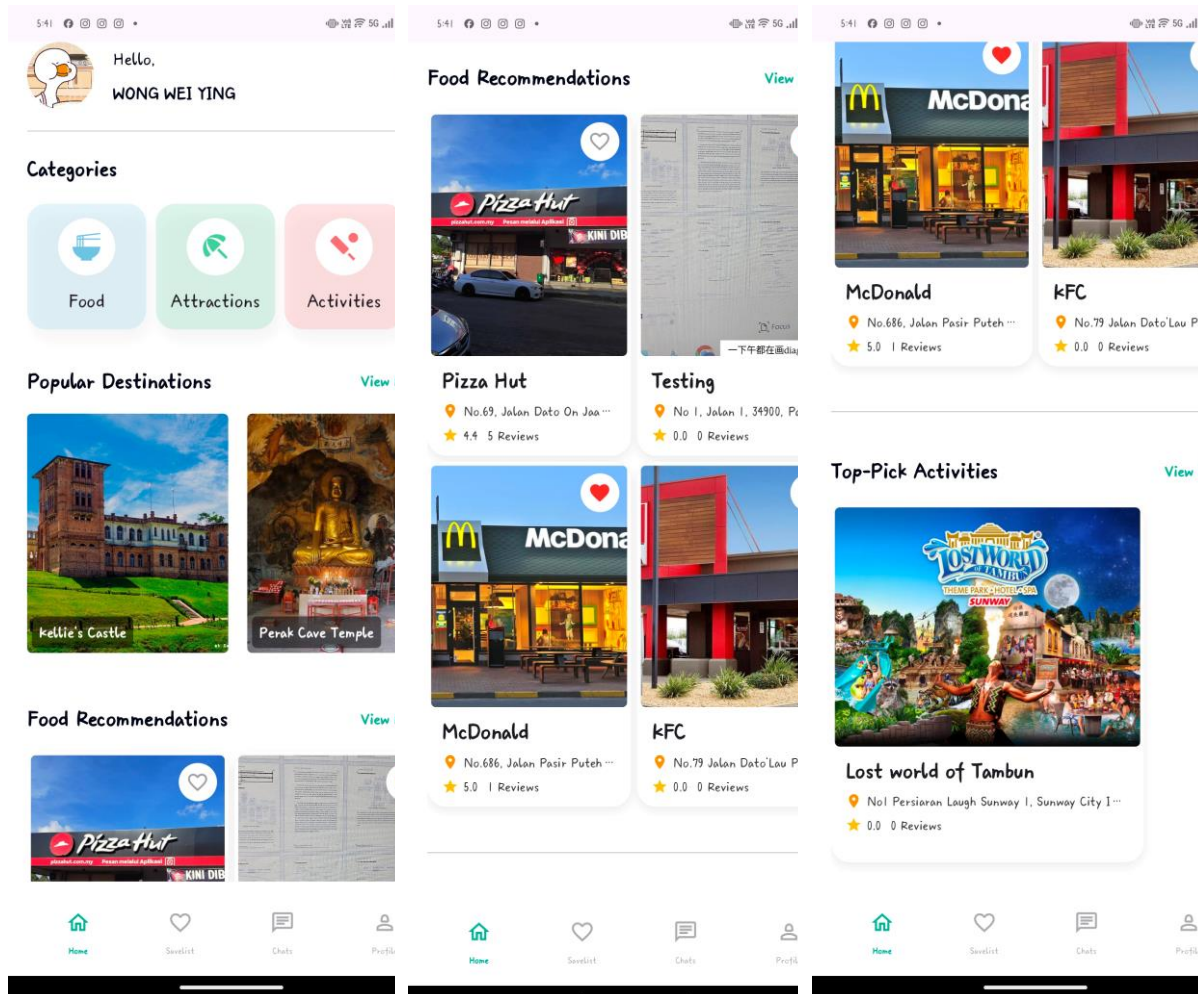


Figure 5.2.9 User Home Page

The User Home Page displays a selection of popular destinations, food recommendations, and suggested activities. Users can explore more items within each category by clicking on the category card or the "View More" button, which will navigate them to the corresponding Category List Page. Tapping the Search icon in the top navigation bar takes the user to the Search Page, where they can look up specific places, food, or activities. A bottom navigation bar is available, providing quick access to the Savelist, Chat List, and Profile Page for seamless user interaction and navigation.

### 5.2.10 Savelist Page

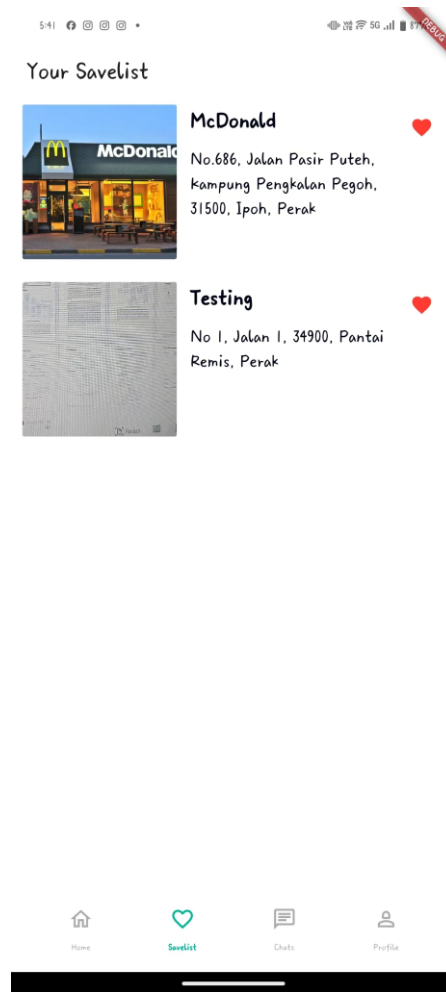


Figure 5.2.10 Savelist Page

The Savelist Page displays all items the user has saved, Each saved item shows a preview, including its name, image and address. When the user taps on a saved item, it will direct user to the item detail page.



### 5.2.11 Food List Page

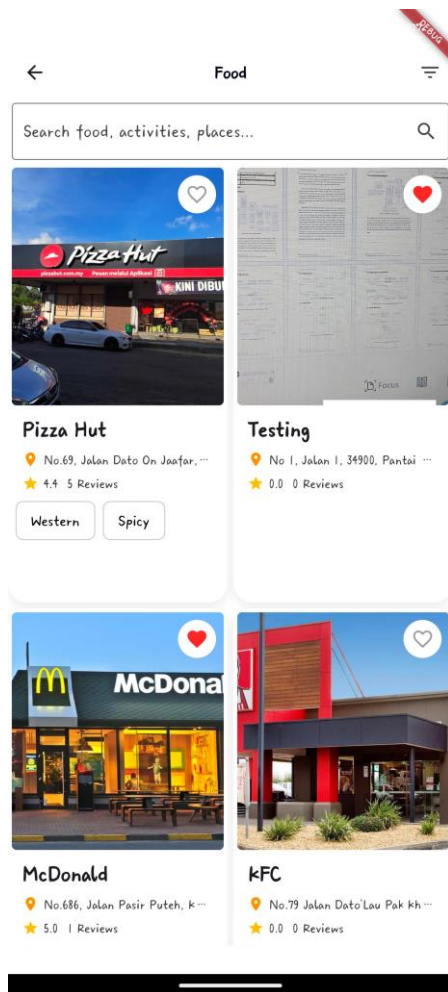


Figure 5.2.11.1 Food List Page

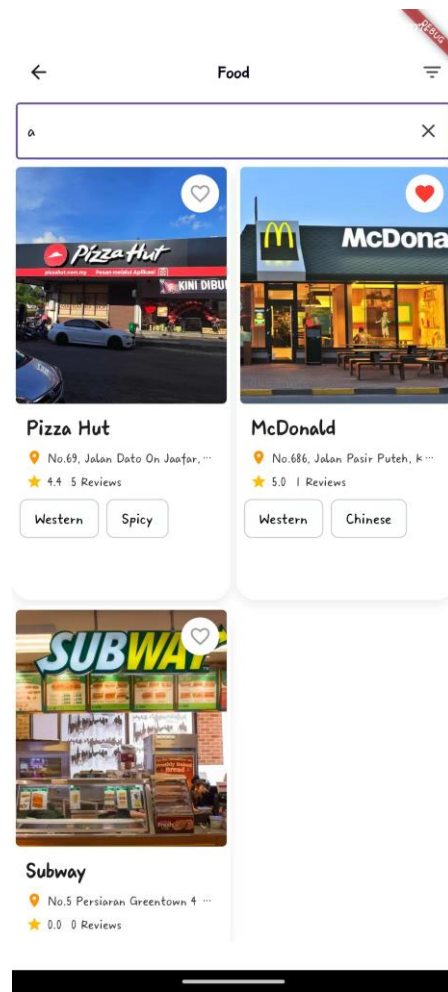


Figure 5.2.11.2 Search Feature on Food List Page

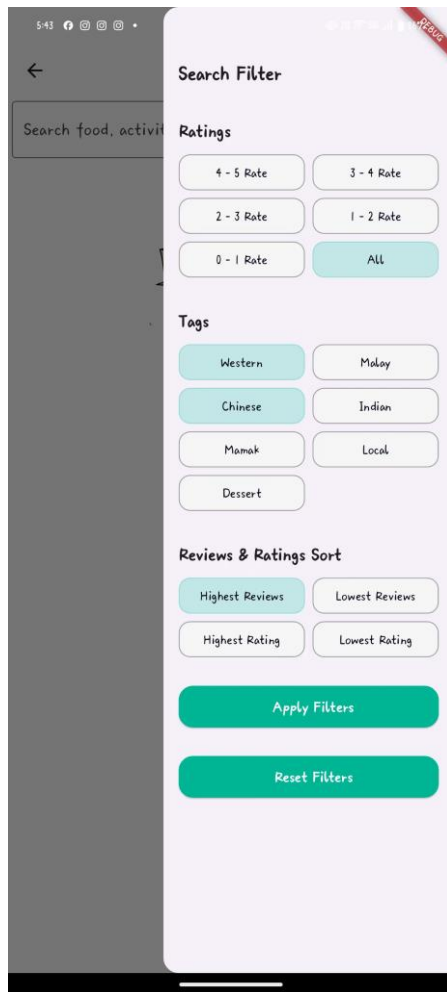


Figure 5.2.11.3 Filter drawer on Food List Page

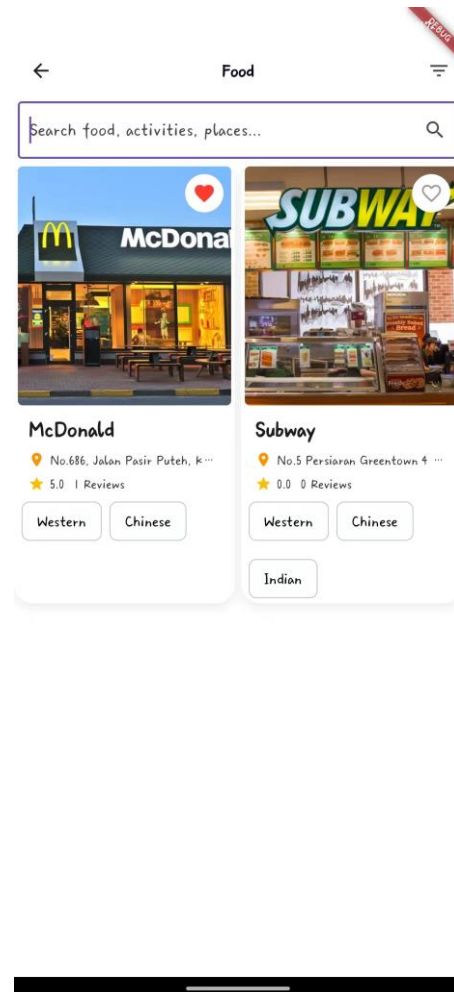


Figure 5.2.11.4 Example of Filtered Food List on Food List Page

The Food List Page displays all available food businesses. Each item shows a preview that includes the first image from the business's photo gallery, up to four category tags, the address, average rating, and total rating count. Users can search for food using the search bar and apply multiple filters to refine their results. Filtering options include:

- Rating Star: Filter businesses based on average star rating.
- Category Tags: Select multiple category tags
- Sorting: Sort results by either average rating or review count.

