

MOBILE APPLICATION FOR F&B REVIEWS

BY

YONG JU WEI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

**BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION
SYSTEMS ENGINEERING**

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2022

REPORT STATUS DECLARATION FORM

Title: MOBILE APPLICATION FOR F&B REVIEWS

Academic Session: Jan 2022

I YONG JU WEI
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

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

Verified by,



(Author's signature)



(Supervisor's signature)

Address:

54, Persiaran Rapat Baru 4,
Medan Lapangan Legenda,
31350 Ipoh, Perak.

Ts. SAW SEOW HUI

Supervisor's name

Date: 21 April 2022

Date: 4/22/2022

Universiti Tunku Abdul Rahman			
Form Title : Sample of Submission Sheet for FYP/Dissertation/Thesis			
Form Number: FM-IAD-004	Rev No.: 0	Effective Date: 21 JUNE 2011	Page No.: 1 of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TUNKU ABDUL RAHMAN

Date: 21 April 2022_____

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that Yong Ju Wei (ID No: 18ACB04479) has completed this final year project/ dissertation/ thesis* entitled “ Mobile Application for F&B Reviews ” under the supervision of Ts Saw Seow Hui (Supervisor) from the Department of Computer Science, Faculty of Information and Communication Technology.

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

Yours truly,



(Yong Ju Wei)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled “**MOBILE APPLICATION FOR F&B REVIEWS**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.



Signature : _____

Name : YONG JU WEI

Date : 21 April 2022

ACKNOWLEDGEMENTS

I would like to express my thanks to my supervisor, Ms Saw Seow Hui who has guided and advised me throughout the development and implementation of this project. She has helped me navigate through this entire process of completing my project reports, preparing presentations as well as providing support when I needed it. I would also like to thank my moderator, Mr Tey Chee Chieh who has helped provide me with innovative ideas to improve my project. It is thanks to my supervisor and moderator that I was able to grow and learn throughout this project. Additionally, I would also like to thank my family members and close friends. Without their support and advice, I would not have been able to succeed in completing this project. In summary, I would like to extend my appreciation and gratitude to everyone who has helped and motivated me to work and complete this project.

ABSTRACT

When given a variety of options, many people often find themselves in difficult situations in which they have trouble deciding on what to choose. This is especially the case when it involves deciding what and where to have your next meal. Thus, this project involves the development of an Android mobile application for F&B Reviews, called EatMore. EatMore aims to provide a reliable solution that helps users in their decision-making process by offering a platform where they can search, review, and discover eateries. The application aims to provide users with a simple, yet informative review system that can appeal to those who are looking for concise reviews, detailed reviews, or both. The application also offers features aimed at Kampar, Perak. This aims to solve the problem in which reviews of an eatery are oftentimes spread out amongst different COPs. Thus, foodies can use this application for exploring the F&B options as well as keeping track of their food journey in Kampar. Besides that, the project adopts the Extreme Programming methodology for the development of the application where minimal set of functionalities are continuously developed and refactored until all core and additional functionalities are eventually achieved. For this project, the application is programmed in Java and relies on Firebase and Google Places API for major features such as the search and review modules. EatMore was successfully developed in this project, meeting the objectives of the project.

TABLE OF CONTENTS

TITLE PAGE	i
REPORT STATUS DECLARATION FORM	ii
FYP THESIS SUBMISSION FORM	iii
DECLARATION OF ORIGINALITY	iv
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	x
LIST OF TABLES	xiii
LIST OF ABBREVIATIONS	xvi
CHAPTER 1 INTRODUCTION	1
1.1 Problem Statement and Motivation	1
1.2 Objectives	2
1.3 Project Scope and Direction	2
1.4 Contributions	4
1.5 Report Organization	5
CHAPTER 2 LITERATURE REVIEW	7
2.1 Yelp [8]	7
2.1.1 Brief Introduction to Yelp	7
2.1.2 Features and Functionalities of Yelp	7
2.1.3 Strengths and Weaknesses of Yelp	8
2.2 Burpple [10]	9
2.2.1 Brief Introduction to Burpple	9
2.2.2 Features and Functionalities of Burpple	9
2.2.3 Strengths and Weaknesses of Burpple	11
2.3 Zomato [11]	11
2.3.1 Brief Introduction to Zomato	11
2.3.2 Features and Functionalities of Zomato	11
2.3.3 Strengths and Weaknesses of Zomato	13

2.4 Features Comparison of Reviewed Applications	13
CHAPTER 3 SYSTEM METHODOLOGY	14
3.1 System Methodology	14
3.2 Project Workflow	15
CHAPTER 4 SYSTEM DESIGN	16
4.1 System Block Diagram	16
4.2 System Architecture Diagram	23
4.3 Storyboard	25
4.4 Use Case	27
4.4.1 Use Case Diagram	27
4.4.2 Use Case Descriptions	28
4.5 Activity Diagram	45
4.6 Database Design	59
CHAPTER 5 SYSTEM IMPLEMENTATION	60
5.1 Technologies and Tools Involved	60
5.1.1 Hardware Setup	60
5.1.2 Software Setup	60
5.2 Project Timeline	68
5.3 System UI	69
CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION	72
6.1 System Testing and Performance Metrics	72
6.1.1 Testing Setup and Result	72
6.2 Implementation Issues and Challenges	85

6.3 Objectives Evaluation	86
CHAPTER 7 CONCLUSION AND RECOMMENDATION	87
7.1 Conclusion	87
7.2 Future Work	87
REFERENCES	88
APPENDICES	90
APPENDIX A: WEEKLY REPORTS	90
POSTER	97
PLAGIARISM CHECK RESULT	98
FYP2 CHECKLIST	102

LIST OF FIGURES

Figure Number	Title	Page
Figure 2.1.1.1	Yelp Logo	7
Figure 2.1.2.1	Yelp List Results	8
Figure 2.1.2.2	Yelp Map Results	8
Figure 2.2.1.1	Burpple Logo	9
Figure 2.2.2.1	Burpple Search Filters	10
Figure 2.2.2.2	Burpple Levels	10
Figure 2.3.1.1	Zomato Logo	11
Figure 2.3.2.1	Zomato 3-star Review	12
Figure 2.3.2.2	Zomato 5-star Review	12
Figure 3.1.1	Prototyping Model SDLC	14
Figure 4.1.1	User Authentication Flow	16
Figure 4.1.2	Search Flow	17
Figure 4.1.3	Eatery Profile Flow	18
Figure 4.1.4	View Review Flow	19
Figure 4.1.5	User Profile Flow	20
Figure 4.1.6	View User Review Flow	21
Figure 4.1.7	View User Favourites Flow	21
Figure 4.1.8	Manage Account and Forgot Password Flow	22
Figure 4.2.1	System Architecture Diagram	23
Figure 4.3.1	Storyboard	25
Figure 4.3.2	Storyboard	26
Figure 4.4.1.1	Use Case Diagram	27
Figure 4.5.1	Log In Activity Diagram	45
Figure 4.5.2	Log Out Activity Diagram	45
Figure 4.5.3	Sign Up Activity Diagram	46
Figure 4.5.4	Forgot Password Activity Diagram	46
Figure 4.5.5	Get User Location Activity Diagram	47
Figure 4.5.6	Search Eatery Activity Diagram	47
Figure 4.5.7	Sort & Filter Search Results Activity Diagram	48

Figure 4.5.8	View Eatery Profile Activity Diagram	48
Figure 4.5.9	Call Eatery Activity Diagram	49
Figure 4.5.10	Get Directions To Eatery Activity Diagram	49
Figure 4.5.11	Favourite Eatery Activity Diagram	49
Figure 4.5.12	Write Review Activity Diagram	50
Figure 4.5.13	Add Photos Activity Diagram	50
Figure 4.5.14	Submit Review Activity Diagram	51
Figure 4.5.15	View Review Activity Diagram	52
Figure 4.5.16	Rate Review Activity Diagram	52
Figure 4.5.17	Edit Review Activity Diagram	53
Figure 4.5.18	Delete Review Activity Diagram	53
Figure 4.5.19	View User Profile Activity Diagram	54
Figure 4.5.20	View User Reviews Activity Diagram	54
Figure 4.5.21	View User Favourites Activity Diagram	55
Figure 4.5.22	Unfavourite Eatery Activity Diagram	55
Figure 4.5.23	View User Lists Activity Diagram	56
Figure 4.5.24	Create List Activity Diagram	56
Figure 4.5.25	View List Activity Diagram	57
Figure 4.5.26	Edit List Activity Diagram	57
Figure 4.5.27	Delete List Activity Diagram	58
Figure 4.5.28	Manage Account Activity Diagram	58
Figure 5.1.2.1	Android Studio Version Details	60
Figure 5.1.2.2	Google Cloud Logo	61
Figure 5.1.2.3	EatMore Cloud Console Dashboard	61
Figure 5.1.2.4	EatMore API Keys	61
Figure 5.1.2.5	Google Maps Platform Logo	62
Figure 5.1.2.6	Place Details Method 1	63
Figure 5.1.2.7	Place Details Method 2	63
Figure 5.1.2.8	Firebase Logo	64
Figure 5.1.2.9	Cloud Firestore Logo	64
Figure 5.1.2.10	Firebase Authentication Logo	64
Figure 5.1.2.11	Cloud Storage Logo	65
Figure 5.1.2.12	EatMore Cloud Storage Console	65

Figure 5.1.2.13	Visual Paradigm Online File	66
Figure 5.1.2.14	EatMore Figma Project	66
Figure 5.1.2.15	EatMore Canva Project	67
Figure 5.1.2.16	TeamGantt Logo	67
Figure 5.2.1	Project Timeline	68
Figure 5.3.2	Log In	69
Figure 5.3.3	Sign Up	69
Figure 5.3.4	Forgot Password	69
Figure 5.3.5	Forgot Password (Email Sent)	69
Figure 5.3.6	Get Device Location	69
Figure 5.3.7	Input Manual Location	69
Figure 5.3.8	Home	69
Figure 5.3.9	Search	69
Figure 5.3.10	Search Results	69
Figure 5.3.11	No Search Results	70
Figure 5.3.12	Eatery Profile General Section	70
Figure 5.3.13	Eatery Profile Review Section	70
Figure 5.3.14	Eatery Profile with No Reviews	70
Figure 5.3.15	Add to List Dialog	70
Figure 5.3.16	Review Form	70
Figure 5.3.17	Review Form	70
Figure 5.3.18	View Own Review	70
Figure 5.3.19	View Other User Review	70
Figure 5.3.20	Create List	71
Figure 5.3.21	User Lists	71
Figure 5.3.22	View List	71
Figure 5.3.23	View List (No items)	71
Figure 5.3.24	Favourites (No favourites)	71
Figure 5.3.25	User Favourites	71
Figure 5.3.26	Manage Account	71
Figure 6.1.1	Black Box Testing	72

LIST OF TABLES

Table Number	Title	Page
Table 2.4.1	Features Comparison of Reviewed Applications	13
Table 4.4.2.1	Log In Use Case Description	28
Table 4.4.2.2	Log Out Use Case Description	29
Table 4.4.2.3	Sign Up Use Case Description	30
Table 4.4.2.4	Forgot Password Use Case Description	30
Table 4.4.2.5	Get User Location Use Case Description	31
Table 4.4.2.6	Get Device Location Use Case Description	31
Table 4.4.2.7	Input Location Manually Use Case Description	32
Table 4.4.2.8	Search Eatery Use Case Description	32
Table 4.4.2.9	Sort & Filter Search Results Use Case Description	32
Table 4.4.2.10	View Eatery Use Case Description	33
Table 4.4.2.11	Call Eatery Use Case Description	33
Table 4.4.2.12	Get Directions to Eatery Use Case Description	34
Table 4.4.2.13	Favourite Eatery Use Case Description	34
Table 4.4.2.14	Add Eatery to List Use Case Description	35
Table 4.4.2.15	Write Review Use Case Description	35
Table 4.4.2.16	View Review Use Case Description	36
Table 4.4.2.17	Rate Review Use Case Description	36
Table 4.4.2.18	Edit Review Use Case Description	37
Table 4.4.2.19	Delete Review Use Case Description	37
Table 4.4.2.20	View User Profile Use Case Description	38
Table 4.4.2.21	View User Reviews Use Case Description	38
Table 4.4.2.22	View User Favouritess Use Case Description	38
Table 4.4.2.23	Unfavourite Use Case Description	39
Table 4.4.2.24	View User Lists Use Case Description	39
Table 4.4.2.25	Create List Use Case Description	40
Table 4.4.2.26	View List Use Case Description	40
Table 4.4.2.27	Edit List Use Case Description	41
Table 4.4.2.28	Delete List Use Case Description	41

Table 4.4.2.29	Manage Account Use Case Description	41
Table 4.4.2.30	Add Dining Type Use Case Description	42
Table 4.4.2.31	Rate Eatery Use Case Description	42
Table 4.4.2.32	Add Photos Use Case Description	42
Table 4.4.2.33	Add Tags Use Case Description	43
Table 4.4.2.34	Write Text Review Use Case Description	43
Table 4.4.2.35	Submit Review Use Case Description	44
Table 4.6.1	Document Structures	59
Table 5.1.1.1	Laptop Specification	60
Table 6.1.1.1	Log In Use Case Testing	72
Table 6.1.1.2	Log Out Use Case Testing	72
Table 6.1.1.3	Sign Up Use Case Testing	73
Table 6.1.1.4	Forgot Password Use Case Testing	73
Table 6.1.1.5	Get User Location and Get Device Location Use Case Testing	74
Table 6.1.1.6	Input Location Manually Use Case Testing	74
Table 6.1.1.7	Search and Sort & Filter Search Results Use Case Testing	75
Table 6.1.1.8	View Eatery Profile Use Case Testing	76
Table 6.1.1.9	Call Eatery Use Case Testing	76
Table 6.1.1.10	Get Directions to Eatery Use Case Testing	76
Table 6.1.1.11	Favourite and Unfavourite Eatery Use Case Testing	77
Table 6.1.1.12	Add Eatery To List Use Case Testing	77
Table 6.1.1.13	Write, Edit and Submit Review Use Case Testing	79
Table 6.1.1.14	View Review Use Case Testing	79
Table 6.1.1.15	Rate Review Use Case Testing	80
Table 6.1.1.16	Delete Review Use Case Testing	80
Table 6.1.1.17	View User Profile Use Case Testing	81
Table 6.1.1.18	View User Reviews Use Case Testing	81
Table 6.1.1.19	View User Favourties Use Case Testing	81
Table 6.1.1.20	View User Lists Use Case Testing	82
Table 6.1.1.21	Create List and Edit List Use Case Testing	82
Table 6.1.1.22	View List Use Case Testing	83

Table 6.1.1.23	Delete List Use Case Testing	83
Table 6.1.1.24	Manage Account Use Case Testing	84
Table 6.1.1.25	Add Photos Use Case Testing	84
Table 6.3.1	Objectives Evaluation	86

LIST OF ABBREVIATIONS

API	Application Programming Interface
AVD	Android Virtual Device
CLI	Command Line Interface
COP	Consumer Opinion Platform
F&B	Food & Beverage
IDE	Integrated Development Environment
SDK	Software Development Kit
SDLC	Software Development Life Cycle
SVG	Scalable Vector Graphics
UI	User Interface
XML	Extensible Markup Language
XP	Extreme Programming

CHAPTER 1 INTRODUCTION

1.1 Problem Statement and Motivation

i) Paradox of Choice - the struggle of choosing

With the abundance of new businesses in the F&B industry, came a problem called the “Paradox of Choice”, a theory proposed by researcher and author, Barry Schwartz. According to Schwartz [1], a wide range of choices can create a psychological burden on an individual as more time and effort is necessary to browse through the options and decide on one that can both satisfy needs and maximize value. This problem affects most consumers especially in terms of trying to decide what to eat. The difficulty in deciding is because the choice made should be worthwhile and entail a good dining experience, so more care is put into choosing where to eat especially for those who are live to eat, rather than just eating to live. The struggle is common to urban workers, students, and even families where eating-out has become a trend as they do not have food at home [2].