These filters can be used individually or in combination to customize the food listing.

Figure 5.2.11.4 illustrates a filtered food list where the results are limited to businesses tagged with Western and Chinese, and sorted by review count in descending order.

## 5.2.12 Place List Pge

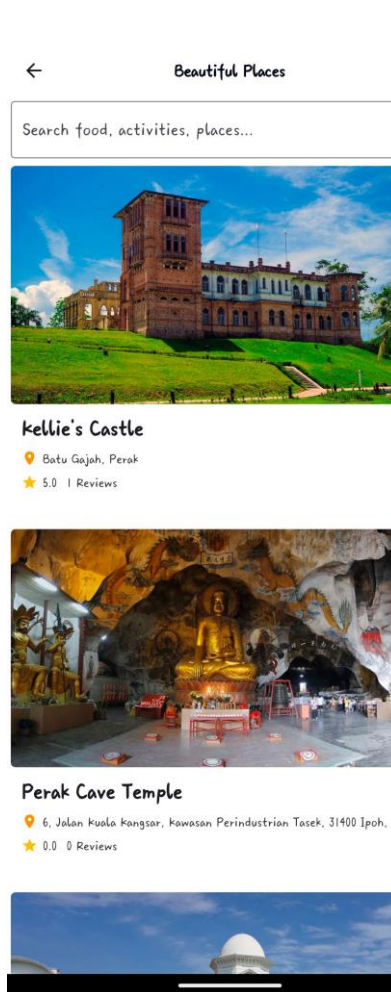


Figure 5.2.12.1  
Place List Page

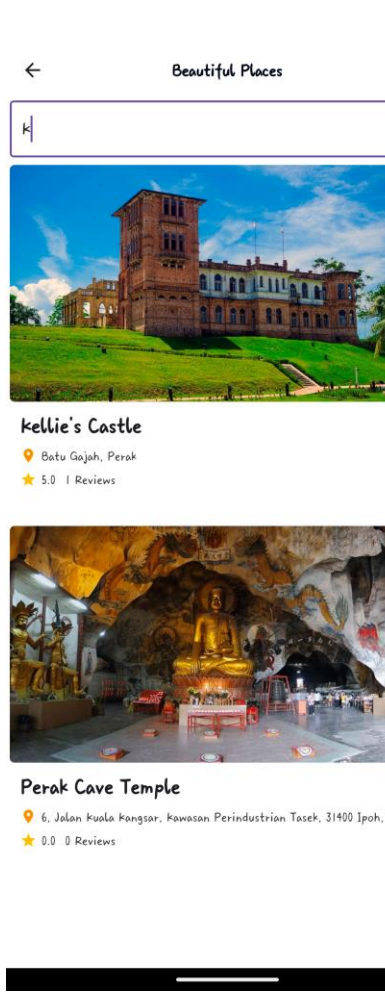


Figure 5.2.12.2 Search  
Feature on Place List Page

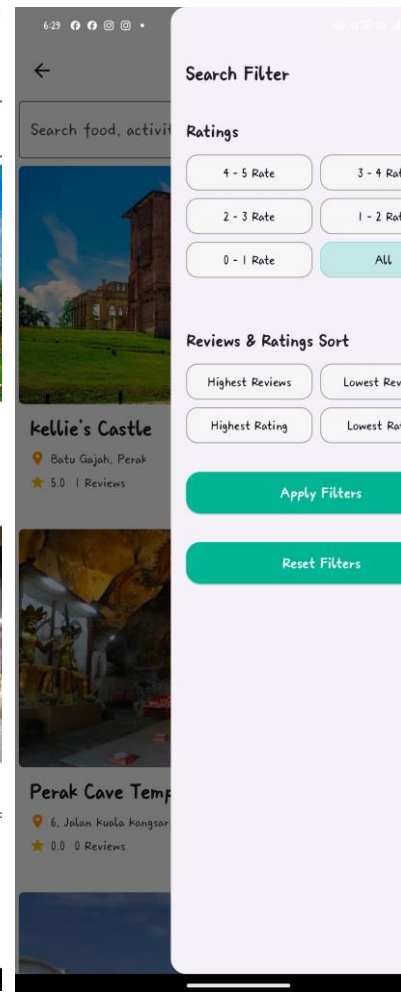


Figure 5.2.12.3 Filter  
drawer on Place List Page

The Place List Page displays all available attractions. Each item shows a preview that includes the image from the business's photo gallery, up to four category tags, the address, average rating, and total rating count. Users can search for attraction using the search bar and apply multiple filters to refine their results. Filtering options include:

- Rating Star: Filter businesses based on average star rating.
- Sorting: Sort results by either average rating or review count.

These filters can be used individually or in combination to customize the place listing.

### 5.2.13 Activity List Pge

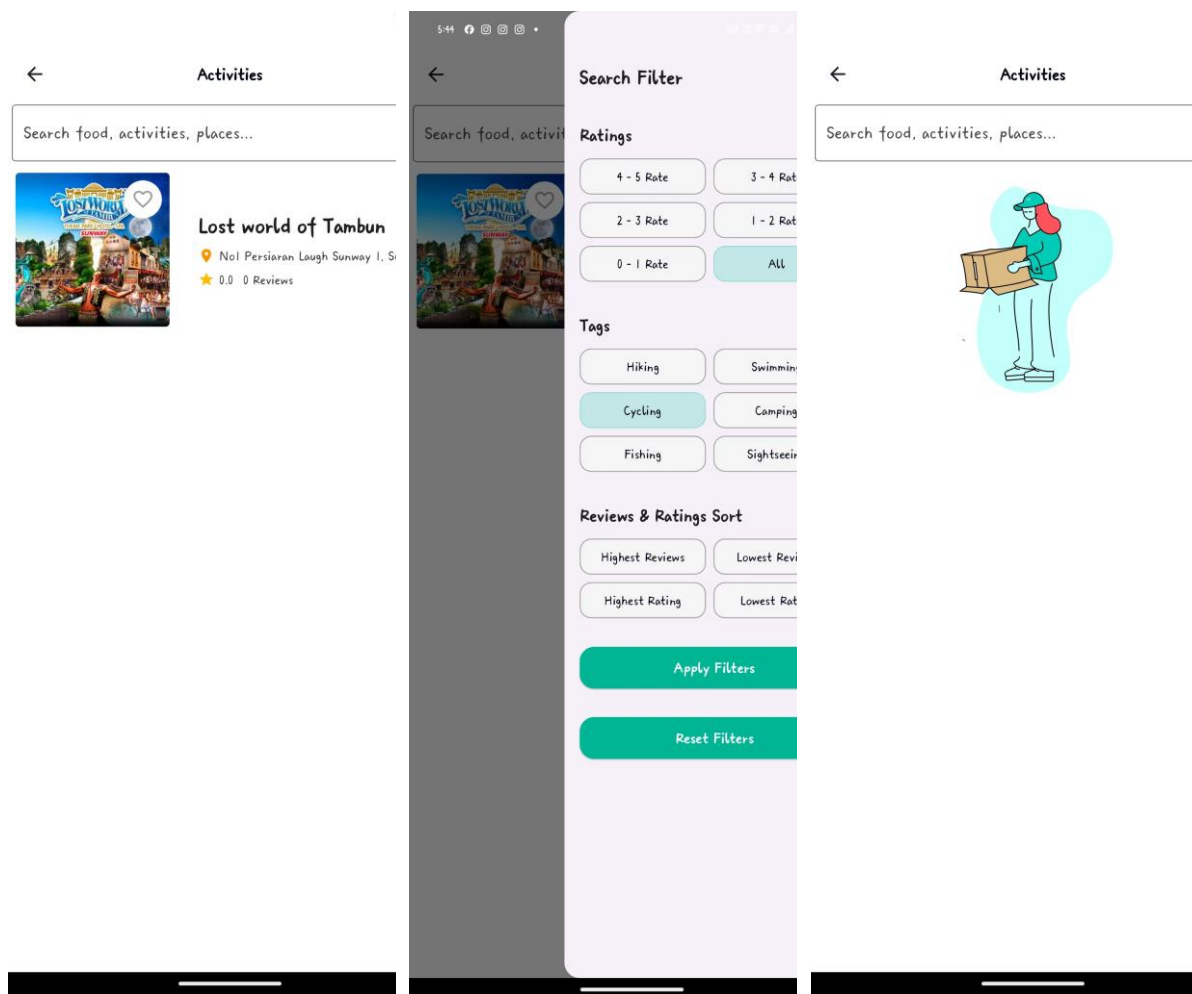


Figure 5.2.13.1 Activity List Page

Figure 5.2.13.2 Filter drawer on Activity List Page

Figure 5.2.13.3 Example of Filtered Activity List on Activity List Page

The UI logic of the Activity List Page is consistent with that of the Food List Page. Figure 5.2.13.3 shows empty filtered list when there is no activity with 'Cycling' tag.

## 5.2.14 Food Detail Page

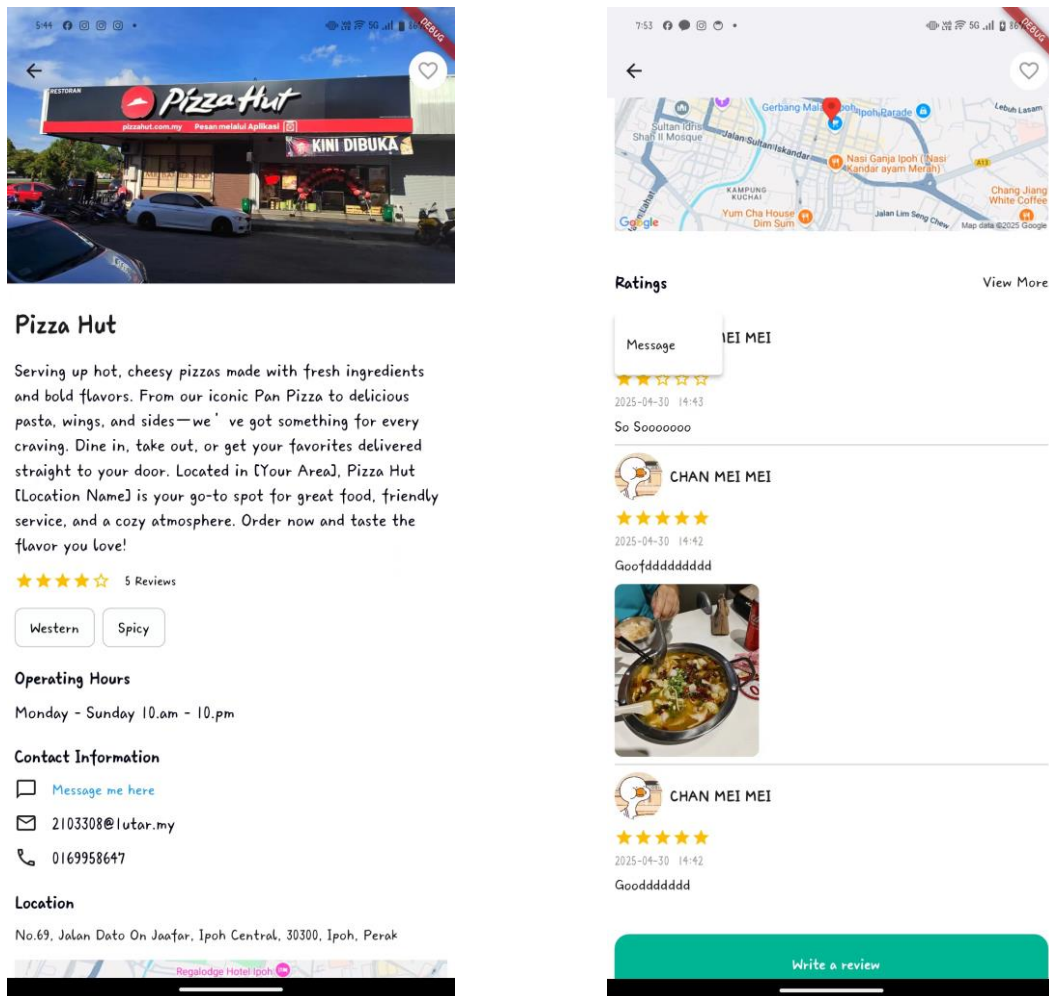


Figure 5.2.14 Food Detail Page

The Food Detail Page provides detailed information about the selected food business. A static map is displayed with a marker indicating the business location. Tapping on the static map navigates the user to the View Map Page for an interactive experience. Below the details, there is up to three user reviews are shown. If the review count less then three, the "View More" button is hidden. Each review displays the reviewer's avatar, name, comment, photos, rating, and timestamp. Tapping on a reviewer's avatar opens a pop-up menu with a "Message" option. Selecting this initiates a chat and redirects the user to the Chat Page to start a conversation. Tapping on 'Message me here' option under contact information section will redirect the user to the Chat Page to start a conversation with the merchant. At the bottom of the review section, a "Write Review" button is available. Tapping this navigates the user to the Write Review Page, where they can submit their own review.



### 5.2.15 Activity Detail Page

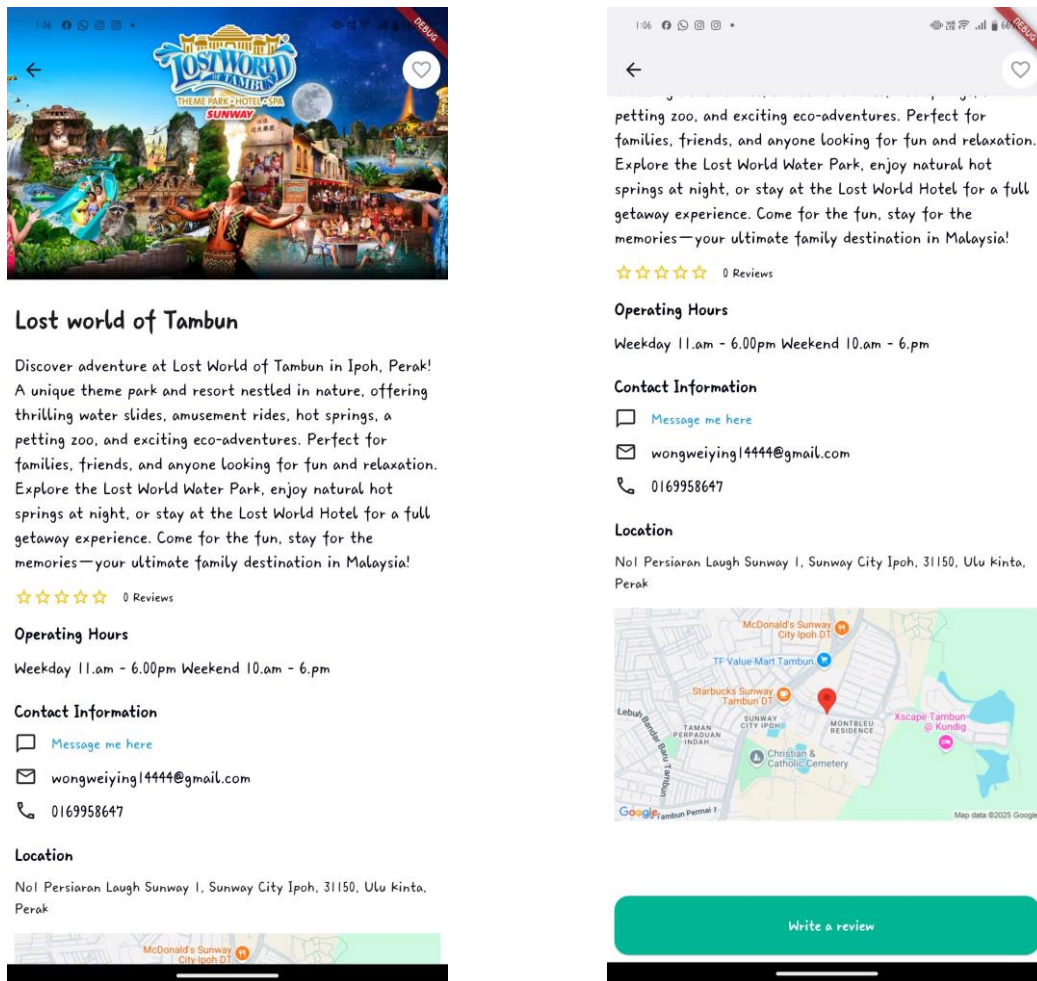


Figure 5.2.15 Activity Detail Page

The UI logic of the Activity Detail Page is consistent with that of the Food Detail Page.

### 5.2.16 Place Detail Page

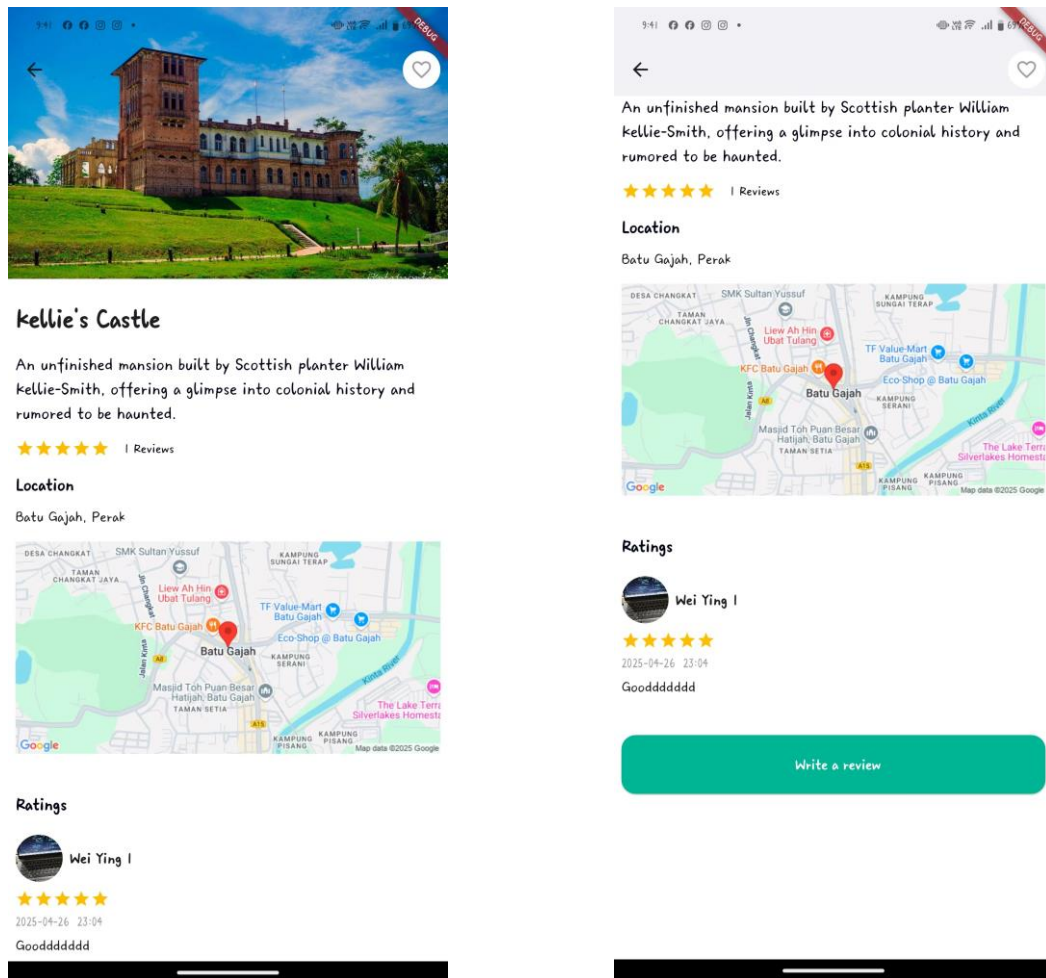


Figure 5.2.16 Place Detail Page

The UI logic of the Place Detail Page is consistent with that of the Food Detail Page, except that there is no option for users to initiate a conversation with the merchant.