This problem is especially evident now amidst the COVID-19 pandemic where most people are stuck in their homes under lockdown. With the new norm being social distancing, food delivery has become more popular than ever before. This is especially so with individuals who do not usually cook or know how to as well as for those who may be trying to have as little contact with others as possible. More restaurants who never offered the service before are hopping on the trend to sustain their businesses [3]. This gives consumers more options to choose from than ever before due to the availability of vast F&B selections.

ii) Business disruptions due to Covid-19

The global Covid-19 pandemic has forced the world to enter lockdown after lockdown which caused businesses and institutions to physically close. This is the same for Kampar, Perak as well where university students were sent home to prevent the spread of the virus. As such, the bustling university town became quiet with most of its student residents having left. With the lockdowns and a large segment of consumers gone, many businesses in Kampar were severely affected as reported in a newspaper article where about 40 business premises were temporarily closed in April 2019. During this quiet period, many businesses suffered while some moved around, essentially evolving the town in different ways [4]. At present, with the lockdown lifted and physical classes back in session, many students are returning to Kampar to find that the changed town.

This application would be a great tool for both newcomers and returning students to rediscover Kampar and what it has to offer in terms of its F&B. With the application, local businesses would also be able to gain traction to their stores and reintroduce themselves to the incoming student.

iii) Too many different COP platforms

Consumers resort to crowd-sourced reviews to help them make better decisions to solve this problem. As more businesses have started moving to online platforms, consumers are able to source out reviews from different COPs. However, the problem that arises is that different businesses tend to use different platforms. For example, one business may actively use Facebook while another may solely market on Yelp. In such a scenario, the likelihood of the number of reviews the business has on just one platform being low is much higher, which is counterproductive as consumers tend to stay on the one platform, they found the business on to find out more about it. According to a study carried out by Northwestern University's Spiegel Research Center, a product with five reviews has a 270% greater purchase likelihood than a product with no reviews [5]. Similarly, the quantity of customer reviews a business has can affect consumer's perception on the business in their process of information search. Thus, with there being so many different platforms, the visibility of each review is lower.

1.2 Objectives

Main objective:

The main objective of this project is to create a reliable and easy-to-use Android mobile application for F&B reviews that can aid users in deciding where and what to eat.

Sub-objectives:

1. To develop a platform where users can browse and discover the F&B of Kampar, Perak
2. To provide users with a centralized COP for the F&B in Kampar, Perak

1.3 Project Scope and Direction

The primary focus of the project is to develop **EatMore**, a F&B reviews Android mobile application that allows users to write and read reviews of patrons' experience and opinions of the eatery. The application features two major features which are the

search and review function as well as additional minor features that are all described below.

i) Search & Discover

For the search function, users can sort, and filter search results based on criteria like price level, distance or whether the establishment is currently open. This allows users to browse and search for more relevant eateries in terms of convenience and proximity. Compared to other similar works, this project will aim to focus on the eateries in Kampar, Perak.

ii) Add and Edit Reviews

For the review function, users can assess an eatery on its F&B quality, cleanliness, and customer service using a 5-star system which will be cumulated and averaged to give the eatery an overall 5-star rating. Users will also need to select the type of dining they had, i.e., dine in or delivery. This is because of how different each experience can entail. In addition, each review also includes optional inputs such as text review, tags, and photos. Users can also edit or remove reviews, allowing them to change their minds on previous opinions.

iii) View Review

Users can also view the full reviews of other users and rate the reviews to be “Helpful” or “Funny”. The author of the review will also be able to view what impact their reviews has had if any. This encourages user engagement to the application.

iv) Eatery Profile

In the eatery profile, users can view the business information and reviews of the eatery. Business information includes the address, contact number, price level, opening hours and website if available. If the eatery contact number is available, users can call the number from the profile. Users can also get directions to the eatery where the application will navigate them to Google Maps directions to the eatery. Users can favourite or add this eatery to a list. Additionally, users can view the average rating for overall, cleanliness, F&B, and service.

v) User Profile

Each user can favourite eateries as well as create and edit lists of eateries which can all be viewed and managed from the user profile. Users can also view all their reviews in their profile. Users can change their username and profile picture. The username will be validated on its availability.

vi) Log In and Sign Up

Users can create an account using email and password. Each user is assigned a unique ID. Alternatively, users who forgot their password, can reset their password through a link that is sent to the mail they signed up with.

vii) Get User Location

By default, the system would request user's permission for access to the device's location and get the location of the device. Alternatively, users can manually enter their location. The coordinates of the user's location are used for displaying suggestions for eateries and for the search function.

1.4 Contributions

Trustworthy reviews can help consumers save valuable time and money as it becomes easier to discern whether a specific business is good or not. The process of searching for a decent or good restaurant to eat at will be sped up and become simpler. Other than consumers, businesses also profit from customer reviews as they can essentially make or break the business. This is because a business' online review can strengthen the business' credibility and gain consumer trust [6]. They can analyse the reviews they receive and reflect upon them to revise their business strategies from the reviews made on the application. Therefore, the benefits and overall utility of reviews reinforces its importance and practice.

The goal of this project is to create a mobile application that will make it easier to write reviews that are both helpful and informative to others, concise and simple. One of the key aspects that achieves this is the input of ratings for specific aspects of customers' experience, namely cleanliness, F&B, and customer service. There are numerous factors that can influence consumer's choice of eatery such as physical appearance, physical environment, food variety, nutrition et cetera. However, these three factors were generalized from the many factors to keep the review concise but informative. This is evident as proven in a study and analysis which identifies the

importance and performance of restaurant attributes which affect customers' restaurant choice, specifically food taste, service quality and physical environment, while other factors which were determined to be of lower priority to customers [7].

Furthermore, what makes this application different from others is that it will offer users with browsing categories for areas in Kampar, Perak specifically as the project aims to focus on eateries in the town specifically per the project objectives. This serves to solve the problem in which there are so many different existing COPs, so for users who want to find out more about eateries in Kampar specifically, this would be the app for them. In other words, this application would be useful to both locals as well as tourists who are looking to discover more about the F&B culture and options in the town or the areas surrounding it.

1.5 Report Organization

This report is organized into 7 chapters: Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Methodology, Chapter 4 System Design, Chapter 5 System Implementation, Chapter 6 System Evaluation and Discussion, and Chapter 7 Conclusion and Recommendation.

1. Chapter 1 Introduction introduces Eat More, and discusses the problem statement and motivation, project objectives, project scope, project contributions as well as report organization.
2. Chapter 2 Literature Review is the literature review and features comparison of existing relevant applications. To better understand and study existing solutions towards the problem domain, three popular COP mobile applications were selected and analysed, Yelp, Burpple and Zomato. The perceived strengths and weaknesses of each reviewed application is also briefly discussed.
3. Chapter 3 System Methodology explains the methodology implemented as well as the general workflow in this project.
4. Chapter 4 System Design describes and shows the system designs of the project, namely the system block diagram, system architecture diagram, storyboard, use case diagram and descriptions, activity diagrams, and database design.
5. Chapter 5 System Implementation discusses the tools and technologies involved in the implementation of the project. This chapter also shows the project timeline as well as the screenshots of the UI for the final product of the project.

6. Chapter 6 System Evaluation and Discussion explains how the application is tested and shows the results of the results of the tests as well as the issues and challenges faced in the implementation. The project objectives are also evaluated in this chapter.
7. Chapter 7 Conclusion and Recommendation is the conclusion and summarizes all previous chapters. A brief analysis of the project's strengths, weaknesses and recommendations for the future are discussed.

CHAPTER 2 LITERATURE REVIEW

2.1 Yelp [8]

2.1.1 Brief Introduction to Yelp



Figure 2.1.1.1 Yelp Logo

Founded in the year 2004, Yelp is a successful online customer review site. The platform acts like an online directory in which users can search and come across local businesses of different industries such as spas, restaurants, cafes, bars, pharmacies, shopping, and everything else. Yelp is available in 35 countries across the world including Malaysia.

2.1.2 Features and Functionalities of Yelp

The core feature of Yelp is its review and rating system. Users of the platform can rate and write feedbacks of places. A 5-star rating system is implemented for the rating. Users can write lengthy reviews or short tips about a business and share multiple photos in their reviews. If the review written by users is very short, Yelp automatically considers it to be a tip. Yelp tips are typically only short descriptions of the business whereas reviews are several paragraphs longer and more detailed [9]. Furthermore, users can edit or delete their reviews as well as rate, comment and share other reviews.

When viewing a business on the platform, the page will display the business' average rating, price range, cuisine type, opening hours, location, contact details as well as amenities and features if any. Users can then also read reviews written by other fellow Yelp users. The application also provides sorting options for a business' reviews. For instance, the reviews can be sorted based on the date of review, ratings, reviews by Yelp Elites and Yelp's own special sort. Furthermore, the platform also provides a search engine for a business' reviews alone. Users can search for reviews of a business that mentions a specific word or phrase.

For the search feature, users can look for businesses near them using the application's location-based search feature. The application will then provide suggestions and recommendations based on the user's geolocation. Other than searching near the user's geolocation, users can search for businesses of specific

categories such as plumbing, café, hairdressers, or even places that offer promotions for Yelp check-ins. Various filtering and sorting options can also be applied to the search such as price range, distance, amenities, ambience as well as payment options. The search results can be listed down in a list or on a map as seen below.

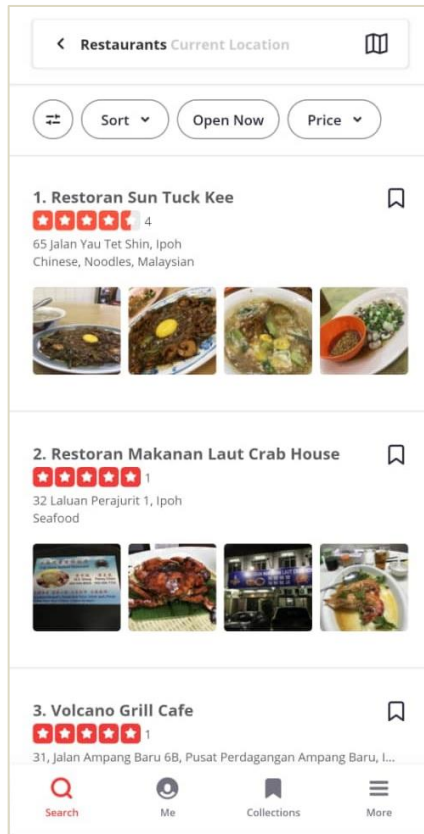


Figure 2.1.2.1 Yelp List Results



Figure 2.1.2.2 Yelp Map Results

To create an account on the platform, users register with their personal email, Google email or Facebook social media account which makes the registration process simple and quick. Users can opt to register for a business account or a personal use account. Normal account users can add a business into the site if the relationship with the business is clarified. Additionally, it is necessary for users to permit and turn on their location services to use the application. Otherwise, an error page will be displayed.

2.1.3 Strengths and Weaknesses of Yelp

Its diverse and comprehensive filtering and sorting options provided for not only the general business search, but for reviews as well is a very helpful feature for users. In addition, the search engine for reviews also allows users to filter reviews effectively and efficiently. Furthermore, other users can rate reviews based on whether they were

“Useful”, “Funny” or “Cool”, which can be helpful in encouraging other users to read the review.

Although Yelp is thorough in its search filters, the review and rating system itself is very general and not specific. Users are prompted to write a text and rate the business, and maybe include a picture or two. This may cause important points in the review users are looking for to be hidden in a long text. Furthermore, the application does not allow manual location input. This may deter users who do not want to permit the application’s use of the user’s geolocation information.

2.2 Burpple [10]

2.2.1 Brief Introduction to Burpple



Figure 2.2.1.1 Burpple Logo

Burpple is a platform that initially started out as a social food journal in 2021 which expanded its features in 2015, becoming a COP platform as well. The platform is primarily focused on eateries from Kuala Lumpur and Singapore, but users can add other businesses manually from other places.

2.2.2 Features and Functionalities of Burpple

One of the main features of this application is to write reviews about eateries. Compared to other applications, users do not access the review form from the eatery profile but instead access it from the home page or bottom navigation bar. The reviews include a picture, title, description, location and one of the user’s personal lists. Burpple lists are lists that user can create to group the reviews they write. A user can create multiple lists and include the same review in more than one list. Lists can also include reviews written by other users. The location and description of the review are optional. Users can set the location as private and exclude a description for the review as the application mainly serves as a food journal, so a user review does not necessarily require a specific location. Users can also modify and delete their reviews. Furthermore, users can like, comment, and share reviews including their own.

For its search functions, Burpple has a location-based search which finds eateries around the user's geolocation. The search feature also comes with filters and sorting options as seen in Figure 2.2.2.1. Like Yelp, the search results can be viewed as pins or in a list. Users can also explore specific categories and neighbourhoods depending on the user's city selection setting. Similarly, users can also browse through specially curated Burpple Guides based on the city. New guides are frequently created and shared by Burpple which can also engage users with new content.

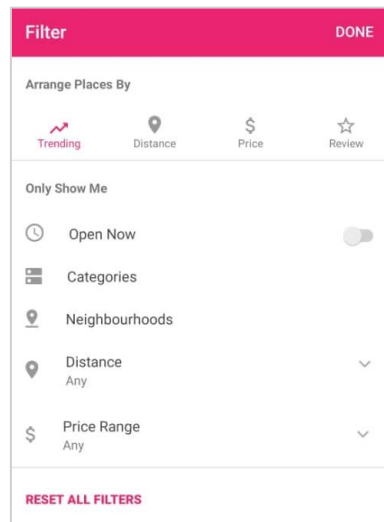


Figure 2.2.2.1 Burpple Search Filters

To create a Burpple account, users are offered several options such as Google sign-in, Facebook login, or personal email login. Users can use the application without having to turn on their location services. However, the application will not be able to suggest nearby places for the user, but aside from that, users will generally be able to fully make use of the app's features. Besides that, Burpple also implements a gamification feature where users can level up based on the number of reviews they contribute to the platform.

Level	Number of Reviews
1	0
2	≥ 1
3	≥ 5
4	≥ 20
5	≥ 50
6	≥ 100
7	≥ 200
8	≥ 500
9	≥ 1000
10	≥ 2000

Figure 2.2.2.2 Burpple Levels

2.2.3 Strengths and Weaknesses of Burpple

Burpple's curated guides and special categories for vicinities in the selected city is an interesting feature. This can both suggest more relevant places to users and keep users engaged with the platform through browsing the different lists. Through this feature, users are provided with a more personalized feed in a way as the suggestions are more relevant to them in terms of location.

However, the application does not provide basic rating options which is generally very important as it acts as an immediate indicator to consumers about the quality of the F&B and service of an eatery. Furthermore, users can only include one picture for each review. If users were to want to share more pictures, they would have to create several reviews which is both inconvenient and troublesome. Thus, discouraging the act altogether which would be considered a loss as other users would miss out on the pictures.

2.3 Zomato [11]

2.3.1 Brief Introduction to Zomato



Figure 2.3.1.1 Zomato Logo

Initially founded as Foodiebay in 2008, Zomato was renamed on 18 January 2010 [12]. It is an online food delivery and review application. The application is available in 24 countries including Malaysia.

2.3.2 Features and Functionalities of Zomato

Zomato's search functions are location-based. Users need to manually enter their locations or turn on their phone's geolocation settings. Users can search for restaurant name, cuisine or specific dishes using a wide variety of sort and filters. For instance, eateries near the user, average rating, price range for two persons, et cetera. Like Burpple, Zomato also has curated collections which users can use to discover new places. When viewing a restaurant, various type of information is displayed in different tabs, namely menu items, business information as well as reviews and ratings. In the reviews section, users can view two different types of ratings for delivery and for

dining. From there, users can read the business' reviews where various sort and filter options are provided. For example, users can sort the reviews according to the popularity, rating, and dates of the review as well as search keywords from existing reviews. Users can rate as “Helpful”, like, comment, and share reviews as well as choose to edit or delete their own. Additionally, Zomato displays the most used tags and keywords in the review highlights.