### 5.2.17 View Map

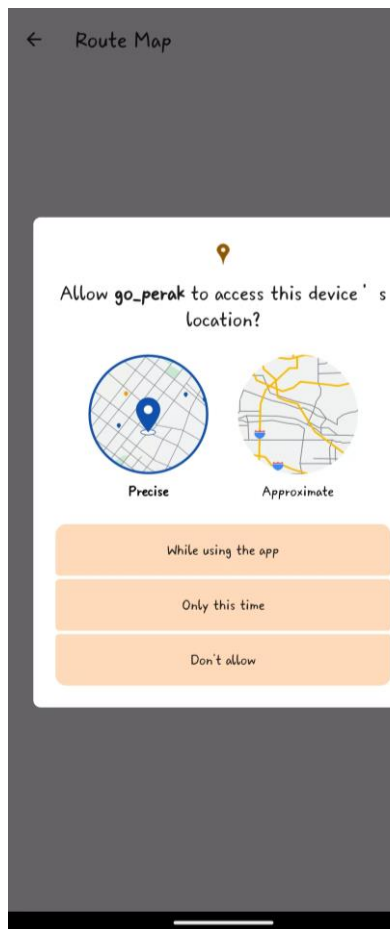


Figure 5.2.17.1 Ask user's permission on getting device's location on the View Map Page



Figure 5.2.17.2 User's permission is not granted on View Map Page

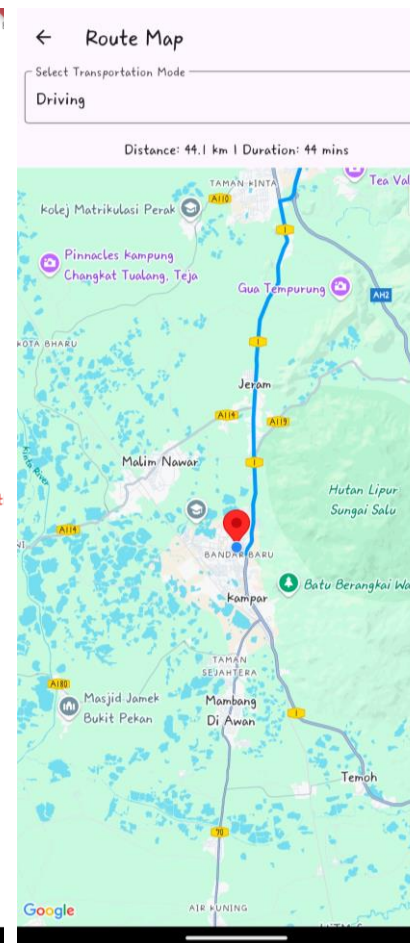


Figure 5.2.17.3 Travel Information by Car on the View Map Page

The system checks the user's permission to access the device's location. If permission is initially denied, the system will prompt the user to grant it. Once granted, the system displays the route, estimated time of arrival (ETA), and total distance. Users can also select different modes of transportation. If the user rejects the access, error message will be displayed.



### 5.2.18 Write Review Page

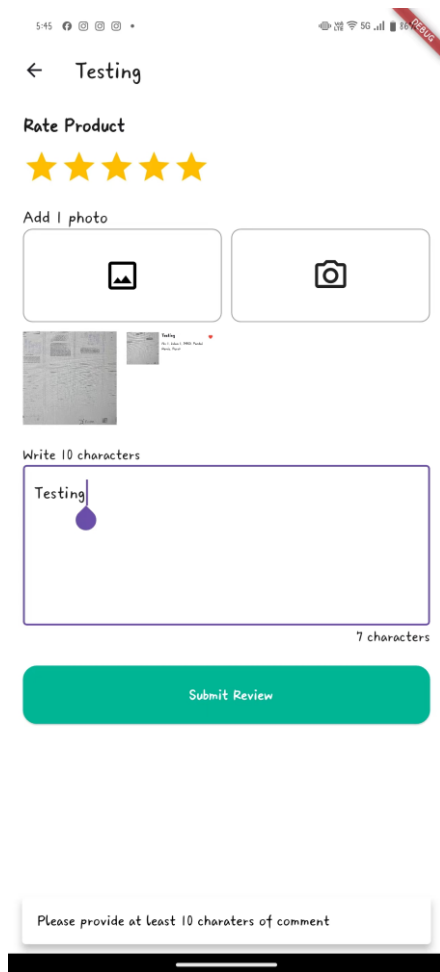


Figure 5.2.18.1 Write Review Page

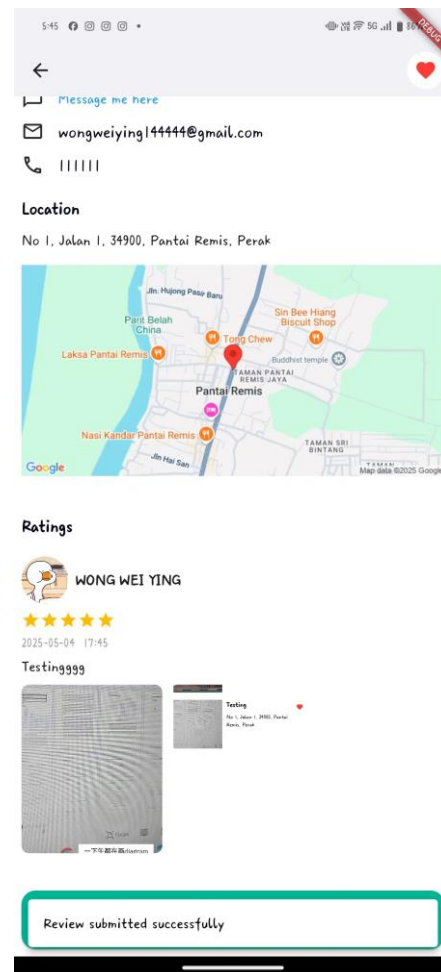


Figure 5.2.18.2 Example of a Refreshed Review Record After a Review is Updated on the Previous Page

The user must select a rating before submitting the review and enter a comment of at least 10 characters. Else, the system will display an error message. The system allows the user to optionally upload photos via the gallery or camera. After the user submits or updates a review, the system will automatically refresh the previous page to display the latest review record.

### 5.2.19 Review List Page

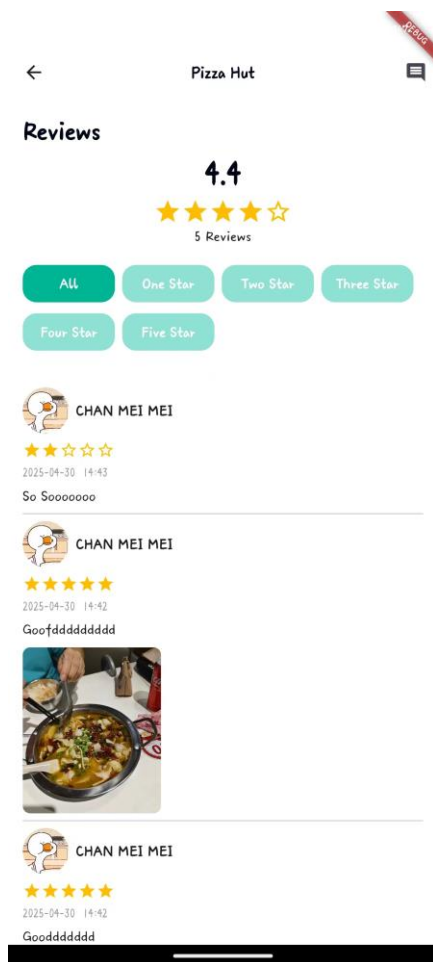


Figure 5.2.19.1 Review List Page

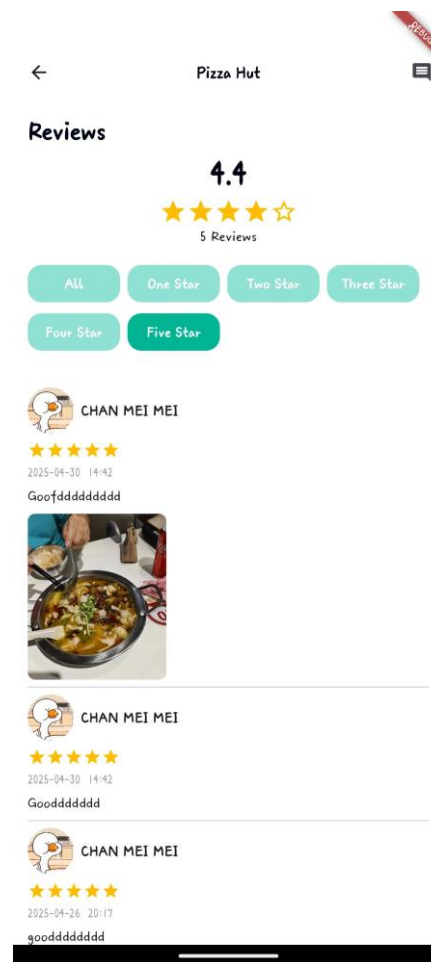


Figure 5.2.19.2 Filter feature on Review List Page

The system will display reviews sorted by timestamp, with the latest reviews shown at the top. Users can filter the reviews by star rating to view only reviews that match the selected rating value.

### 5.2.20 Chat Room Page

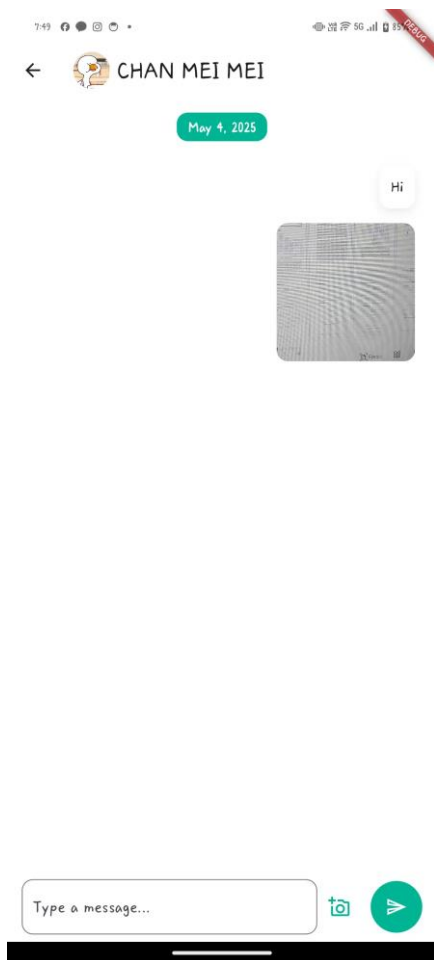


Figure 5.2.20.1 Chat Room Page (Sender)

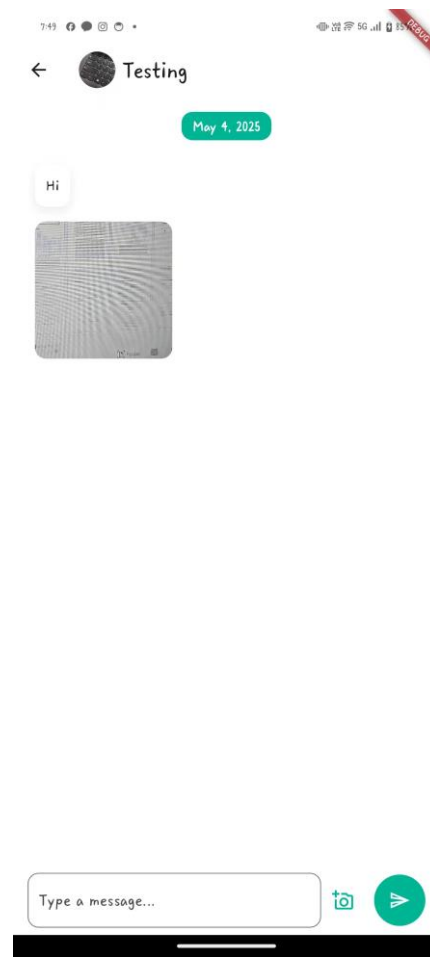


Figure 5.2.20.2 Chat Room Page (Receiver)

The user can send both text messages and images in the chat room. The system will display messages in real-time as they are sent and received.

### 5.2.21 Chat List Page



Figure 5.2.21.1 Chat List Page  
(Sender from Figure 5.2.19.1)



Figure 5.2.21.2 Chat List Page  
(Receiver from Figure 5.2.19.2)

The system will display a list of chats showing the sender's profile picture, username, last message, and the number of unread messages. When the user enters a chat room, the unread message count for that conversation will automatically reset to zero

### 5.2.22 Profile Page

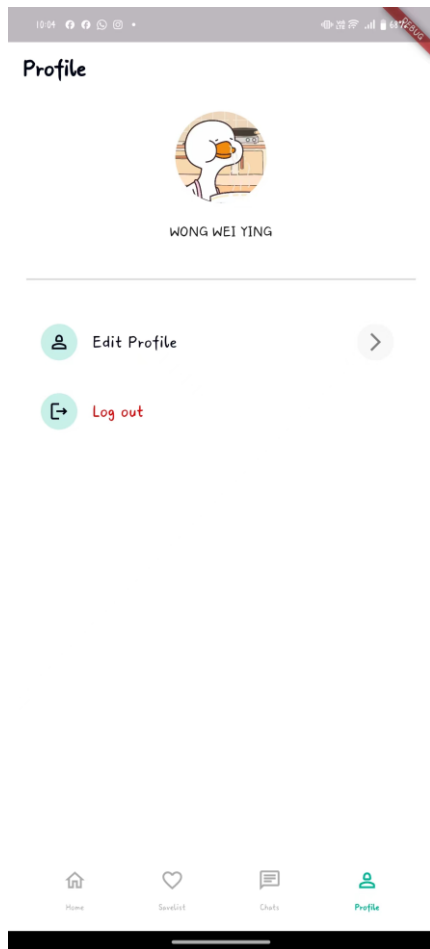


Figure 5.2.22.1 Profile Page

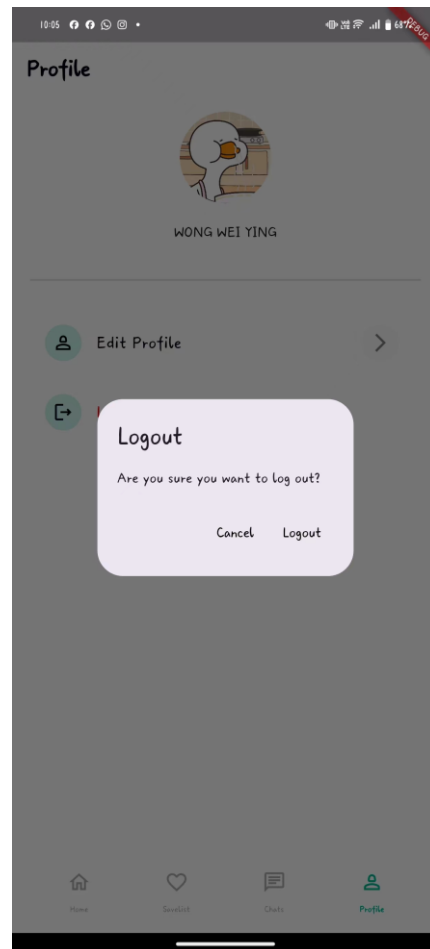


Figure 5.2.22.2 Log out on Profile Page

The profile page provides two options: 'Edit Profile' and 'Log Out'. Selecting 'Edit Profile' will redirect the user to the edit profile page. Selecting 'Log Out' will trigger a confirmation dialog; if the user confirms, the system will log them out and redirect them to the login page.

### 5.2.23 Edit Profile Page

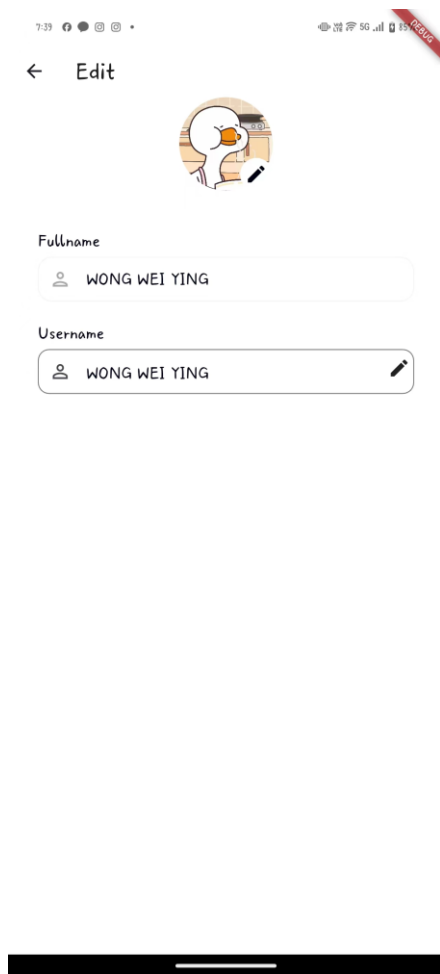


Figure 5.2.23.1 Edit Profile Page

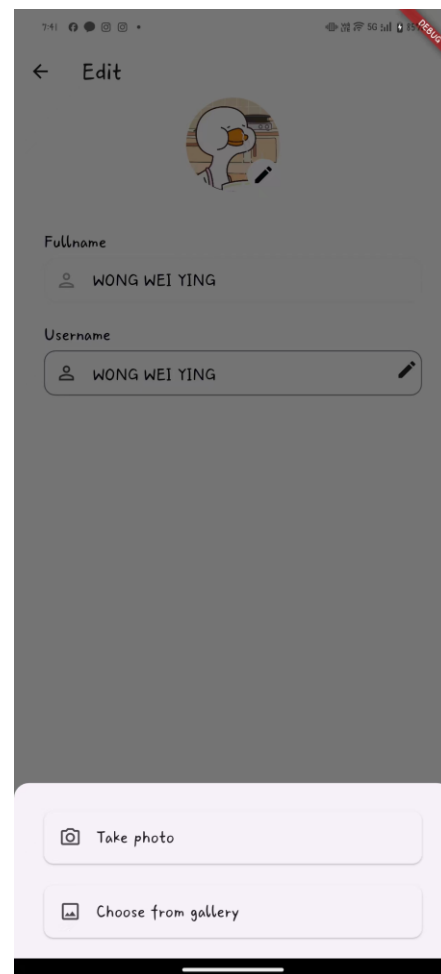


Figure 5.2.23.2 Update Profile Picture on Edit Profile Page

On the edit profile page, the user can update their profile picture by tapping on the avatar. A bottom sheet is triggered for the user to upload a new photo from the gallery or capture one via the camera. After the upload is successful, a success message is displayed and the latest profile image will be reflected across the entire system. When the user clicks on the username text field, they will be directed to the edit profile form page.

### 5.2.24 Edit Profile Form Page

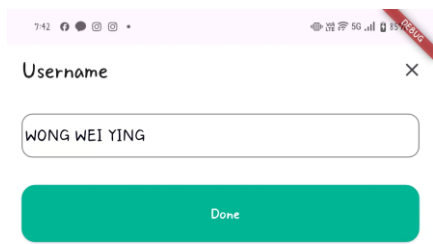


Figure 5.2.24.1 Edit Profile Form Page

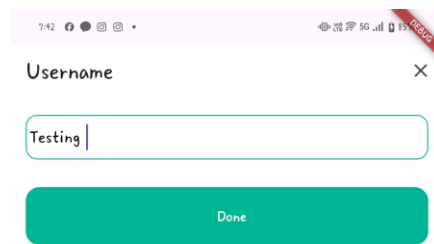


Figure 5.2.24.2 Example of updating username on Edit Profile Form Page

On Edit Profile Form Page, old username is passed into the textfield. Before update request is sent to the Firestore, the system will check if the new username is similar to the old username. Then, the system will check if new username is taken by other user. After the upload is successful, a success message is displayed and the user will be redirected to the previous page. The latest username will be reflected across the entire system.

### 5.2.25 Merchant Home Page

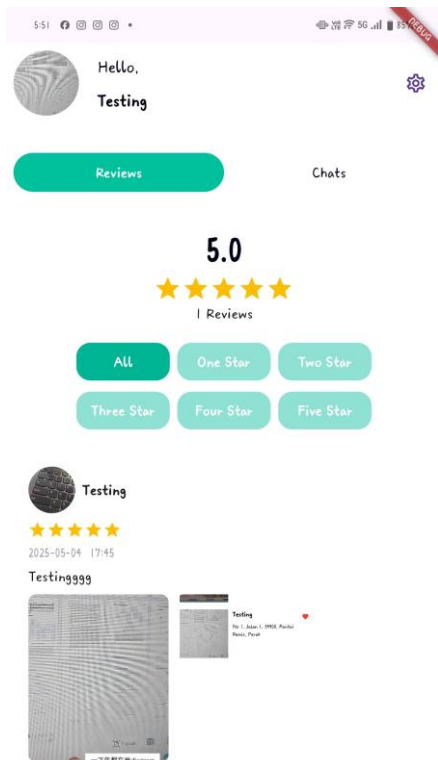


Figure 5.2.25.1 Merchant Home Page  
(Review Tab)

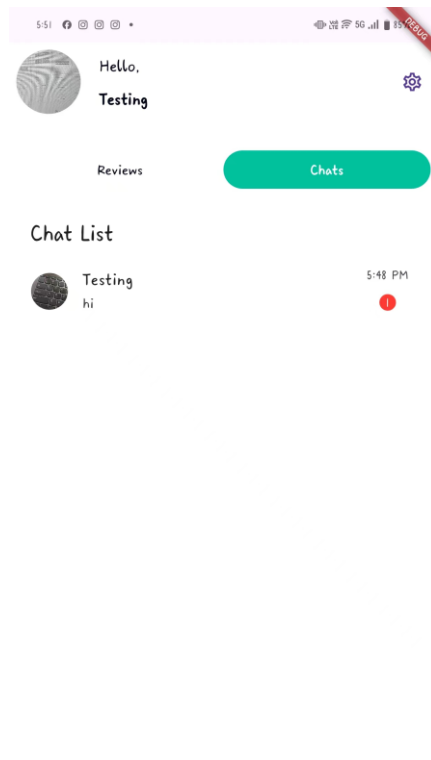


Figure 5.2.25.2 Merchant Home Page  
(Chat List Tab)

On the Merchant Home Page, the system will display customer reviews. The merchant can filter the reviews based on the rating value. The merchant can swipe to right to see the chat list. Tapping on the settings icon will direct the merchant to the Setting Page.