Zomato implements a review and rating system in which users must include the type of dining and a rating out of 5-stars while they can opt to write text reviews of minimum 100 characters, insert tags, and add pictures. Zomato tags are a form of summarized reviews that was introduced as a new content currency in the platform [13]. The input for Zomato's form is dynamic and depends on the user input star rating. For instance, users can set positive and negative tags for 3-star ratings while users are prompted to select positive tags for 5-star ratings as seen in Figure 2.3.2.1 and Figure 2.3.2.2 respectively.

Figure 2.3.2.1 Zomato 3-star Review

Figure 2.3.2.2 Zomato 5-star Review

To create an account, users can use their phone number, personal email, Google email or Facebook account. When users launch the application, they will be prompted to permit their device's location services or manually input their location instead. Without location details, the application does not allow users to proceed. Other than that, Zomato previously implemented a user level feature based on a foodie points system. However, in recent updates, the gamification feature was removed.

2.3.3 Strengths and Weaknesses of Zomato

One of Zomato's strengths is its dining type and tags in reviews. This helps users determine how the business does in terms of providing dining services or delivery. Furthermore, tags are a great substitute for lengthy reviews that some may find boring and irrelevant to them. This allows users to write reviews quickly and effectively as they can now opt to not write long text reviews. The application's algorithm in tag predictions and suggestions also helps speed up the process of creating a user review. The application's overall UI design is also very neat and clean while being able to display a lot of information.

However, the text review for Zomato requires users to write a minimum of 100 characters for their review. This can discourage users from writing reviews as they can feel somewhat pressured to have to write enough points and details that meets the character minimum limit.

2.4 Features Comparison of Reviewed Applications

App		Yelp	Burpple	Zomato
Features				
Review Form	5-star rating	√		√
	Review description	√	√	√
	Pictures	√	√	√
	Dining type			√
	Tags			√
View reviews	Search keyword	√	√	√
	Sort and Filter	√	√	√
	Like and Rate	√	√	√
	View count			
Search & discover	Cuisine Category	√	√	√
	Vicinity Category		√	
	Map and List Results	√	√	
	Sort and Filter	√	√	√
Account	Email	√	√	√
	Google sign-in	√	√	√
	Facebook sign-in	√	√	
	Level system based on user reviews		√	

Table 2.4.1 Features Comparison of Reviewed Applications

CHAPTER 3 SYSTEM METHODOLOGY

3.1 System Methodology

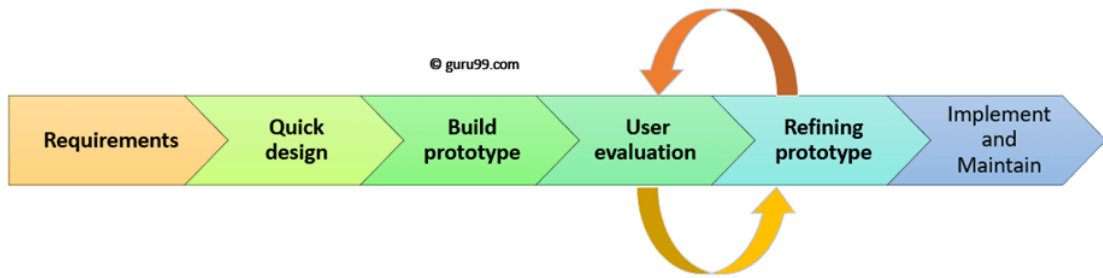


Figure 3.1.1 Prototyping Model SDLC

For this project, the prototyping methodology is implemented where the prototype is built, tested, and reworked until an acceptable prototype is obtained. The prototyping model is a popular software methodology that consists of six phases as seen in the figure above. This methodology is used when the exact requirements of the project are not known yet [14]. The prototyping method involves an iterative trial and error process. As this project is carried out by a single developer, this methodology is suitable as it helps identify any missing functionality in the system, gain a better understanding of user requirements, encourages flexibility in the design and is an overall straightforward model that is easy to understand and carry out. Furthermore, errors can be detected much earlier thereby saving a lot of effort and time which is suitable for the project's time constraints.

The methodology involves six phases, namely requirements, quick design, build prototype, user evaluation, refining prototype as well as implement and maintain. In the requirement phase, a requirement analysis is carried out to identify and define the requirements and expectations of the system. Then, a quick design of the system is created to get a brief idea of the system. Next, the actual prototype is designed based on the acquired information on requirements and quick design. In the next two phases, the prototype is repeatedly evaluated and refined until all the system requirements are met. Finally, the prototype is developed into a final system where it is thoroughly tested and prepared for production [15].

3.2 Project Workflow

1. Requirements

In this phase, a requirement analysis was carried out to identify the system requirements and expectations based on the project objectives, scope, and direction. To identify general requirements, similar applications were reviewed as described in CHAPTER 2 where the most common features and functions were identified.

2. Quick Design

In this phase, simple designs of the system were created. The draft designs were used to develop a better understanding of the system requirements as well as to identify any missing features that were not identified in the requirement analysis. The deliverables of this phase include the block diagrams, storyboard, use case diagram and activity diagrams.

3. Build Prototype

In this phase, a working prototype was developed based on the quick designs of the previous phase.

4. User Evaluation

In this phase, the built prototype was evaluated by different users. The feedback and criticisms from users were then passed forward to the next phase.

5. Refining Prototype

In this phase, the prototype is refined and refactored based on the feedback from user evaluation. Then, the refined prototype is then evaluated again, iterating the User Evaluation and Refining Prototype phase until the final prototype meets all the system requirements.

6. Implement Product and Maintain

In this phase, the final prototype was thoroughly tested. This was carried out by performing use case testing which is further discussed in CHAPTER 6

CHAPTER 4 SYSTEM DESIGN

4.1 System Block Diagram

Visual Paradigm Online Free Edition

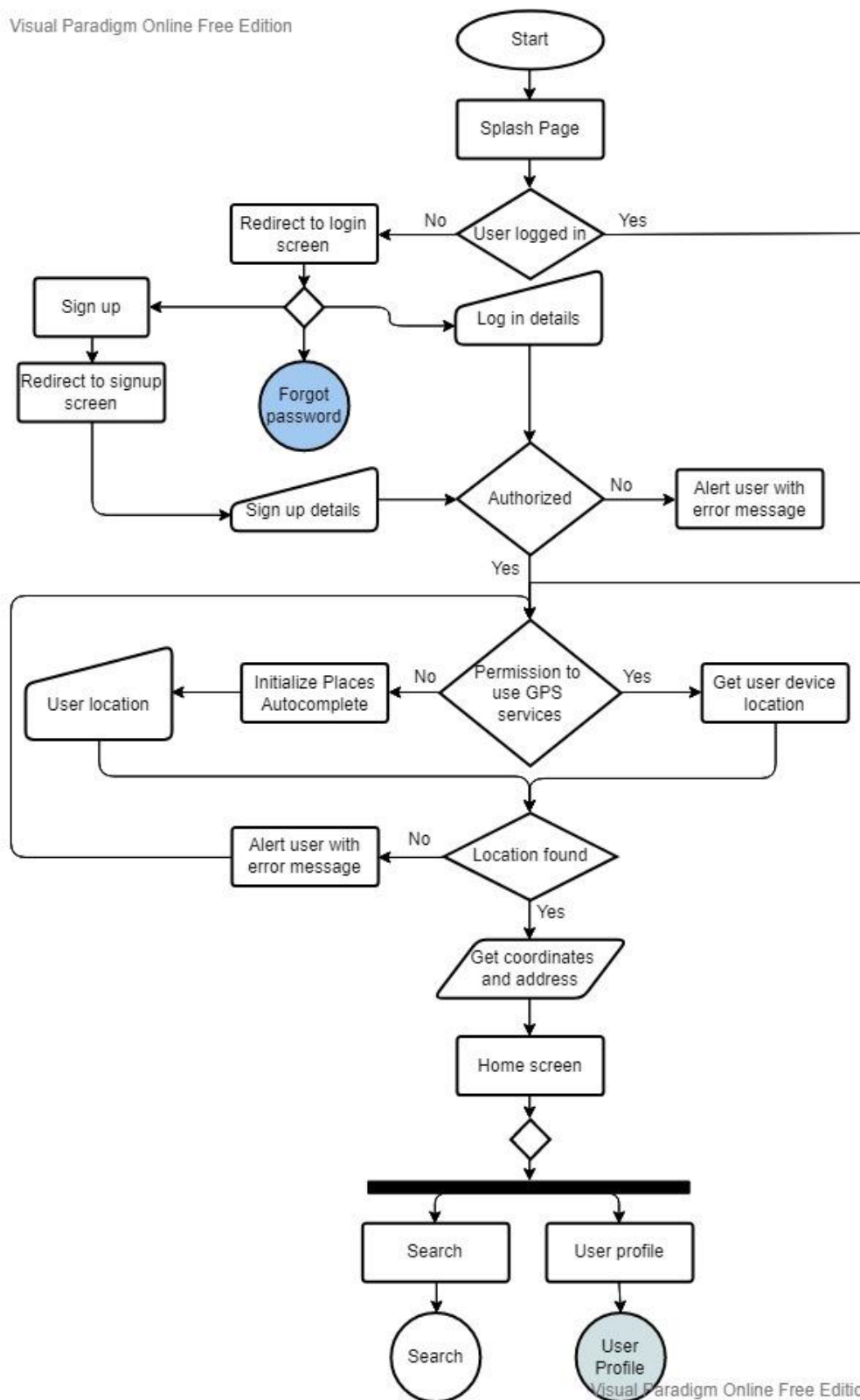


Figure 4.1.1 User Authentication Flow

On application launch, users are first directed to the login screen where they can choose to login, signup or if they have forgotten their password and would like to reset it, choose forgot password. If users are already logged in, they are redirected to the get location screen where they are prompt for location. If users want to sign up, they are brought to the Registration screen where they can input their signup details and register an account. Alternatively, users who do have an account can login by entering their correct credentials. Once users are authenticated on either login or signup, they are then prompt for their location. The system will request users to allow permission for location access. If users allow, the system will then retrieve the user device's current location coordinates and address. Alternatively, users who deny the permission will be prompt to enter their location manually. This feature is implemented by using Place Autocomplete. Similarly, one the user's location is retrieved, the system will get the coordinates and address. Then, users are redirected to the home screen where they can either search and browse or go to their user profiles.

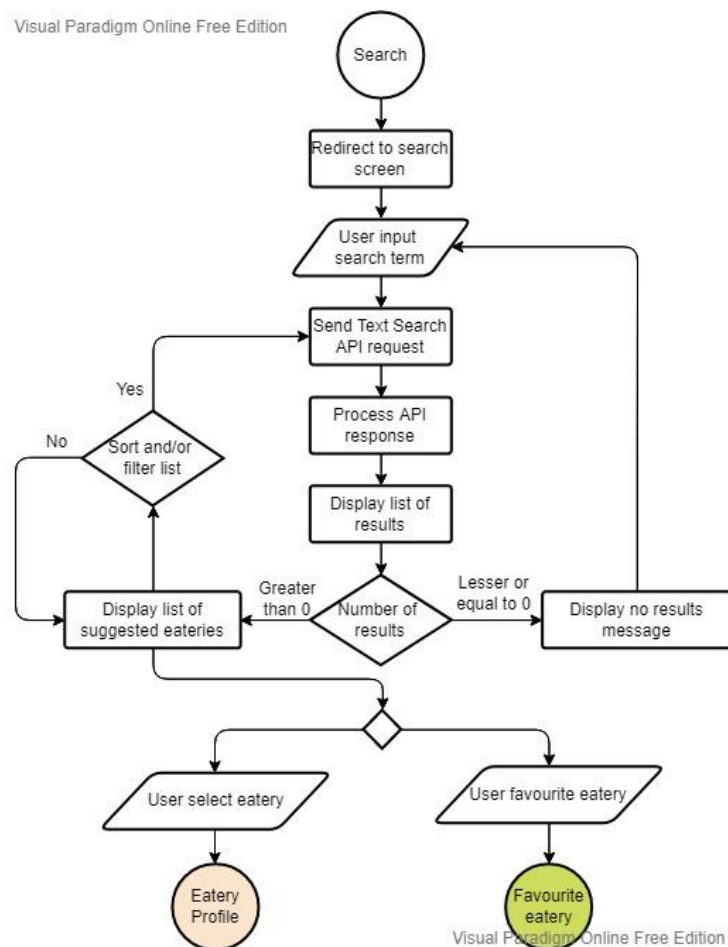


Figure 4.1.2 Search Flow

On search, user search queries are sent to Places API Text Search. The API response is then processed and displayed in the UI. For any filters that users select, a new API request is made again. Users can select search result items to view the profile page as well as tap on the favourite icons to favourite the eatery.

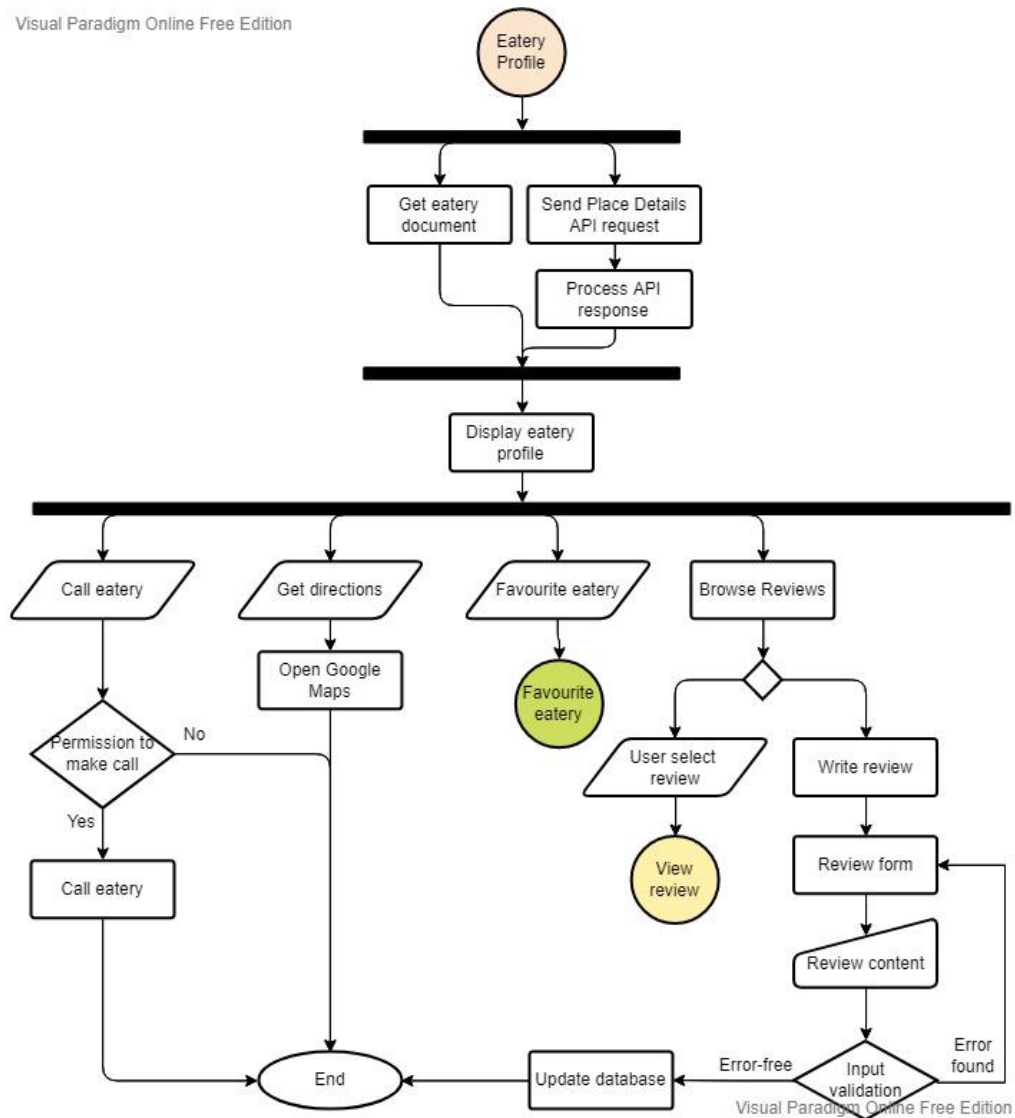


Figure 4.1.3 Eatery Profile Flow

When users open an eatery’s profile, the system will fetch the eatery document from the database and send a Place Details API request. Both responses are processed and displayed in the profile. For call eatery, users are first prompt for permission for the application to make calls. If users allow the call the system will make a call to the eatery’s contact number. Otherwise, the call will not be made. For getting directions, Google Maps will be opened with the eatery as the destination. For eatery reviews, users can view reviews as well as write reviews for the eatery.