### 5.2.26 Merchant Setting Page

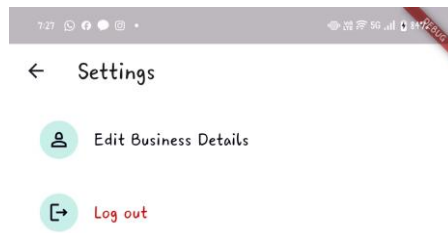


Figure 5.2.26 Merchant Setting Page

The Setting Page provides two options: 'Edit Business Detail and 'Log Out'. Selecting 'Edit Profile' will redirect the user to the edit business page. Selecting 'Log Out' will trigger a confirmation dialog; if the merchant confirms, the system will log them out and redirect them to the login page.

### 5.2.27 Merchant Edit Business Detail Page

10:15

Business Details

SAVE

Description

Testing

Tags

Western Malay

Operating Hours

Testing

Contact Information

wongweiyong14444@gmail.com

111111

Location

No 1, Jalan 1, 34900, Pantai Remis, Perak

Photos

Figure 5.2.27 Edit Business Detail Page

On Edit Business Detail Page, the merchant can update photos via gallery. Tapping on the edit icon on photos section will open the user's gallery. By tapping on other edit icon, it will direct user to Edit Business Detail Form Page. The user need to edit the detail one-by-one.

5.2.28 Edit Business Detail Form Page

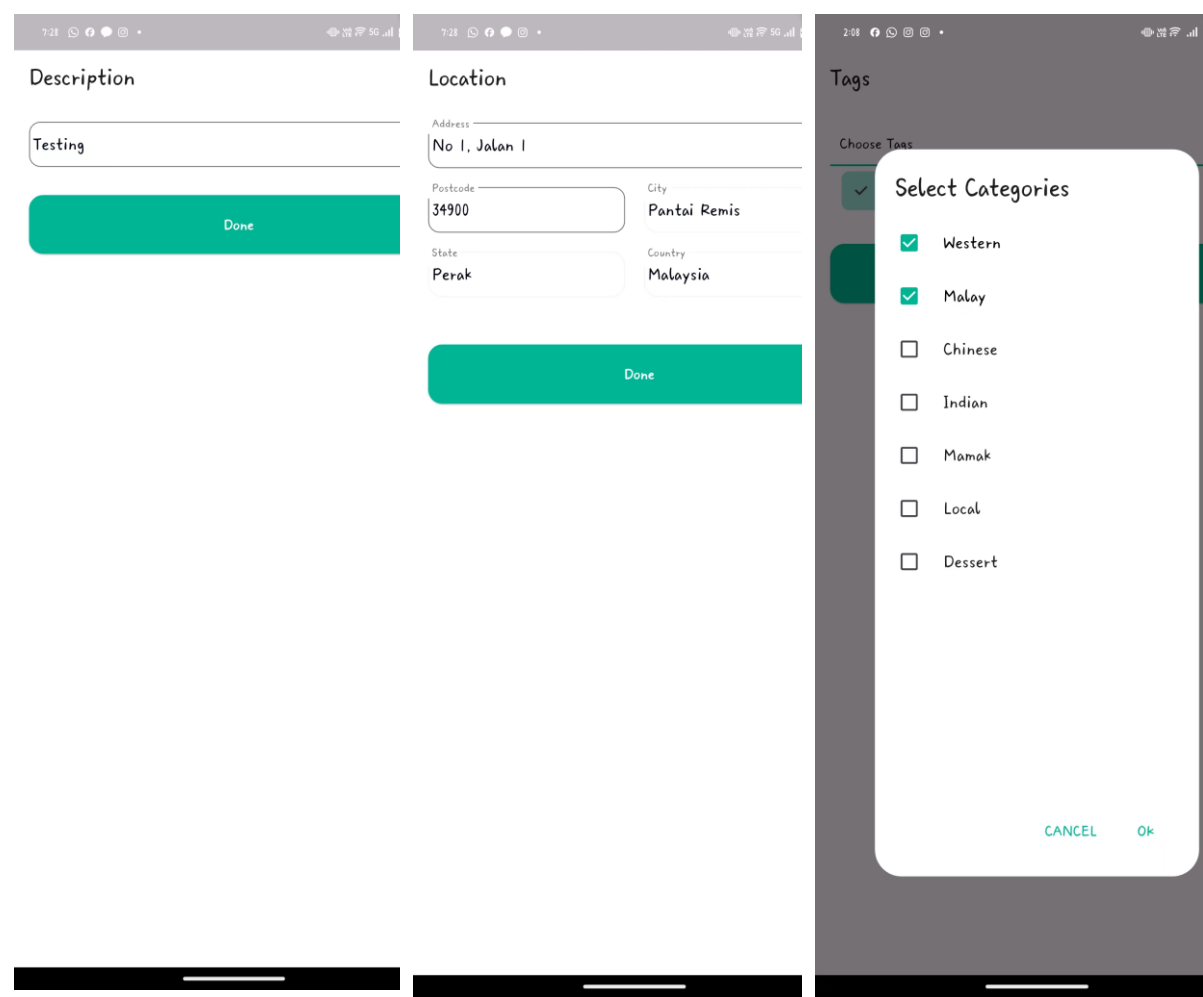


Figure 5.2.28.1 Update text data on Edit Business Detail Form Page

Figure 5.2.28.2 Update address on Edit Business Detail Form Page

Figure 5.2.28.3 Update tags on Edit Business Detail Form Page

On Edit Business Detail Form Page, old data is passed in. Once the data is updated successfully, a success message will be displayed and it will pop merchant back to the Edit Business Detail Page. The latest information will be reflected across the system, including user view.

## Chapter 6 System Testing

### 6.1 Test Cases

#### 6.1.1 Sign Up Functionality

Module Name		Sign Up Functionality			
Created By		Wong Wei Ying			
Created Date		25 April 2025			
Executed Date		25 April 2025			
Precondition Steps		1. Select Sign Up button on Login Page 2. Navigate to Select Role Page 3. Select Role 4. Navigate to Scan IC Page			
Test Case	Test Description	Test Steps	Expected Result	Actual Result	Status
<b>Scan IC Page</b>					
F0101	Upload correct IC image	1. Select Scan IC button 2. Select gallery/ camera 3. Upload valid IC image	IC number and fullname extracted	As expected	Pass
F0102	Upload invalid IC image	1. Select Scan IC button 2. Select gallery/ camera 3. Upload invalid IC image	Show error message	As expected	Pass
<b>Face Comparison Page</b>					
F0103	Face comparison is correct	1. Click 'Proceed to Face Comparison' button on Scan IC Page 2. Navigate to Face Comparison Page 3. Scan face via live camera	Show success message and direct user to Sign Up Page	As expected	Pass

## CHAPTER 6 SYSTEM TESTING

F0104	Face comparison fail	1. Click 'Proceed to Face Comparison' button on Scan IC Page 2. Navigate to Face Comparison Page 3. Scan face via live camera	Show error dialog	As expected	Pass
<b>Sign Up Page</b>					
F0105	Empty inputs	1. Click Sign Up/Continue button	Show error message	As expected	Pass
F0106	Email address is in wrong format	1. Enter invalid email address 2. Enter password 3. Click Sign Up/Continue button	Show error message	As expected	Pass
F0107	Password length < 6	1. Enter valid email address 2. Enter invalid password 3. Click Sign Up/Continue button	Show error message	As expected	Pass
F0108	Email address is registered on Firebase Auth	1. Enter registered email address 2. Enter valid password 3. Click Sign Up/Continue button	Show error message	As expected	Pass
F0109	Successful sign up and selected role is 'User'	1. Enter valid email address 2. Enter valid password 3. Click Sign Up button	Direct user to Login Page	As expected	Pass
<b>Merchant Detail Page</b>					

F0110	Continue to fill up merchant detail when selected role is 'Merchant'	1. Enter valid email address 2. Enter valid password 3. Click Continue button	Direct merchant to Merchant Detail Page	As expected	Pass
F0111	Empty inputs	1. Click Sign Up button	Show error message	As expected	Pass
F0112	Successful merchant sign up	1. Fill up all merchant detail 2. Click Sign Up button	Show success message and direct merchant to Login Page	As expected	Pass

Table 6.1.1 Sign Up functionality Test Cases

### 6.1.2 Login Functionality

Module Name		Login Functionality			
Created By		Wong Wei Ying			
Created Date		25 April 2025			
Executed Date		25 April 2025			
Test Case	Test Description	Test Step	Expected Result	Actual Result	Status
F0201	Incorrect email format	1. Enter invalid email address 2. Enter invalid password 3. Click Login button	Show error message	As expected	Pass
F0202	Email address not found on Firebase Auth	1. Enter unregistered email address 2. Enter invalid password 3. Click Login button	Show error message	As expected	Pass

F0203	Incorrect password	1. Enter valid email address 2. Enter invalid password 3. Click Login button	Show error message	As expected	Pass
F0204	Empty inputs	1. Click Login button	Show error message	As expected	Pass
F0205	Successful log in of user	1. Enter valid email address 2. Enter valid password 3. Click Login button	Direct user to User Home Page	As expected	Pass
F0206	Successful log in of merchant	1. Enter valid email address 2. Enter valid password 3. Click Login button	Direct user to Merchant Home Page	As expected	Pass

Table 6.1.2 Login functionality Test Cases

### 6.1.3 Log out Functionality

Module Name		Logout Functionality			
Created By		Wong Wei Ying			
Created Date		25 April 2025			
Executed Date		25 April 2025			
Precondition Steps		1. Click log out on Profile page by user Or 1. Click log out on Account Page by merchant			
Test Case	Test Description	Test Step	Expected Result	Actual Result	Status
F0301	Confirm log out	1. Click 'Ok' on confirmation dialog	Direct user to Login Page	As expected	Pass
F0302	Cancel logout	1. Click 'Cancel' on confirmation dialog	Remain in same page	As expected	Pass

Table 6.1.3 Log out functionality Test Cases



**6.1.4 Reset Password Functionality**

Module Name		Reset Password Functionality			
Created By		Wong Wei Ying			
Created Date		25 April 2025			
Executed Date		25 April 2025			
Test Case	Test Description	Test Step	Expected Result	Actual Result	Status
F0401	Wrong email address format	1. Click 'Forgot Password' 2. Navigate to 'Reset Password' Page 3. Enter invalid email address 4. Close 'Send Link' button	Show error message	As expected	Pass
F0402	Correct email address format	1. Click 'Forgot Password' 2. Navigate to 'Reset Password' Page 3. Enter valid email address 4. Close 'Send Link' button	Show success message and receive a link on email	As expected	Pass
F0403	Use updated password to login	1. Navigate to Login Page 2. Enter valid email address and password 3. Click Login button	Show success message and direct to homepage	As expected	Pass

Table 6.1.4 Reset Password functionality Test Cases



**6.1.5 Direct Message Functionality**

Module Name		Direct Message Functionality			
Created By		Wong Wei Ying			
Created Date		25 April 2025			
Executed Date		25 April 2025			
Test Case	Test Description	Test Step	Expected Result	Actual Result	Status
<b>Chat Room Page</b>					
F0501	Send one text message	1. Send one message	On receiver chat list, total of unread message increase by one and show 'Testing'	As expected	Pass
F0502	Send multiple text messages	1. Send three messages	On receiver chat list, total of unread message increase by three and show 'Testing 3'	As expected	Pass
F0503	Send one photo	1. Send one photo	On receiver chat list, total of unread message increase by one and show '  Photo'	As expected	Pass
F0504	Send multiple photos at once	2. Send three photos	On receiver chat list, total of unread message increase by one and show '  3 photos'	As expected	Pass
F0505	Receiver reads message	1. Read messages	On receiver chat list, total of unread message turns zero	As expected	Pass
F0506	Ensure real-time communication	1. Sender sends message 2. Receiver receives message	On receiver chat list, total of unread message turns zero and show 'Hi'	As expected	Pass

<b>Review's List on Food/Activity/Place Detail Page and View More Review Page</b>					
F0507	Start communication with user	1. Click reviewer's avatar 2. Click 'Message' option on pop up drop down menu	Navigate to Chat Room Page	As expected	Pass
<b>Food/Activity/Place Detail Page</b>					
F0508	Start communication with merchant	1. Click 'Message Me Here' text button	Navigate to Chat Room Page	As expected	Pass

Table 6.1.5 Direct Message functionality Test Cases

### 6.1.6 View and Filter Reviews Functionality

Module Name		View and Filter Review Functionality			
Created By		Wong Wei Ying			
Created Date		25 April 2025			
Executed Date		25 April 2025			
<b>Test Case</b>	<b>Test Description</b>	<b>Test Step</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status</b>
<b>Food/Place/Activity Detail Page</b>					
F0601	Able to view item's rating	1. Clicks View More on detail page 2. Navigate to Review List Page	Show review list with average rating, rating count, and reviews sorted by timestamp (descending)	As expected	Pass
F0602	Filter reviews by rating	1. Clicks View More on detail page 2. Navigate to Review List Page	Show all reviews with 4-star rating are displayed	As expected	Pass

		3. Select filter option			
F0603	Filter returns no results	1. Clicks View More on detail page 2. Navigate to Review List Page 3. Select filter option that has no matching reviews	No reviews are displayed	As expected	Pass
F0604	View review photos in fullscreen	1. Click on review's photo	Photo opens in fullscreen. User can swipe left/right to view other photos posted by the reviewer	As expected	Pass
F0605	View More button is shown when reviews > 3	1. Check if 'View More' button appear	View More button is shown when reviews > 3	As expected	Pass

Table 6.1.6 View and Filter Reviews functionality Test Cases

### 6.1.7 View Map Functionality

Module Name		View Map Functionality			
Created By		Wong Wei Ying			
Created Date		25 April 2025			
Executed Date		25 April 2025			
Precondition Steps		1. Click map on Food/Place/Activity Detail Page 2. Navigate to View Map Page			
<b>Test Case</b>	<b>Test Description</b>	<b>Test Step</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status</b>

F0701	Allow access on user's device location	1. Allow access on user's device location	Show route, total distance in km and total time travel by car.	As expected	Pass
F0702	Select different transportation mode	1. Select transportation mode on drop down menu	Show new route, total distance in km and total time travel by walking	As expected	Pass
F0703	Deny access on user's device location	1. Deny access on user's device location	Map is not displayed	As expected	Pass

Table 6.1.7 View Map functionality Test Cases

### 6.1.8 Write Review Functionality

Module Name		Write Review Functionality			
Created By		Wong Wei Ying			
Created Date		25 April 2025			
Executed Date		25 April 2025			
Test Case	Test Description	Test Step	Expected Result	Actual Result	Status
<b>Write Review Page</b>					
F0801	Empty inputs	1. Click Submit button	Show error message	As expected	Pass
F0802	Only photo input is empty	1. Select rating star 2. Enter comment 3. Click Submit button	Show success message and direct user to previous page with new review on top	As expected	Pass
F0803	Successful submission of a rating	1. Select rating star 2. Enter comment 3. Upload photos 3. Click Submit button	Show success message and direct user to previous page	As expected	Pass

			with new review on top		
F0804	Comment character < 10	1. Select rating star 2. Enter comment 3. Click Submit button	Show error message	As expected	Pass
F0805	Upload photo via camera	1. Click on camera icon 2. Take photo via camera	Show selected photos	As expected	Pass
F0806	Upload photo via gallery	1. Click on gallery icon 2. Pick photos	Show selected photos in grid view with three images per row	As expected	Pass

Table 6.1.8 Write Review functionality Test Cases

### 6.1.9 Edit Profile Functionality

Module Name		Edit Profile Functionality			
Created By		Wong Wei Ying			
Created Date		25 April 2025			
Executed Date		25 April 2025			
Precondition Steps		1. Click 'Edit Profile' option on profile page 2. Navigate to Edit Profile Page			
Test Case	Test Description	Test Step	Expected Result	Actual Result	Status
F0901	Change profile picture with camera	1. Click profile picture 2. Click camera option on bottom sheet 3. Take photo 4. Click Ok button on photo preview	New profile picture is updated on every part of the system	As expected	Pass
F0902	Change profile	1. Click profile picture	New profile picture is	As expected	Pass

	picture with gallery	2. Click gallery option on bottom sheet 3. Choose a photo	updated on every part of the system		
F0903	Same username	1. Click username textfield 2. Navigate to Edit Profile form and pass old username to textfield 5. Click Submit button	Show error message	As expected	Pass
F0904	Empty username	1. Click username textfield 2. Navigate to Edit Profile form and pass old username to textfield 3. Clear old username 3. Click Submit button	Show error message	As expected	Pass
F0905	Register a username that has already been taken by another user	1. Click username textfield 2. Navigate to Edit Profile form and pass old username to textfield 3. Input registered username 4. Click Submit button	Show error message	As expected	Pass
F0906	Valid username	1. Click username textfield 2. Navigate to Edit Profile form and pass old username to textfield 3. Input valid username 4. Click Submit button	Show success message. Pop back to Edit Profile Page. Username is updated on every part of the system	As expected	Pass

Table 6.1.9 Edit Profile functionality Test Cases

**6.1.10 Real-time Search Merchant Functionality**

Module Name		Search Merchant Functionality			
Created By		Wong Wei Ying			
Created Date		25 April 2025			
Executed Date		25 April 2025			
Precondition Steps		1. Click search icon on home page 2. Navigate to Search Explore Page			
Test Case	Test Description	Test Step	Expected Result	Actual Result	Status
F1001	Search merchant	1. Start typing (e.g. a)	Show all food, activity, place with character 'a'	As expected	Pass
F1002	Search merchant	1. Start typing (e.g. Pizza)	Show all food stores with Pizza	As expected	Pass
F1003	Cancel search	1. Start typing 2. Click cancel icon on search bar	Show the first four items from place, food and activity	As expected	Pass
F1004	Empty search input	-	Show the first four items from place, food and activity	As expected	Pass

Table 6.1.10 Real-time Search functionality Test Cases

**6.1.11 Filter Merchant Functionality**

Module Name		Filter Merchant Functionality			
Created By		Wong Wei Ying			
Created Date		25 April 2025			
Executed Date		25 April 2025			

Precondition steps		1. Open filter drawer on Food/ Place/ Activity List Page				
Test Case	Test Description	Test Step	Test Data	Expected Result	Actual Result	Status
<b>Food/ Place/ Activity List Page</b>						
F1101	Filter by rating star	1. Select rating star to be filtered 2. Click 'Apply Filter' button	1-2 rating star	Show all items with average rating between 1-2	As expected	Pass
F1102	Sort merchant	1. Select sorting option 2. Click 'Apply Filter' button	Highest Rating	Show all items sorted by rating in descending order	As expected	Pass
F1103	Filter by rating star and sort the list at once	1. Select rating star 2. Select sorting option 3. Click 'Apply Filter' button	4-5 rating star, Highest rating count	Show all items with average rating star between 4 and 5 and sorted by rating count in descending order	As expected	Pass
F1104	Reset filter	1. Click 'Reset Filter' button	N/A	Show all items	As expected	Pass
<b>Food List Page</b>						
F1105	Filter by one	1. Select one tag 2. Click 'Apply Filter' button	Western	Show all items with Western tag	As expected	Pass



## CHAPTER 6 SYSTEM TESTING

	category tag					
F1106	Filter by multiple category tag	1. Select two tags 2. Click 'Apply Filter' button	Western, Chinese	Show all items with Western and Chinese tags	As expected	Pass
F1107	Filter by rating star and category tags at once	1. Select rating star 2. Select two tags 3. Click 'Apply Filter' button	4-5 rating star, Western, Chinese	Show all items with average rating star between 4 and 5 and with Western and Chinese tags	As expected	Pass
F1108	Filter by tags and sort the list at once	1. Select tags 2. Select sorting option 3. Click 'Apply Filter' button	Western, Highest rating count	Show all items with Western tag and sorted by the rating count in descending order	As expected	Pass
F1109	Filter by rating star, tags and sort	1. Select rating star 2. Select tags 3. Select sorting option	4-5 rating star, Western,	Show all items with average rating star between 4	As expected	Pass

	the list at once	4. Click 'Apply Filter' button	Highest rating count	and 5, with Western tag and sorted by the rating count in descending order		
<b>Activity List Page</b>						
F1110	Filter by one category tag	1. Select one tag 2. Click 'Apply Filter' button	Sight seeing	Show all activities with Sightseeing tag	As expected	Pass
F1111	Filter by multiple category tags	1. Select two tags 2. Click 'Apply Filter' button	Sight seeing Camping	Show all items with Sightseeing and Camping tags.	As expected	Pass
F1112	Filter by rating star and category tags at once	1. Select rating star 2. Select two tags 3. Click 'Apply Filter' button	4-5 rating star, Sight seeing Camping	Show all items with average rating star between 4 and 5 and Sightseeing and Camping tags	As expected	Pass
F1113	Filter by tags and sort the	1. Select tags 2. Select sorting option	Sight Seeing,	Show all items with Sightseeing	As expected	Pass

	list at once	3. Click 'Apply Filter' button	Highest rating count	tag and sorted by the rating count in descending order		
F1114	Filter by rating star, tags and sort the list at once	1. Select rating star 2. Select tags 3. Select sorting option 4. Click 'Apply Filter' button	4-5 rating star, Sighseei ng, Highest rating count	Show all activities with average rating star between 4 and 5, with Sightseeing tag and sorted by the rating count in descending order	As expected	Pass