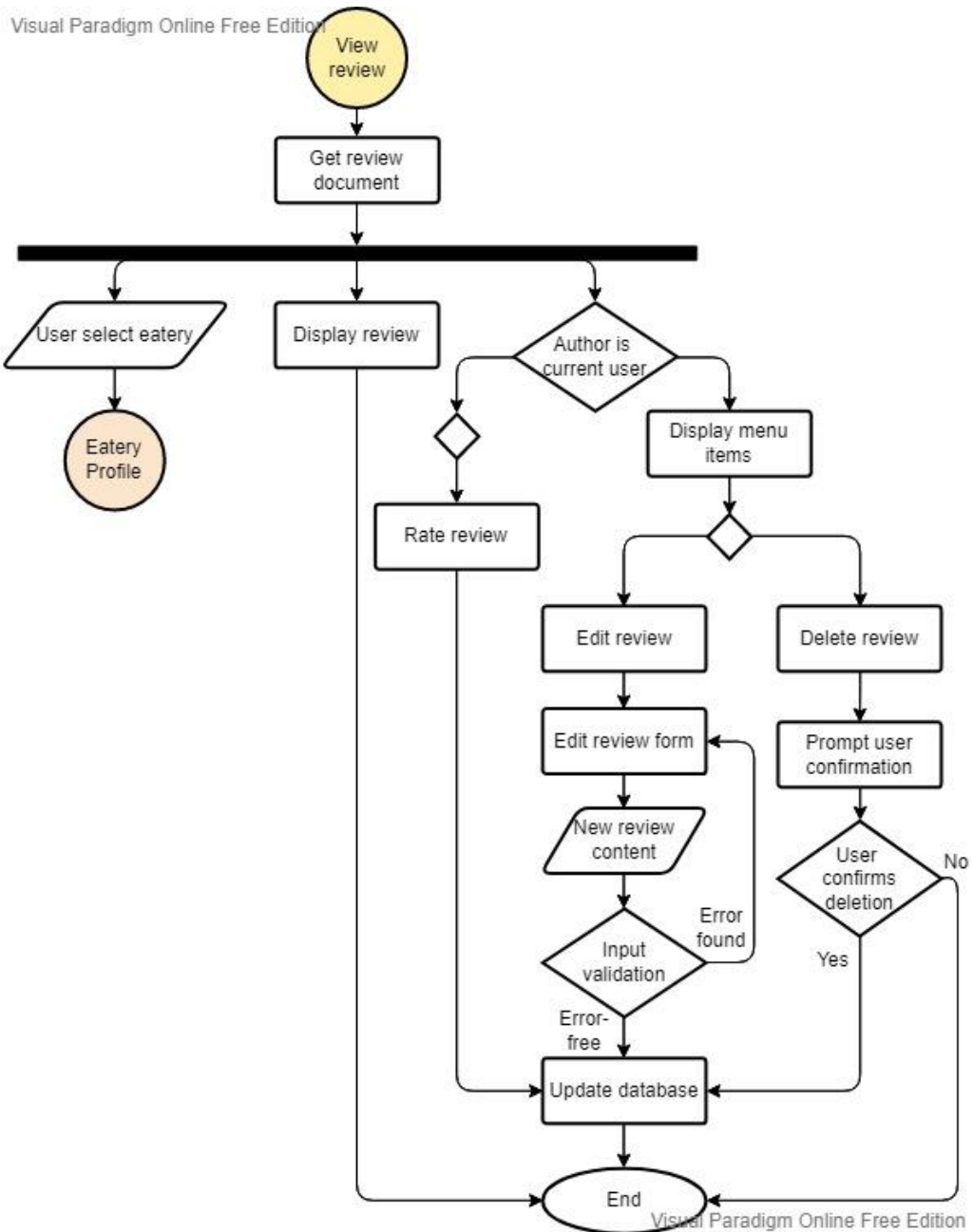


Figure 4.1.4 View Review Flow

When users tap on a review to view it, they are redirected to the view review screen where the system will first fetch the review document from the database whereby the data is processed and displayed in the UI. Users can opt to tap the eatery of the review to view the profile. At the same time, the author ID of the review and the current user ID will be compared and checked to determine if the author is the current user. If so, the user will be able to access the menu items to edit or delete the review. Otherwise, users will view be viewing the review as another party where they can rate the reviews as either “Helpful” or “Funny”.

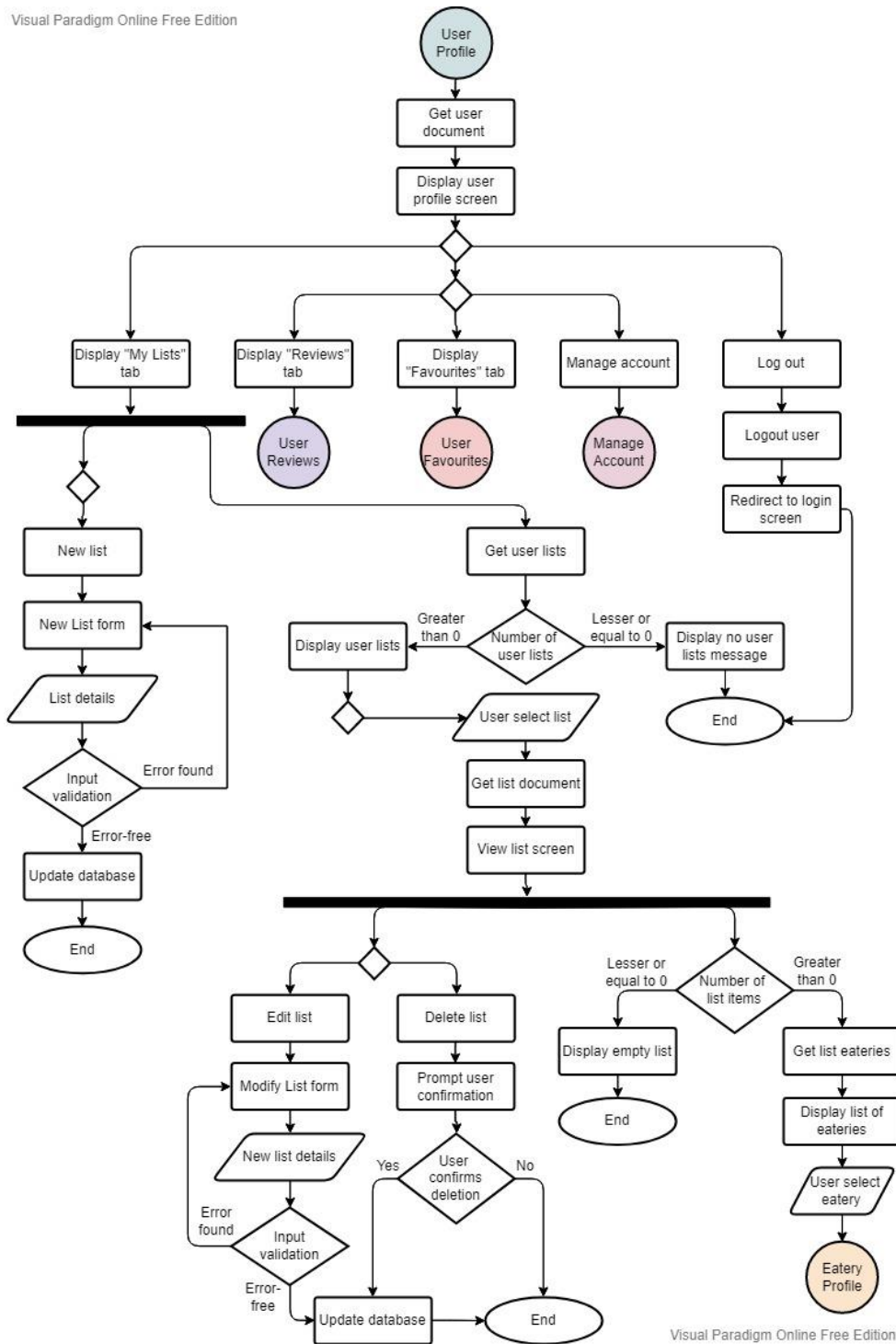


Figure 4.1.5 User Profile Flow

In the user profile, the system first gets the user document and then displays the data accordingly in the UI. Users can view their lists, reviews, favourites and manage their account as well as logout from the application from this activity. For the “My Lists” tab, the users’ lists will be retrieved and displayed accordingly if any. Users can create a new list whereby they enter the valid list details input and then the database is updated.

When user selects a list, they are redirected to the view list screen where the relevant list document is retrieved from the database and displayed in the UI. In the view list screen, users can select list items to view the eatery profiles, edit the list or delete the list.

For user reviews, the system fetches the users' reviews documents and then displays the reviews accordingly. When users select a review, they are redirected to the view review screen. If users have not made any reviews, the system displays a no review message. For user favourites, the system fetched the user favourites and displays the list accordingly. Uses can select an eatery to view the eatery profile as well as unfavourite an item from the list.

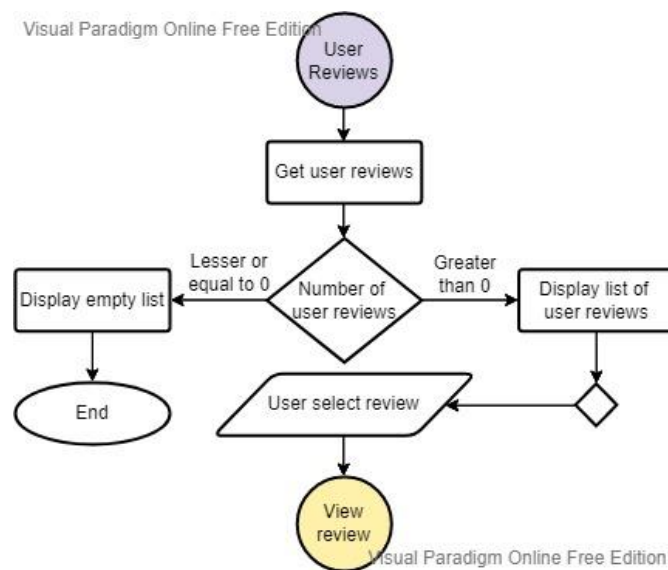


Figure 4.1.6 View User Review Flow

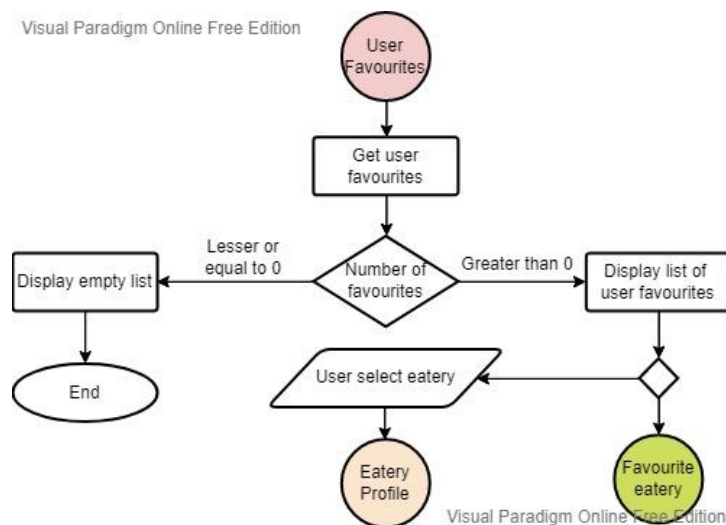


Figure 4.1.7 View User Favourites Flow

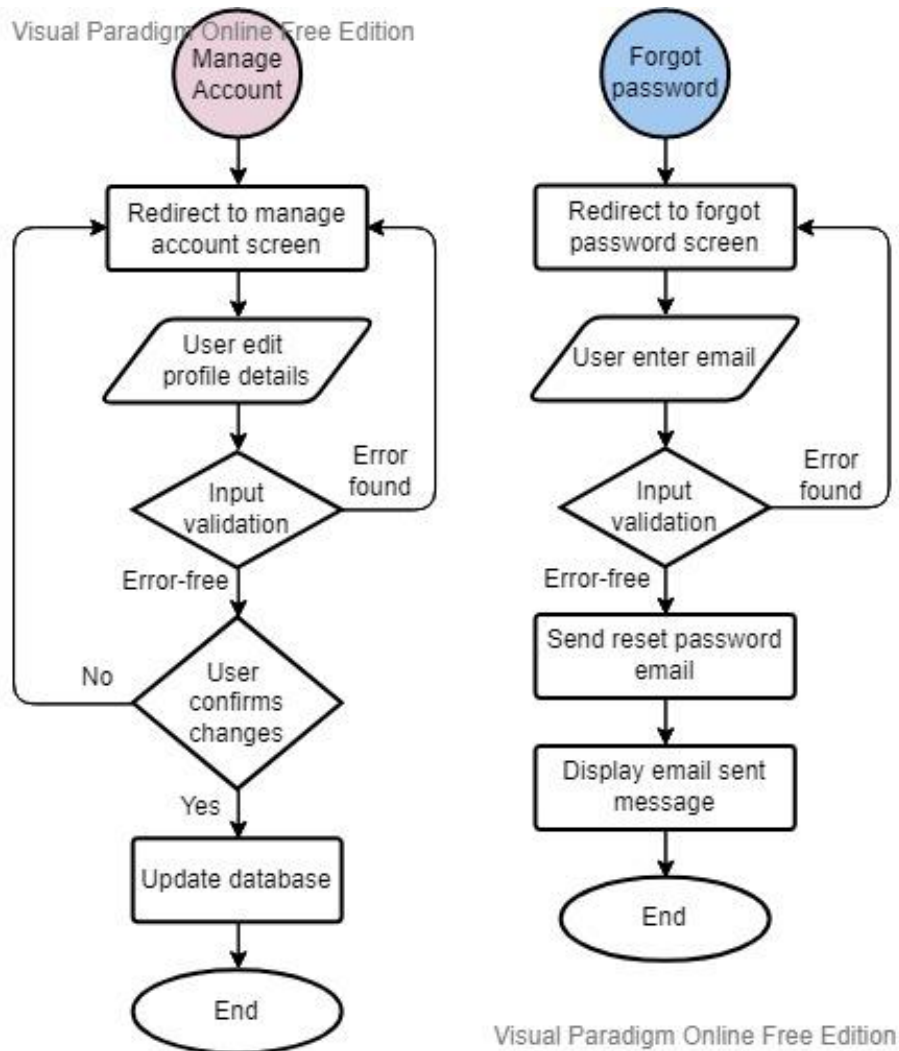


Figure 4.1.8 Manage Account and Forgot Password Flow

In the manage account activity, users can edit their profile details. The input is validated as they input any changes. Then, when users confirm the changes, the database is updated accordingly. For forgot password, users are redirected to the forgot password screen where they are prompt to input their email. The email is validated, and a reset password email is sent to the user’s email. The system displays a message to inform the users about the email.

Once they create a review, a “review” document will be created in the database, and if users included photos in their reviews, the photos will be uploaded to Cloud Storage. Subsequently, the “user” document of the author and the “eatery” document in which the review is written about will be updated in the Firestore database as well.