Table 6.1.11 Filter Merchant functionality Test Cases

**6.1.12 Add/Remove Merchant from Savelist Functionality**

Module Name		Add/Remove Merchant to Savelist Functionality				
Created By		Wong Wei Ying				
Created Date		25 April 2025				
Executed Date		25 April 2025				
<b>Test Case</b>	<b>Test Description</b>	<b>Test Step</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status</b>	

F1201	Add merchant to savelist	1. Click grey color save icon 2. Navigate to Savelist page	Saved item appears on Savelist Page	As expected	Pass
F1202	Remove merchant from Savelist	1. Click red color save icon 2. Navigate to Savelist page	The removed saved item does not show on Savelist Page	As expected	Pass
F1203	Check save icon	1. Check all the save icon on every part of the system	Saved item shows red color save icon; Unsaved item shows grey color save icon	As expected	Pass

Table 6.1.12 Add/Remove Merchant from Savelist functionality Test Cases

### 6.1.13 Edit Business Detail Functionality

Module Name		Edit Business Detail Functionality			
Created By		Wong Wei Ying			
Created Date		25 April 2025			
Executed Date		25 April 2025			
Precondition Steps		1. Click 'Edit Business Detail option on Setting Page 2. Navigate to Edit Business Page			
Test Case	Test Description	Test Step	Expected Result	Actual Result	Status
F1301	Change business photos	1. Click edit icon on business photo 2. Select new photos from gallery	Show new updated photos on Edit Business Detail page	As expected	Pass
F1302	Change text-form detail	1. Click edit icon on text-form detail	Show error message	As expected	Pass

## CHAPTER 6 SYSTEM TESTING

	where the new detail is same as old detail	2. Navigate to Edit Business Detail Form and pass the old detail to text field 3. Click Submit button			
F1303	Change address with invalid postcode	1. Click edit icon on text-form detail 2. Navigate to Edit Business Detail Form and pass the old detail to text field 3. Enter invalid postcode 4. Click Submit button	Show error message	As expected	Pass
F1304	Change text-form detail successfully	1. Click edit icon on text-form detail 2. Navigate to Edit Business Detail Form and pass the old detail to text field 3. Enter valid detail 4. Click Submit button	Show success message and pop back to Edit Business Detail Page with updated detail	As expected	Pass
F1305	Empty input	1. Click edit icon on text-form detail 2. Navigate to Edit Business Detail Form and pass the old detail to text field 3. Clear old detail 4. Click Submit button	Show error message	As expected	Pass
F1306	Change business tags	1. Click edit icon on tags 2. Navigate to Edit Business Detail Form and show all selected tags 3. Click on the multi-select menu 4. Select new tags	Show success message and pop back to Edit Business Detail Page with updated tags	As expected	Pass

		5. Click Ok button to close the menu			
		6. Click Submit button			

Table 6.1.13 Edit Business Detail functionality Test Cases

### Chapter 7: Conclusion

#### 7.1 Project Review

In conclusion, this project successfully delivers a tourism-focused mobile application designed to enhance the travel experience in Perak, Malaysia, by providing personalized food and activity recommendations. By removing non-essential features, the system is optimized for performance and offers a streamlined user experience. The implementation of a categorization system ensures that recommendations are relevant and tailored to individual user preferences. Additionally, to address the growing concern of fake reviews, the system incorporates a credibility-driven approach by requiring users to scan their IC and complete facial verification. This verification process strengthens the reliability of user-generated content on the platform. The project is conducted according to the Software Development Life Cycle (SDLC), from requirement gathering to system design, implementation and testing, to ensure the robustness and usability of the application. Overall, the system meets its defined objectives and demonstrates its potential to serve as a trusted and efficient platform for both tourists and local businesses.

#### 7.2 Implementation Issues and Challenges

Throughout the development of the application, several key implementation issues and challenges were encountered that required practical decisions and adaptive strategies. One of the primary challenges was the integration of IC scanning and facial recognition. It requires accurate matching between the live photo and the photo on the identification card with third-party integration. This introduced issues related to lighting conditions, image clarity, and device compatibility, which sometimes led to false rejections or verification delays.

Initially, IC photo for verification is planned to store in the database, however, this approach was revised due to data privacy and security concerns. As a result, only the IC number was stored in the database, and the system performs live face comparison without saving sensitive images. This change ensured better compliance with data protection practices but required a reliable verification method that did not compromise accuracy.

## CHAPTER 7 CONCLUSION

Another major challenge arose when storage policy of Firebase Storage is updated. It introduces paid tiers for photo storage, which posed budget constraints. To address this, the image storage system was migrated to Supabase, an alternative platform offering more cost-effective and flexible image handling. This transition involved significant changes in the backend integration and image retrieval logic across the system.

The application also utilizes Google Maps API to display tourism locations and navigation features. However, the project faced issues due to Google Maps API quota limitations, which restricted the number of daily map loads and interactions. Consequently, the map features were scaled down and optimized to remain functional within the free quota, resulting in limited interactivity in some parts of the application.

### 7.3 Future Work

While the current system successfully delivers core functionalities such as tourism search, identity verification, and credible user reviews, several improvements are planned for future development to enhance usability, performance, and accessibility.

First, current application is only supported on Android and restricted to Malaysian users due to the IC-based verification system. In future iterations, the system may be expanded to support iOS platforms and international users, possibly by adopting passport verification or other global identification methods.

Second, the facial verification process can be strengthened by incorporating advanced liveness detection and exploring potential integration with official national digital ID systems, subject to legal and privacy compliance. This would improve the robustness of user authentication and further minimize the risk of fake reviews.

To overcome existing limitations, alternative solutions will also be explored. For instance, due to Google Maps API quota limits, future versions may adopt mapping solutions like Mapbox or OpenStreetMap for more extensive and flexible location features. Similarly, given Firebase's pricing changes, migrating media content to more cost-effective platforms like Supabase will continue to be evaluated.



## CHAPTER 7 CONCLUSION

Moreover, the review system may be enhanced using machine learning algorithms to detect fake content, provide sentiment analysis, and offer smarter review summaries. The recommendation engine could also be refined through AI-based personalization by learning user behavior and preferences over time.

To ensure better privacy and data control, features such as encrypted data storage and user-controlled privacy settings will be considered. The inclusion of multi-language support and offline functionality can further expand accessibility for users with diverse needs or limited internet access.


## REFERENCES

## REFERENCES

- [1] N. A. Bakar, N. A. Hashim, N. M. M. Nawi, M. A. Rahim, A. Muhamed Yusoff, C. Aziz, and G. Ahmad, "Travel mobile applications: the use of unified acceptance technology model," in *Int. J. Innov. Technol. Explor. Eng.*, vol.9, no.3, pp . 3118, 2020.
- [2] "TripAdvisor Insights," TripAdvisor, [Online]. Available: <https://www.tripadvisor.com/TripAdvisorInsights/w3680> (accessed: April. 17, 2024).
- [3] TripAdvisor, "Review Posting Guidelines," TripAdvisor, [Online]. Available: [https://www.tripadvisor.com/Trust-lvBd3L1aU38Y-Review\\_posting\\_guidelines.ht](https://www.tripadvisor.com/Trust-lvBd3L1aU38Y-Review_posting_guidelines.ht). (accessed: April. 17, 2024).
- [4] TripAdvisor, "TripAdvisor Insights," TripAdvisor. [Online]. Available: <https://www.tripadvisor.com/TripAdvisorInsights/w3688>. (accessed: April. 17, 2024).
- [5] TripAdvisor, "TripAdvisor Insights," TripAdvisor. [Online]. Available: <https://www.tripadvisor.com/TripAdvisorInsights/w3703>. (accessed: April. 17, 2024).
- [6] Traveloka, "Writing a Helpful Review," Traveloka. [Online]. Available: <https://www.traveloka.com/en-id/help/hotel/accommodation-product-information/hotel-information/review/writing-a-helpful-review>. (accessed: April. 17, 2024).
- [7] Visit a City, "Terms and Conditions," Visit a City. [Online]. Available: <https://www.visitacity.com/terms-and-conditions>. (accessed: April. 17, 2024).
- [8] Tripadvisor. (2014). Tripadvisor (Version 57.3) [Mobile application software]. Retrieved from Google Play Store: <https://play.google.com/store/apps/details?id=com.tripadvisor.tripadvisor>. (accessed: April. 17, 2024).
- [9] Traveloka. (Year). Traveloka (Version X.X.X) [Mobile application software]. Retrieved from Google Play Store: <https://play.google.com/store/apps/details?id=com.traveloka.android>. (accessed: April. 17, 2024).
- [10] Google. Firebase Authentication: Android Phone Authentication. Retrieved from <https://firebase.google.com/docs/auth/android/phone-auth> (accessed April. 17, 2024).


## REFERENCES

- [11] Android Studio, “Meet android studio : Android developers,” Android Developers, <https://developer.android.com/studio/intro> (accessed April. 17, 2024).
- [12] N. Naidu, “Software Engineering | Agile Software Development,” *GeeksforGeeks*, Jul. 12, 2018.  
<https://www.geeksforgeeks.org/software-engineering-agile-software-development/>  
(accessed Sep. 2, 2024).




Wholly owned by UTAR Education Foundation

## FACULTY OF INFORMATION COMMUNICATION AND TECHNOLOGY



---

### GoPerak

Discover the best food, attractions, and activities in Perak, Malaysia,  
with our easy-to-use Google Map-powered tourism search app." 

**OBJECTIVE**

- To provide streamlined features, focusing exclusively on tourism search
- To implement a balanced food priority
- To maintain credibility of user review

**PROPOSED METHOD**

**Tourism search focus**

- Purely dedicated to searching food, activities


**User Centric filtering**

- Categorization system for personalized search results based on user preferences

**Credibility-Driven Reviews**

- Users are required to perform facial recognition using IC during sign up

**TECH TOOLS USED**




**SYSTEM ARCHITECTURE**



```

graph LR
    Actor((Actor)) --> View[View]
    View --> Controller[Controller]
    Controller --> Model[Model]
    Model --> DB[(DB)]
    DB --> Model
    
```



**PROJECT DEVELOPER : WONG WEI YING**

**PROJECT SUPERVISOR: TS DR OOI CHEK YEE**