In the user profile, users can view their reviews, favourites, and lists which are all fetched from the user document in Firestore. When viewing reviews, the data will be fetched from the “review” documents from Firestore. User favourites are updated in the “user” document. When users create lists, “list” documents are created, and the “user” document is updated in the Firestore database. Each list can also be edited or deleted. Additionally, users can change their username and profile picture where the profile picture will be uploaded to Cloud Storage and the user document in Firestore will be updated accordingly.

4.3 Storyboard

To visualize the required elements in the layout and navigation of the application, a storyboard of the application was designed as seen in the figures below. The storyboard outlines the overall structure of the app as well as the user journey across the multiple screens and functions. It is an important tool to help effectively design the final product as well as to visualize a story.

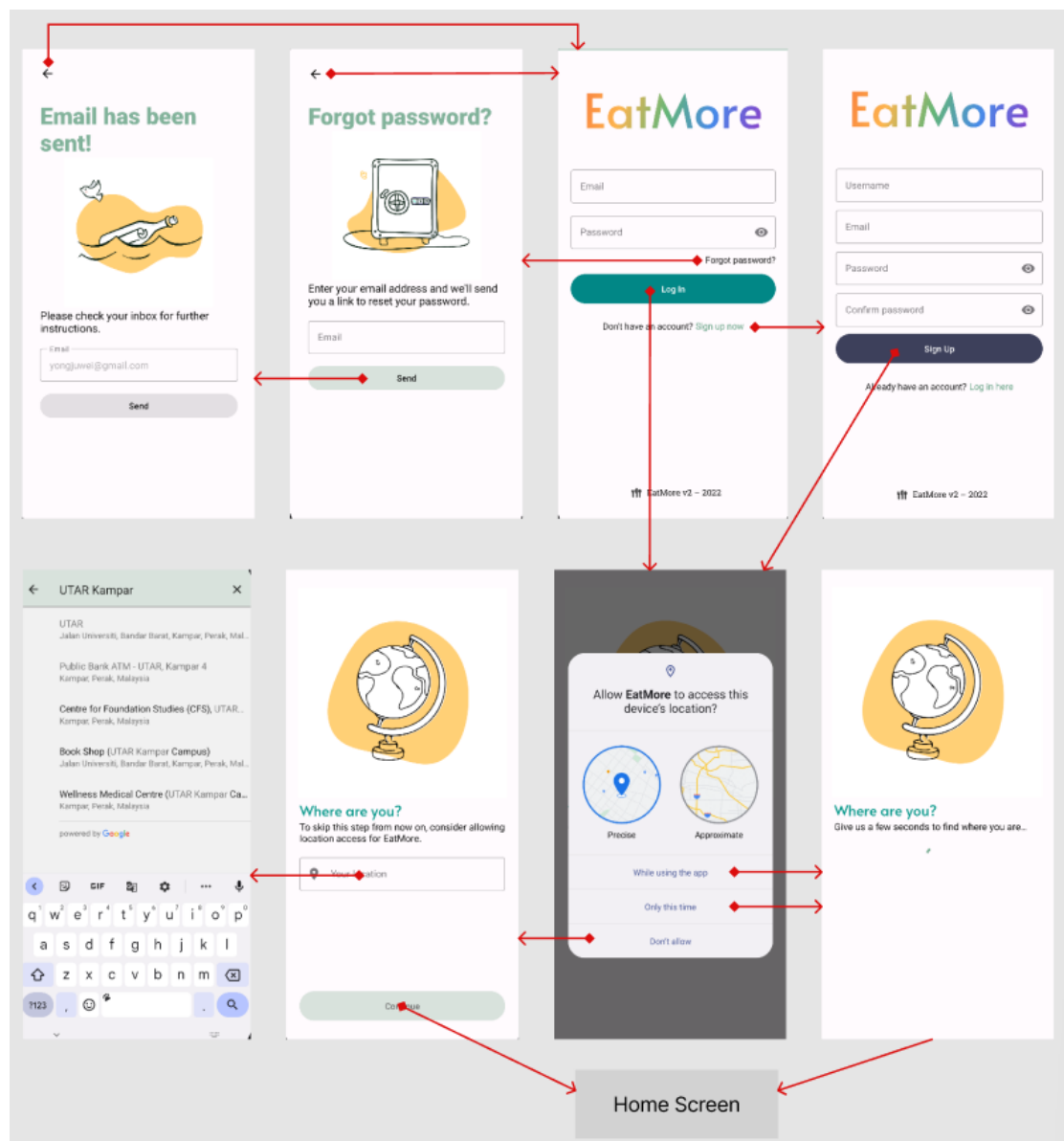


Figure 4.3.1 Storyboard

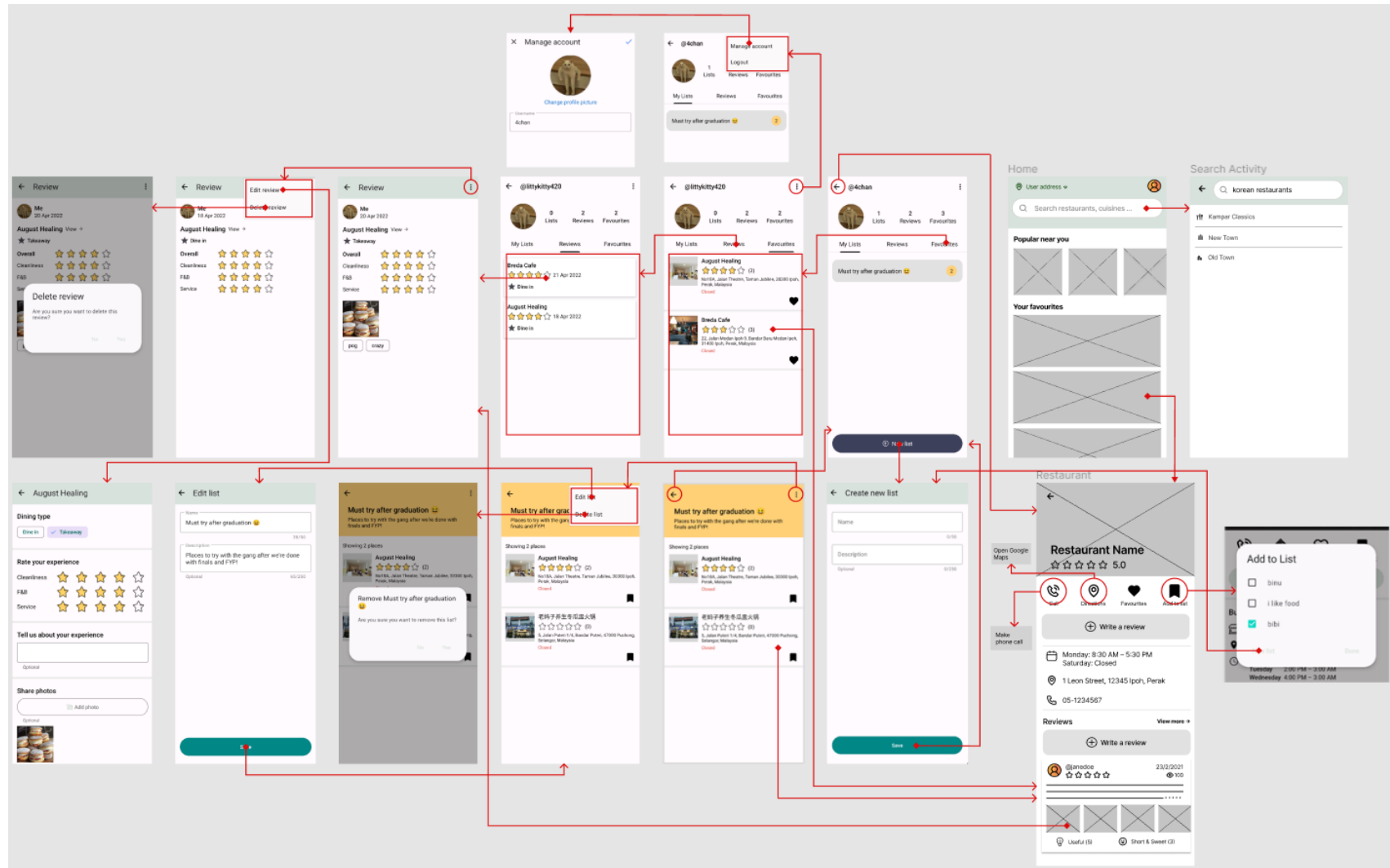


Figure 4.3.2 Storyboard

4.4 Use Case

4.4.1 Use Case Diagram

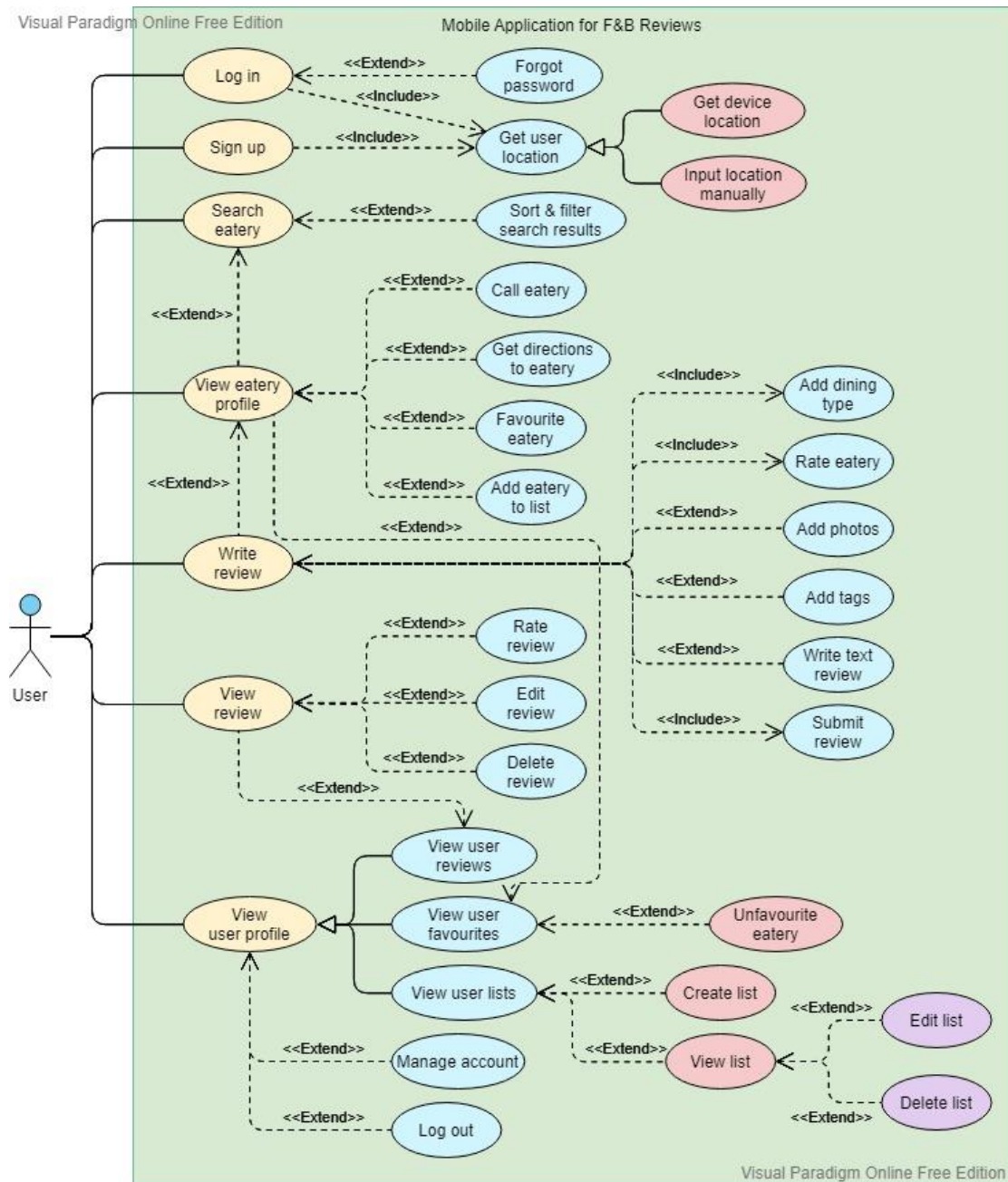


Figure 4.4.1.1 Use Case Diagram

4.4.2 Use Case Descriptions

i) Log in

Use Case ID	UC001	
Name	Log in	
Description	User wants to log in to the application	
Actor	User	
Trigger	User launches the application or logs out	
Precondition	<ol style="list-style-type: none"> 1. User is not logged in 2. User has a registered account 	
Basic Flow	Step	Action
	1	User launches the application
	2	User is redirected to log in screen
	3	User enters credentials
	4	System authenticates User
	5	User is redirected to home screen
Alternative Flow	A1 – User intends to sign up	
	3.1	User selects “Sign up now” text button
	3.2	User is redirected to sign up screen
	A2 – User is already logged in	
	2.1	User is redirected to get user location screen
	A3 – User forgot password	
	3.1	User selects “Forgot password” text button
	3.2	User is redirected to forgot password screen
Exception Flow	E1 – Authentication failed	
	5.1	System displays error message and prompts user for correct credentials.
Post condition	<ol style="list-style-type: none"> 1. User is logged in to the application 2. User is directed to get user location screen 	
Relationships	Include:	UC005 – Get user location
	Extend:	UC004 – Forgot password

Table 4.4.2.1 Log In Use Case Description

ii) Log out

Use Case ID	UC002	
Name	Log out	
Description	To allow users to log out of the application	
Actor	User	
Trigger	User wants to log out of the system	
Precondition	<ol style="list-style-type: none"> 1. User is logged in 2. User wants to log out 3. User is on view user profile screen 	
Basic Flow	Step	Actions
	1	User selects menu item “Log out”
	2	System logs out user
	3	User is redirected to log in screen
Post condition	<ol style="list-style-type: none"> 1. User is logged out 2. User is redirected to the log in screen 	

Table 4.4.2.2 Log Out Use Case Description

iii) Sign up

Use Case ID	UC003	
Name	Sign up	
Description	To allow users to register an account	
Actor	User	
Trigger	User selects “Sign up now” text button in log in screen	
Precondition	<ol style="list-style-type: none"> 1. User does not have an existing account in the system 2. User is not logged in 	
Basic Flow	Step	Action
	1	System prompts user to log in or sign up
	2	User selects registration option
	3	User is redirected to sign up screen
	4	User enters information
	5	System verifies username availability and information entered by User
	6	System authenticates User and adds new user document to database
	7	User is redirected to get user location screen
Alternative Flow	A1 – Cancel registration	
	4.1	User selects “Log in now”
	4.2	User is redirected to log in screen and any information entered by user is cleared
Exception Flow	E1 – Invalid information entered	

	5.1	System displays error message and prompts user for valid input
Post condition	<ol style="list-style-type: none"> 1. User is registered in the system 2. User document is created in the database 3. User is logged in to the application 	
Relationships	Include:	UC005 – Get user location

Table 4.4.2.3 Sign Up Use Case Description

iv) Forgot password

Use Case ID	UC004	
Name	Forgot password	
Description	User has forgotten password and wants to reset password	
Actor	User	
Trigger	User selects “Forgot Password” text button in log in screen	
Precondition	User has a registered account	
Basic Flow	Step	Action
	1	System prompts for user email
	2	User enters email
	3	System authenticates user email
	4	System sends reset password email to user
Exception Flow	E1 – Email is not registered	
	4.1	System displays error message
Post condition	User receives email to reset password	

Table 4.4.2.4 Forgot Password Use Case Description

v) Get user location

Use Case ID	UC005	
Name	Get user location	
Description	To get coordinates of user’s location	
Actor	User	
Trigger	User successfully logs in to the application	
Precondition	<ol style="list-style-type: none"> 1. User is logged in 	
Basic Flow	Step	Action
	1	System prompts user permission for access to user location
	2	User enters decision
	3	System gets user location
	4	User is redirected to home screen
Alternative Flow	A1 – User allows permission	
	3.1	System gets user device location
	A2 – User denies permission	

	3.1	System prompts user to input location manually
Post condition	<ol style="list-style-type: none"> 1. System has user coordinates 2. User is redirected to home screen 	
Relationships	Generalization:	UC006 – Get device location UC007 – Input location manually

Table 4.4.2.5 Get User Location Use Case Description

vi) Get device location

Use Case ID	UC006	
Name	Get device location	
Description	Get coordinates and address of user's device	
Actor	Actor	
Trigger	User logs in to the application	
Precondition	<ol style="list-style-type: none"> 1. User device has GPS service 2. User allowed application permission to access location 	
Basic Flow	Step	Action
	1	System accesses GPS services
	2	System retrieves device coordinates
	3	System gets location address based on coordinates
Alternative Flow	A1 – GPS services not enabled	
	2.1	System enables GPS services
Post condition	<ol style="list-style-type: none"> 1. System has user device location coordinates 2. User is redirected to home screen 	

Table 4.4.2.6 Get Device Location Use Case Description

vii) Input location manually

Use Case ID	UC007	
Name	Input location manually	
Description	User wants to input location manually	
Actor	User	
Trigger	<ol style="list-style-type: none"> 1. User denied application permission to location 2. User wants to change their location coordinates for session 	
Precondition	User is logged in	
Basic Flow	Step	Action
	1	User selects location input button
	2	System initializes Place Autocomplete
	3	User enters and selects location
	4	System gets location coordinates and address
	5	User is redirected to home screen

Post condition	<ol style="list-style-type: none"> 1. System has user location coordinates and address 2. User is redirected to home screen
-----------------------	---

Table 4.4.2.7 Input Location Manually Use Case Description

viii) Search eatery

Use Case ID	UC_008	
Name	Sign up	
Description	To allow users to search for eateries	
Actor	User	
Trigger	User selects search bar in home screen	
Precondition	System has user location coordinates	
Basic Flow	Step	Action
	1	System is displaying the home screen
	2	User selects search bar
	3	User is redirected to search screen
	4	User enters search keyword
	5	System displays results
Alternative Flow	A1 – Cancel search	
	4.1	User selects navigation icon
	4.2	User is redirected to home screen
Exception Flow	E1 – No results found	
	5.1	System displays “No results found” message
Post condition	System displays results	
Relationships	Extend:	UC009 – Sort & filter search results

Table 4.4.2.8 Search Eatery Use Case Description

ix) Sort & filter search results

Use Case ID	UC009	
Name	Sort & filter search results	
Description	To allow users to sort and filter search results	
Actor	User	
Trigger	User wants to filter results	
Precondition	Search results are not empty	
Basic Flow	Step	Action
	1	User selects filter option
	2	System displays filtered search results
Post condition	User can view filtered list of search results	

Table 4.4.2.9 Sort & Filter Search Results Use Case Description

x) View eatery profile

Use Case ID	UC010	
Name	View eatery profile	
Description	To allow users to view eatery profile	
Actor	User	
Trigger	User wants to view eatery profile	
Precondition	System has eatery place ID	
Basic Flow	Step	Action
	1	User selects eatery
	2	User is redirected to eatery profile screen
Exception Flow	E1 – Eatery not found	
	2.1	System displays a toast with error message
	2.2	User is redirected to previous screen
Post condition	User can view eatery profile screen	
Relationships	Extend:	UC011 – Call eatery UC012 – Get directions to eatery UC013 – Favourite eatery UC014 – Add eatery to list UC015 – Write review

Table 4.4.2.10 View Eatery Use Case Description

xi) Call eatery

Use Case ID	UC011	
Name	Call eatery	
Description	User wants to contact eatery	
Actor	User	
Trigger	User taps call button or eatery contact number	
Precondition	Eatery contact number is available	
Basic Flow	Step	Action
	1	System prompts user for permission to call
	2	User allows permission
	3	User is redirected to device phone call screen
	4	System makes call to eatery contact number
Alternative Flow	A1 – User denies permission	
	2.1	System displays permission denied message
Post condition	User is redirected to device phone call screen	

Table 4.4.2.11 Call Eatery Use Case Description

xii) Get directions to eatery

Use Case ID	UC012	
Name	Get directions to eatery	
Description	To allow users to get directions to eatery	
Actor	User	
Trigger	User taps get directions button or eatery address	
Precondition	Eatery address is available	
Basic Flow	Step	Action
	1	User is redirected to Google Maps
Post condition	User is redirected to Google Maps	

Table 4.4.2.12 Get Directions to Eatery Use Case Description

xiii) Favourite eatery

Use Case ID	UC013	
Name	Favourite eatery	
Description	To allow users to favourite the eatery	
Actor	User	
Trigger	User wants to add eatery to favourites	
Precondition	Eatery is not a user favourite	
Basic Flow	Step	Action
	1	User taps on favourite button
	2	System adds eatery to list of favourites in user document
Post condition	Eatery is a user favourite	

Table 4.4.2.13 Favourite Eatery Use Case Description

xiv) Add eatery to list

Use Case ID	UC014	
Name	Add eatery to list	
Description	To allow users to add eatery to one or more of the user's lists	
Actor	User	
Trigger	User wants to add eatery to a user list	
Precondition	User is viewing eatery profile	
Basic Flow	Step	Action
	1	User taps on add to list button
	2	System displays add to list dialog
	3	User selects list
	4	System adds eatery to list document in database
	5	User dismisses dialog
Alternative Flow	A1 – Add new list	

	3.1	User selects “New list”
	3.2	User is redirected to create list screen
	3.3	User enters list details
	3.4	System updates database
	3.5	User is redirected back to eatery profile screen
Post condition	Eatery is added in user list	

Table 4.4.2.14 Add Eatery to List Use Case Description

xv) Write review

Use Case ID	UC015	
Name	Write review	
Description	To allow users to create reviews for selected eatery	
Actor	User	
Trigger	User selects “Write a review” button	
Precondition	<ol style="list-style-type: none"> 1. User is viewing an eatery profile screen 2. User wants to write review for eatery 	
Basic Flow	Step	Action
	1	User selects “Write a review” button
	2	User is redirected to review form screen
	3	User enters information
	4	System validates information
	5	System updates user document, eatery document and adds new review document to review collection in database
	6	User is redirected to eatery profile screen
Exception Flow	E1 – Invalid form input	
	5.1	System displays error message and prompts user for correct input
Post condition	<ol style="list-style-type: none"> 1. User review is recorded in review collection, eatery document and user document in database 2. Eatery profile screen is updated with new review 	
Relationships	Include:	UC030 – Add dining type UC031 – Rate eatery UC035 – Submit review
	Extend:	UC032 – Add photos UC033 – Add tags UC034 – Write text review

Table 4.4.2.15 Write Review Use Case Description

xvi) View review

Use Case ID	UC_007	
Name	View review	
Description	To allow users to view a review in full detail	
Actor	User	
Trigger	User wants to view a review	
Precondition	Review exists	
Basic Flow	Step	Action
	1	System displays summarized version of review
	2	User selects review
	3	User is redirected to view review screen
Post condition	User is on view review screen	
Relationships	Extend:	UC010 – View eatery profile UC017 – Rate review UC018 – Edit review UC018 – Delete review

Table 4.4.2.16 View Review Use Case Description

xvii) Rate review

Use Case ID	UC017	
Name	Rate review	
Description	To allow users to assess whether a review was helpful or funny	
Actor	User	
Trigger	User wants to rate a review	
Precondition	User is not the author of the review	
Basic Flow	Step	Action
	1	User selects either “Helpful” or “Funny” chip button
	2	System adds user ID to helpful or funny array in review document
Post condition	<ol style="list-style-type: none"> 1. Helpful or funny count is updated on chip button 2. Review document is updated 	

Table 4.4.2.17 Rate Review Use Case Description

xviii) Edit review

Use Case ID	UC018
Name	Edit review
Description	To allow user to modify a review made
Actor	User
Trigger	User wants to change their opinion or modify contents of a review
Precondition	User is the author of the review

Basic Flow	Step	Action
	1	User selects menu item “Edit review”
	2	User is redirected to edit review screen
	3	User enters review details
	4	System validates user input
	5	System updates review document in database
	6	User is redirected back to view review screen
Post condition	<ol style="list-style-type: none"> 1. Review document updated in database 2. User is redirected back to view review screen 	

Table 4.4.2.18 Edit Review Use Case Description

xix) Delete review

Use Case ID	UC019	
Name	Delete review	
Description	To allow users to remove their review	
Actor	User	
Trigger	User wants to delete the review	
Precondition	<ol style="list-style-type: none"> 1. User is the author of the review 2. User is viewing the review 	
Basic Flow	Step	Action
	1	User selects menu item “Delete review”
	2	System prompts user to confirm
	3	User confirms delete review
	4	System deletes review document and updates user document and relevant eatery document
	5	User is redirected to previous screen
Alternative Flow	A1 – Cancel delete	
	3.1	User cancels delete
Post condition	<ol style="list-style-type: none"> 1. Review document is deleted from database 2. User and eatery document is updated in database 3. User is redirected back to previous screen 	

Table 4.4.2.19 Delete Review Use Case Description

xx) View user profile

Use Case ID	UC020
Name	View user profile
Description	To allow users to view their user profile
Actor	User
Trigger	User selects profile icon in home screen

Precondition	User is logged in	
Basic Flow	Step	Action
	1	User selects profile icon in home screen
	2	User is redirected to user profile screen
Post condition	User can access user profile functions	
Relationships	Extend:	UC002 – Log out UC029 – Manage account
	Generalization:	UC021 – View user reviews UC022 – View user favourites UC024 – View user lists

Table 4.4.2.20 View User Profile Use Case Description

xxi) View user reviews

Use Case ID	UC021	
Name	View user reviews	
Description	User can view their reviews if any	
Actor	User	
Trigger	User opens user profile and taps on “Reviews” tab	
Precondition	User is logged in	
Basic Flow	Step	Action
	1	System displays list of user reviews if any
Post condition	User can view their reviews if any	
Relationships	Extend:	UC016 – View review

Table 4.4.2.21 View User Reviews Use Case Description

xxii) View user favourites

Use Case ID	UC022	
Name	View user favourites	
Description	User can view their favourites if any	
Actor	User	
Trigger	User opens user profile and taps on “Favourites” tab	
Precondition	User is logged in	
Basic Flow	Step	Action
	1	System displays list of user favourites if any
Post condition	User can view their reviews if any	
Relationships	Extend:	UC010 – View eatery profile

Table 4.4.2.22 View User Favouritess Use Case Description

xxiii) Unfavourite eatery

Use Case ID	UC023	
Name	Unfavourite eatery	
Description	To allow users to unfavourite the eatery	
Actor	User	
Trigger	User wants to unfavourite the eatery	
Precondition	Eatery is a user favourite	
Basic Flow	Step	Action
	1	User taps on favourite button
	2	System removed eatery from user favourites in database
Post condition	User document is updated in database	

Table 4.4.2.23 Unfavourite Use Case Description

xxiv) View user lists

Use Case ID	UC024	
Name	View user lists	
Description	User can view their lists if any	
Actor	User	
Trigger	User opens user profile and taps on “My Lists” tab	
Precondition	User is logged in	
Basic Flow	Step	Action
	1	System displays list of user lists if any
Post condition	User can view their lists if any	
Relationships	Extend:	UC025 – Create list UC026 – View list

Table 4.4.2.24 View User Lists Use Case Description

xxv) Create list

Use Case ID	UC025	
Name	Create list	
Description	To allow users to create new lists	
Actor	User	
Trigger	User selects “New list” button	
Precondition	User is on view user profile screen	
Basic Flow	Step	Action
	1	User is redirected to add list screen
	2	User enters list details
	3	System validates input

	4	System creates new list document and updates user document in database
	5	User is redirected to previous screen
Post condition	List document is created, and user document is updated in database	

Table 4.4.2.25 Create List Use Case Description

xxvi) View list

Use Case ID	UC026	
Name	View list	
Description	To allow users to view user's list(s) of eateries	
Actor	User	
Trigger	User selects list tab in user profile	
Precondition	<ol style="list-style-type: none"> 1. User is logged in 2. User is on user profile screen 	
Basic Flow	Step	Action
	1	User selects list tab in user profile screen
	2	System displays User's lists
	3	User selects a list
	4	User is redirected to selected list screen
Alternative Flow	A1 – User has no lists	
	2.1	System displays “No lists” message
Post condition	User can view all lists (if any) and access list functions	
Relationships	Extend:	UC027 – Edit list UC028 – Delete list

Table 4.4.2.26 View List Use Case Description

xxvii) Edit list

Use Case ID	UC027	
Name	Edit list	
Description	To allow users to edit their list name or description or both	
Actor	User	
Trigger	User selects menu “Edit list” in view list screen	
Precondition	User is on view list screen	
Basic Flow	Step	Action
	1	User is redirected to edit list screen
	2	User enters list details
	3	System validates input
	4	System creates updates list document in database
	5	User is redirected to previous screen

Post condition	List document is updated in database
-----------------------	--------------------------------------

Table 4.4.2.27 Edit List Use Case Description

xxviii) Delete list

Use Case ID	UC028	
Name	Delete list	
Description	To allow users to delete user list	
Actor	User	
Trigger	User selects menu item “Delete list” in view list screen	
Precondition	User is on view list screen	
Basic Flow	Step	Action
	1	System prompts user confirmation
	2	User confirms delete list
	3	System deletes list document and updates user document
	4	User is redirected to previous screen
Alternative Flow	A1 – Cancel delete	
	2.1	User cancels delete
Post condition	List document is deleted, and user document is updated in database	

Table 4.4.2.28 Delete List Use Case Description

xxix) Manage account

Use Case ID	UC029	
Name	Manage account	
Description	To allow users to change their username or profile picture	
Actor	User	
Trigger	User selects menu item “Manage account” in user profile screen	
Precondition	User is on user profile screen	
Basic Flow	Step	Action
	1	User is redirected to manage account screen
	2	User enters profile details
	3	System validates input
	4	System updates user document in database
Post condition	User document is updated in database	

Table 4.4.2.29 Manage Account Use Case Description

xxx) Add dining type

Use Case ID	UC030	
Name	Add dining type	
Description	Users need to select the dining type for their review between “Dine in” or “Takeaway”	
Actor	User	
Trigger	User is writing review	
Precondition	User is on add or edit review screen	
Basic Flow	Step	Action
	1	System prompts user input
	2	User selects a dining type

Table 4.4.2.30 Add Dining Type Use Case Description

xxxii) Rate eatery

Use Case ID	UC031	
Name	Rate eatery	
Description	Users need to rate the eatery on their cleanliness, F&B and service	
Actor	User	
Trigger	User is writing review	
Precondition	User is on add or edit review screen	
Basic Flow	Step	Action
	1	System prompts user input
	2	User rates eatery

Table 4.4.2.31 Rate Eatery Use Case Description

xxxiii) Add photos

Use Case ID	UC032	
Name	Add photos	
Description	Users can choose to add photos to their reviews	
Actor	User	
Trigger	User is writing review	
Precondition	User is on add or edit review screen	
Basic Flow	Step	Action
	1	User taps “Add photos” button
	2	User is redirected to user device’s gallery device
	3	User selects photo(s)
	4	System processes user photos for upload to Firebase Storage

Table 4.4.2.32 Add Photos Use Case Description

xxxiii) Add tags

Use Case ID	UC033	
Name	Add tags	
Description	User can opt to add content tags in their reviews	
Actor	User	
Trigger	User is writing review	
Precondition	User is on add or edit review screen	
Basic Flow	Step	Action
	1	System prompts user input
	2	User enters tags

Table 4.4.2.33 Add Tags Use Case Description

xxxiv) Write text review

Use Case ID	UC034	
Name	Write text review	
Description	User can opt to include text review in their reviews	
Actor	User	
Trigger	User is writing review	
Precondition	User is on add or edit review screen	
Basic Flow	Step	Action
	1	System prompts user input
	2	User enters text review

Table 4.4.2.34 Write Text Review Use Case Description

xxxv) Submit review

Use Case ID	UC035	
Name	Submit review	
Description	Create review based on user input	
Actor	User	
Trigger	User is writing review	
Precondition	User is on add or edit review screen	
Basic Flow	Step	Action
	1	User selects "Submit review" button
	2	System validates input
	3	System creates new review document and updates eatery and user document in database
Exception Flow	E1 – Invalid form input	

	5.1	System displays error message and prompts user for correct input
Post condition		Review document is created, and user and eatery document are updated in database

Table 4.4.2.35 Submit Review Use Case Description

4.5 Activity Diagram

i) Log in

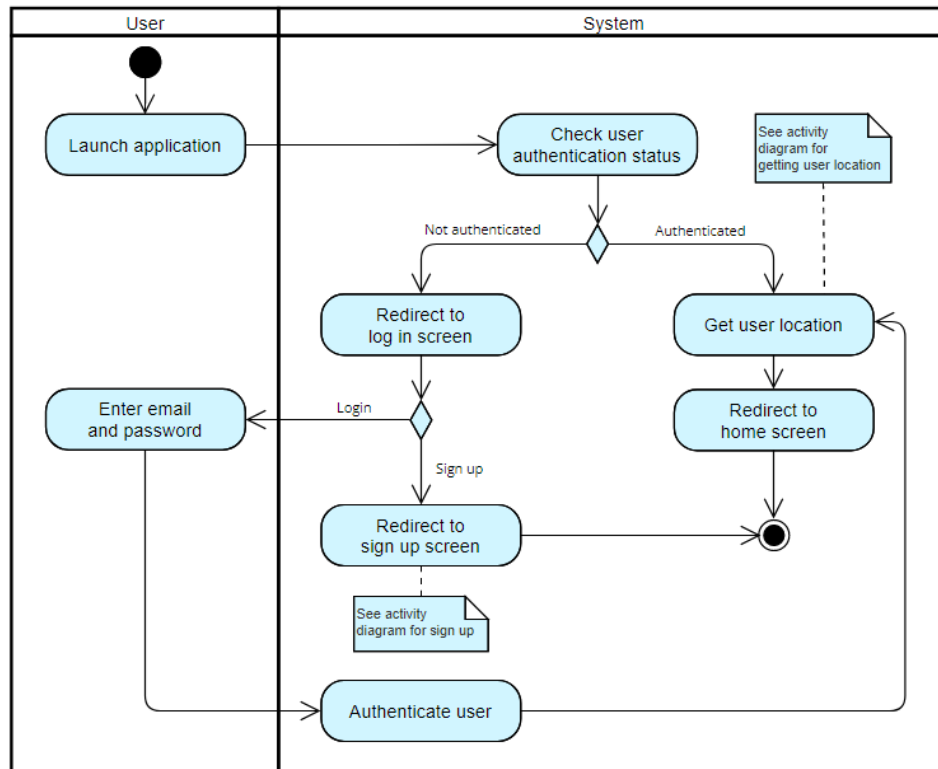


Figure 4.5.1 Log In Activity Diagram

ii) Log out

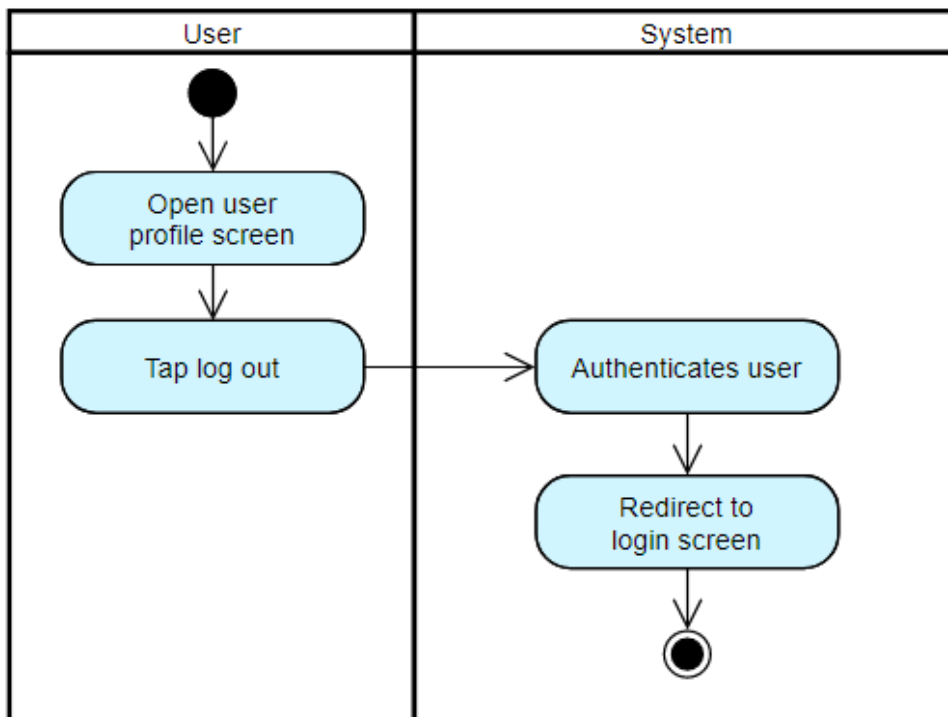


Figure 4.5.2 Log Out Activity Diagram

iii) Sign up

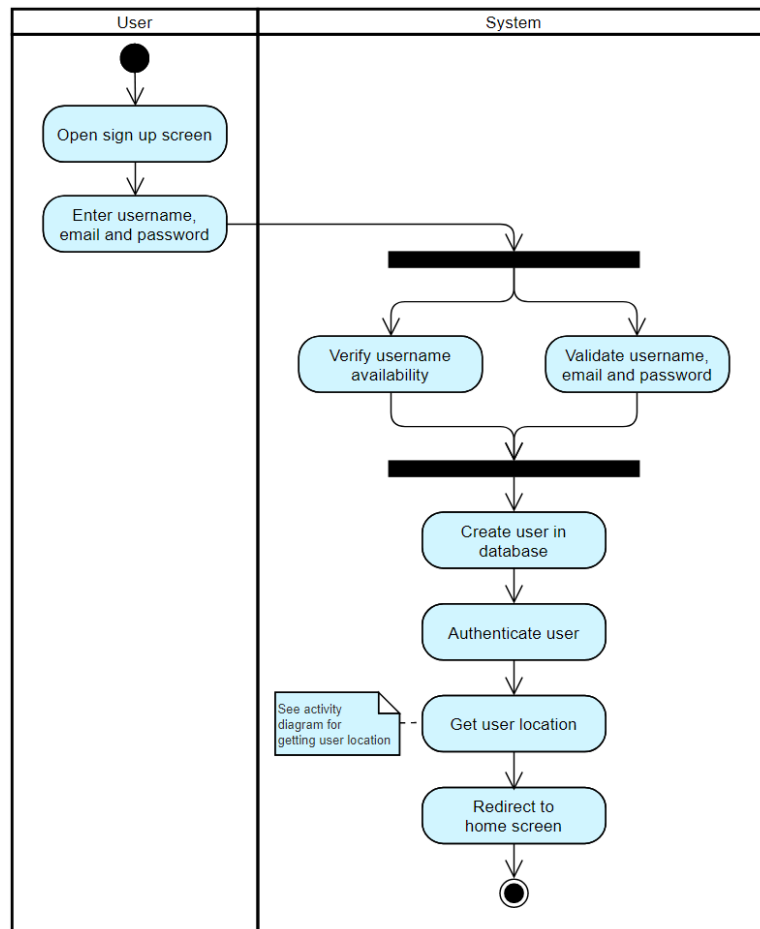


Figure 4.5.3 Sign Up Activity Diagram

iv) Forgot password

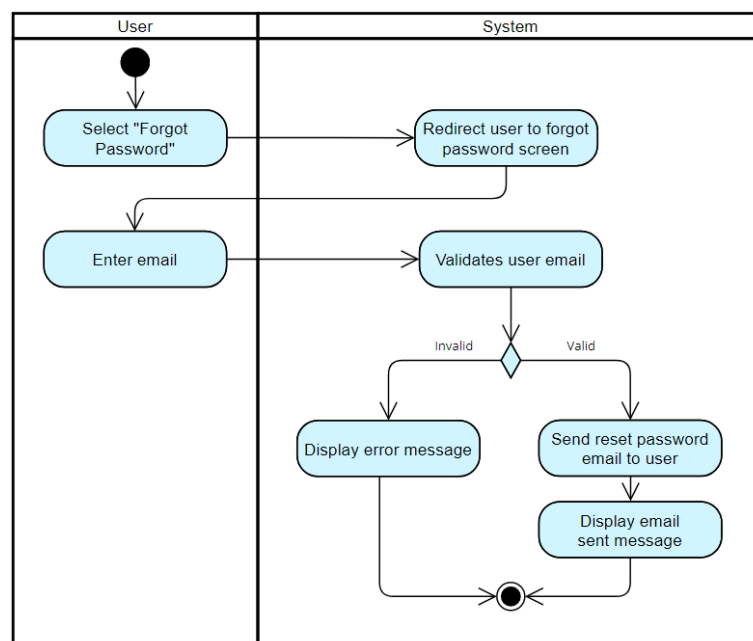


Figure 4.5.4 Forgot Password Activity Diagram

v) Get user location

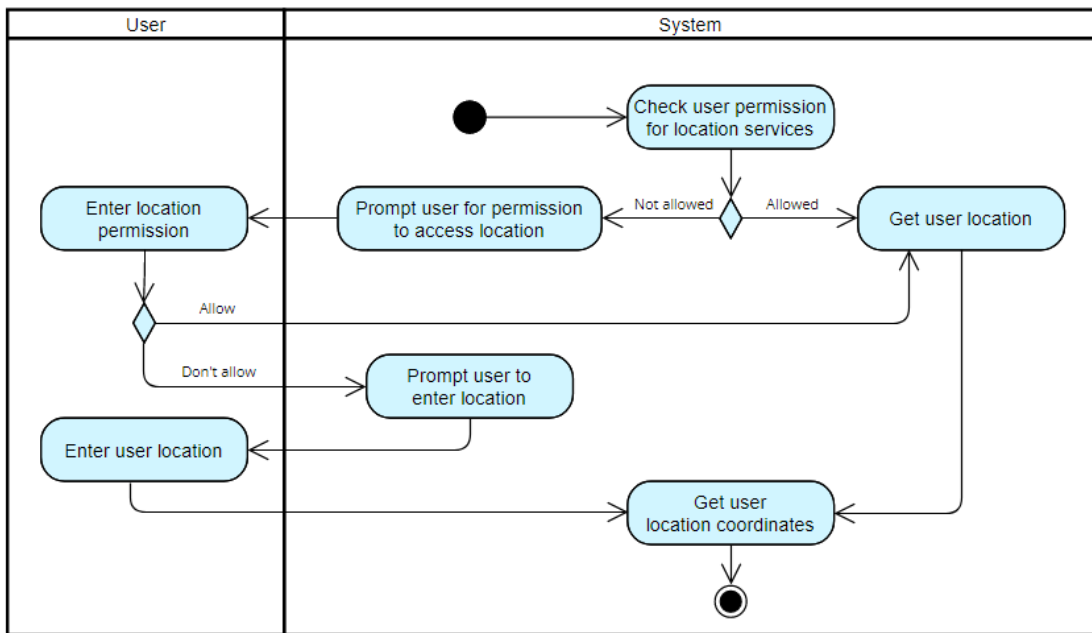


Figure 4.5.5 Get User Location Activity Diagram

vi) Search eatery

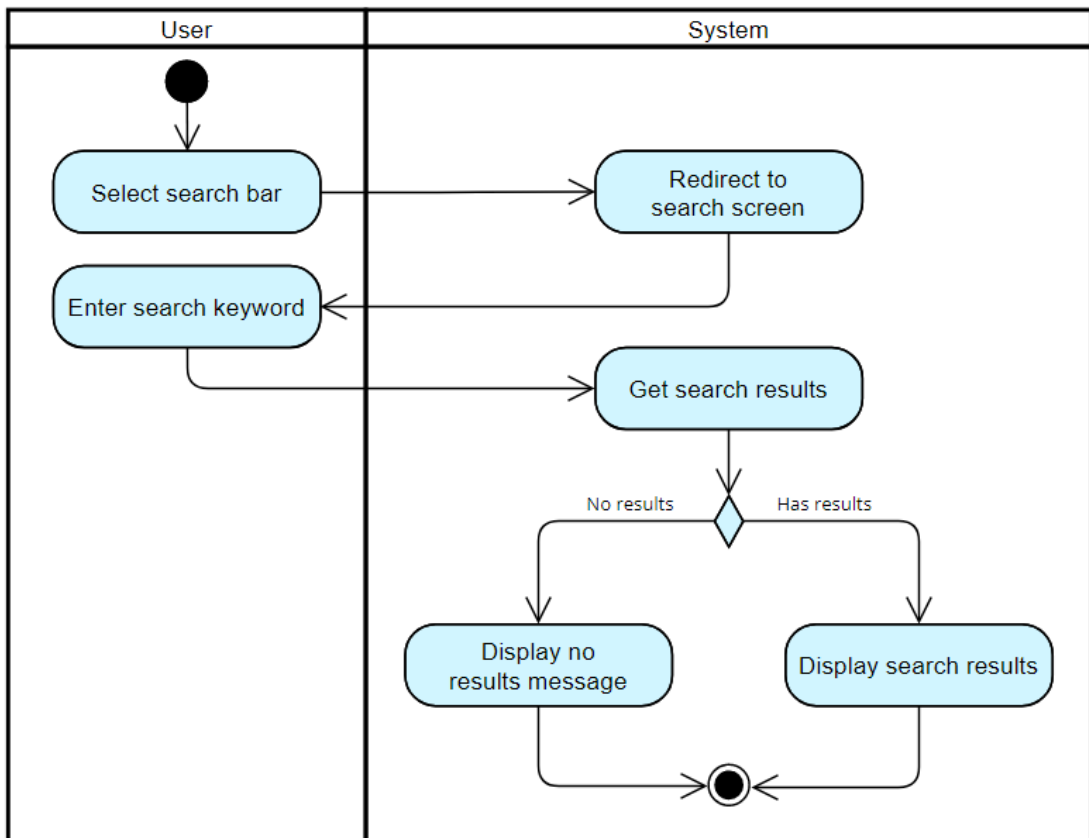


Figure 4.5.6 Search Eatery Activity Diagram

vii) Sort & filter search results

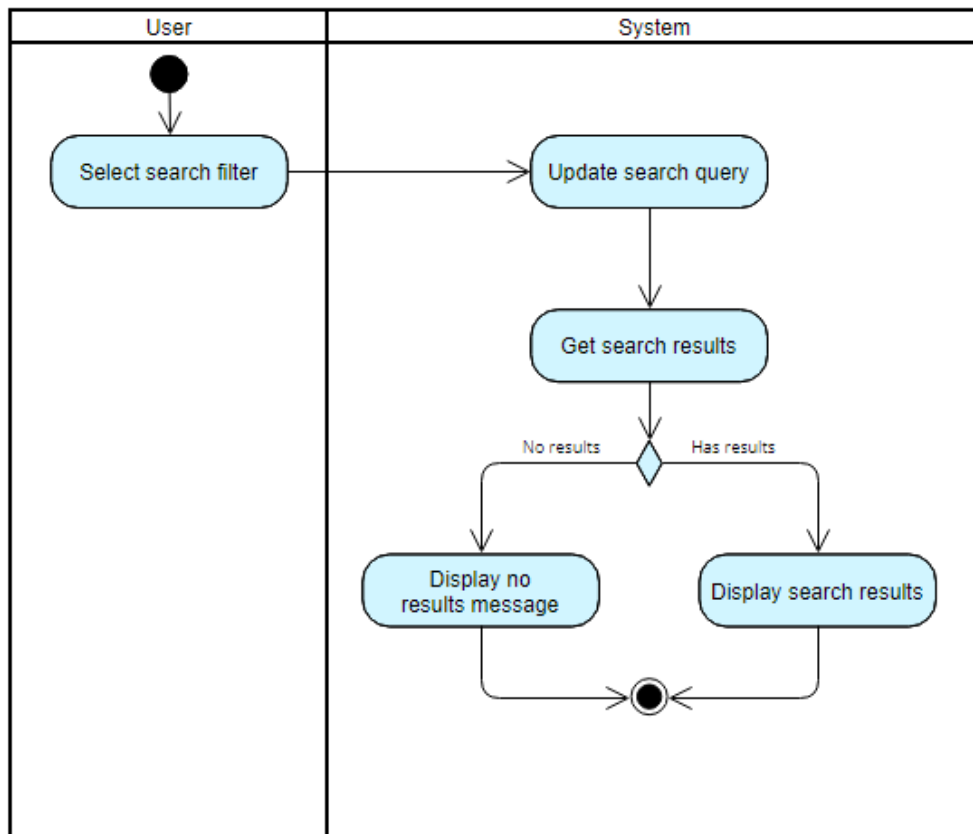


Figure 4.5.7 Sort & Filter Search Results Activity Diagram

viii) View eatery profile

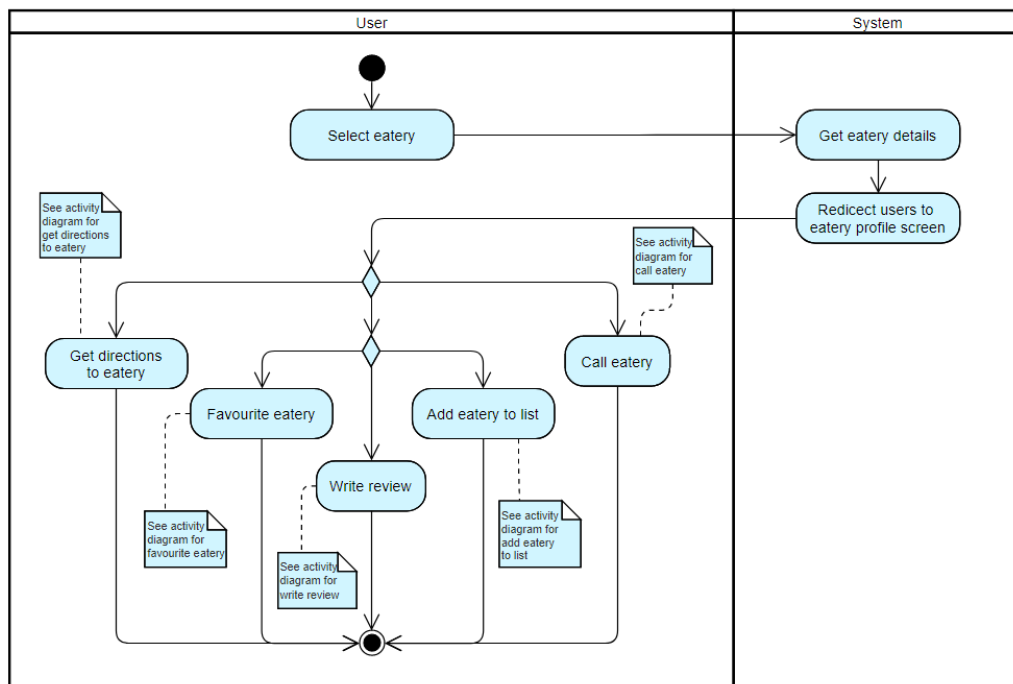


Figure 4.5.8 View Eatery Profile Activity Diagram

ix) Call eatery

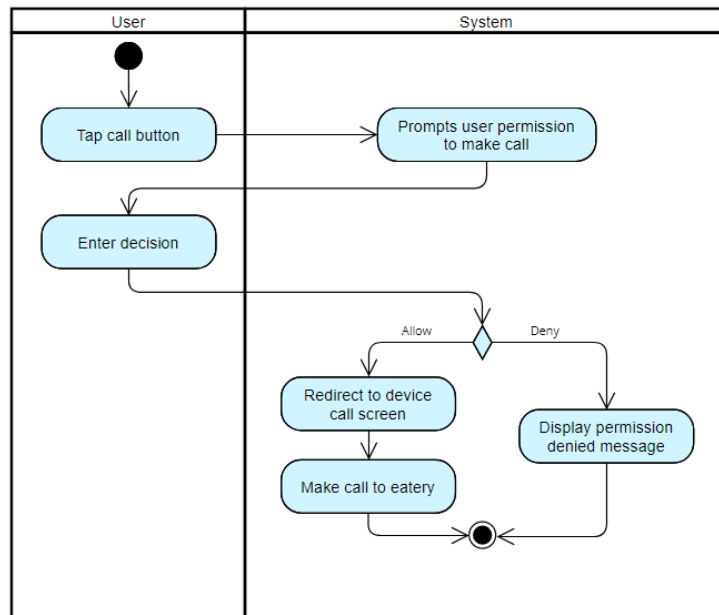


Figure 4.5.9 Call Eatery Activity Diagram

x) Get directions to eatery

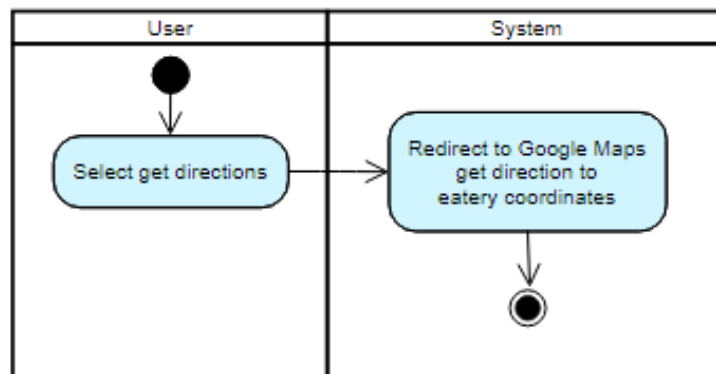


Figure 4.5.10 Get Directions To Eatery Activity Diagram

xi) Favourite eatery

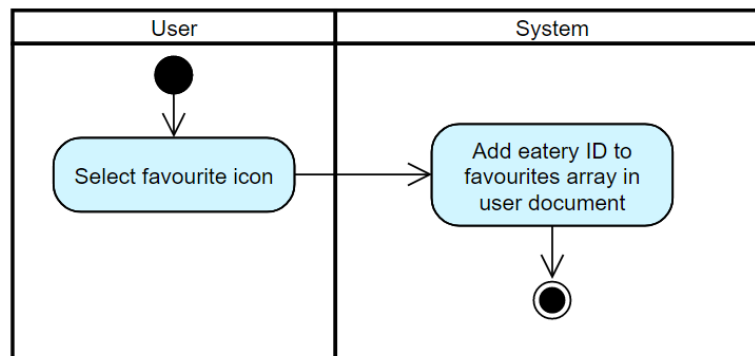


Figure 4.5.11 Favourite Eatery Activity Diagram

xii) Write review

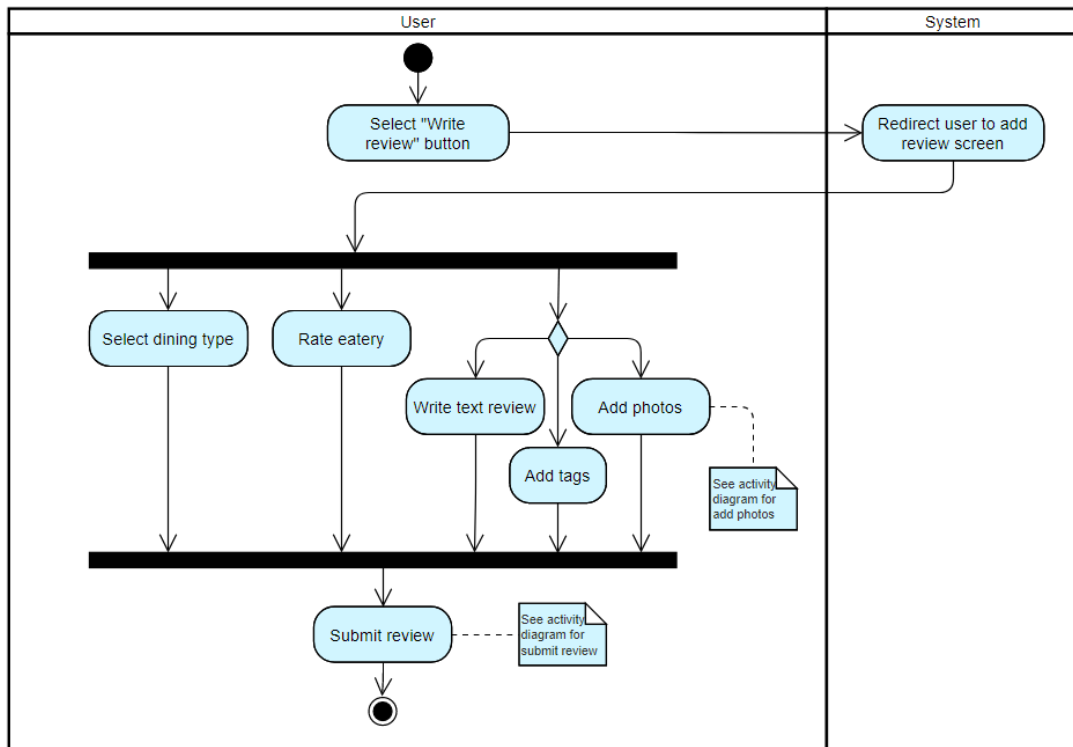


Figure 4.5.12 Write Review Activity Diagram

xiii) Add photo

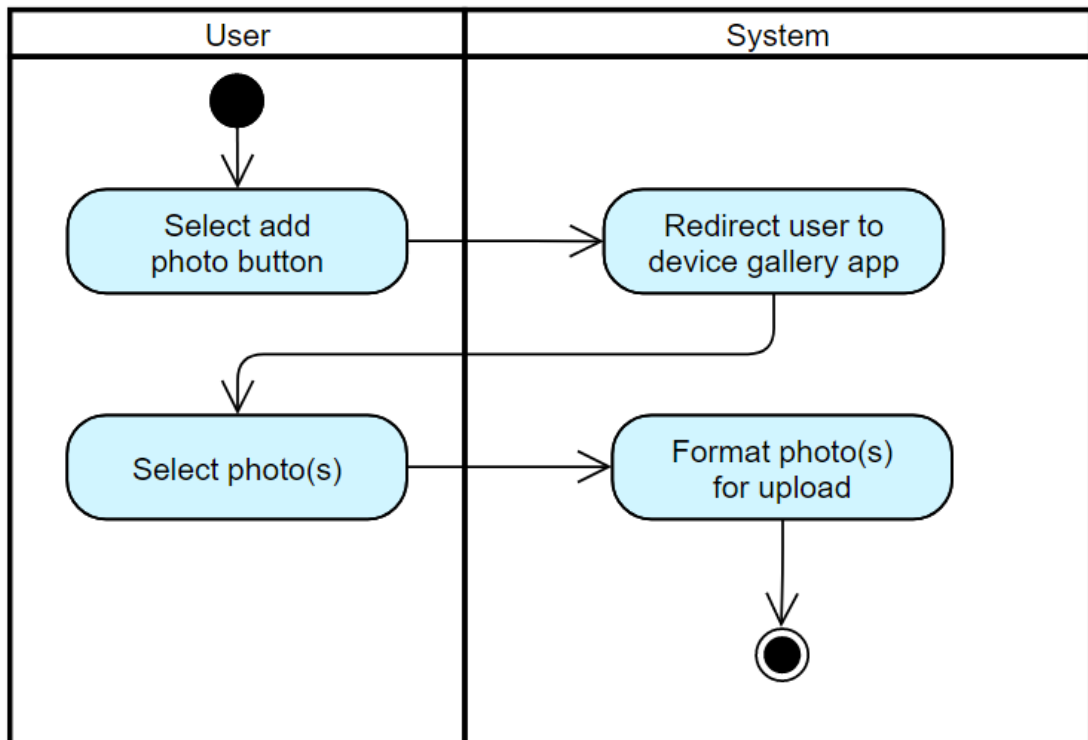


Figure 4.5.13 Add Photos Activity Diagram

xiv) Submit review

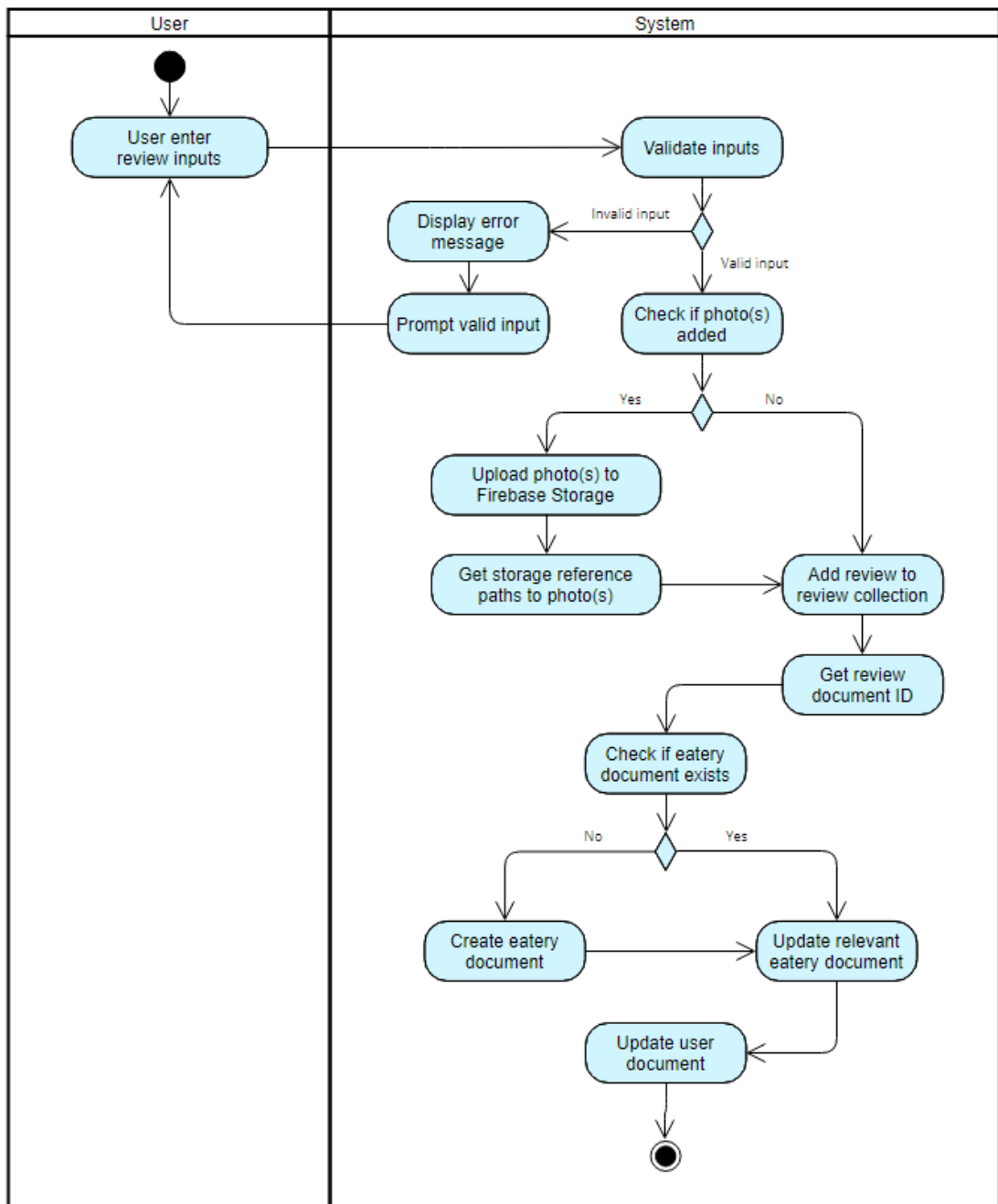


Figure 4.5.14 Submit Review Activity Diagram

xv) View review

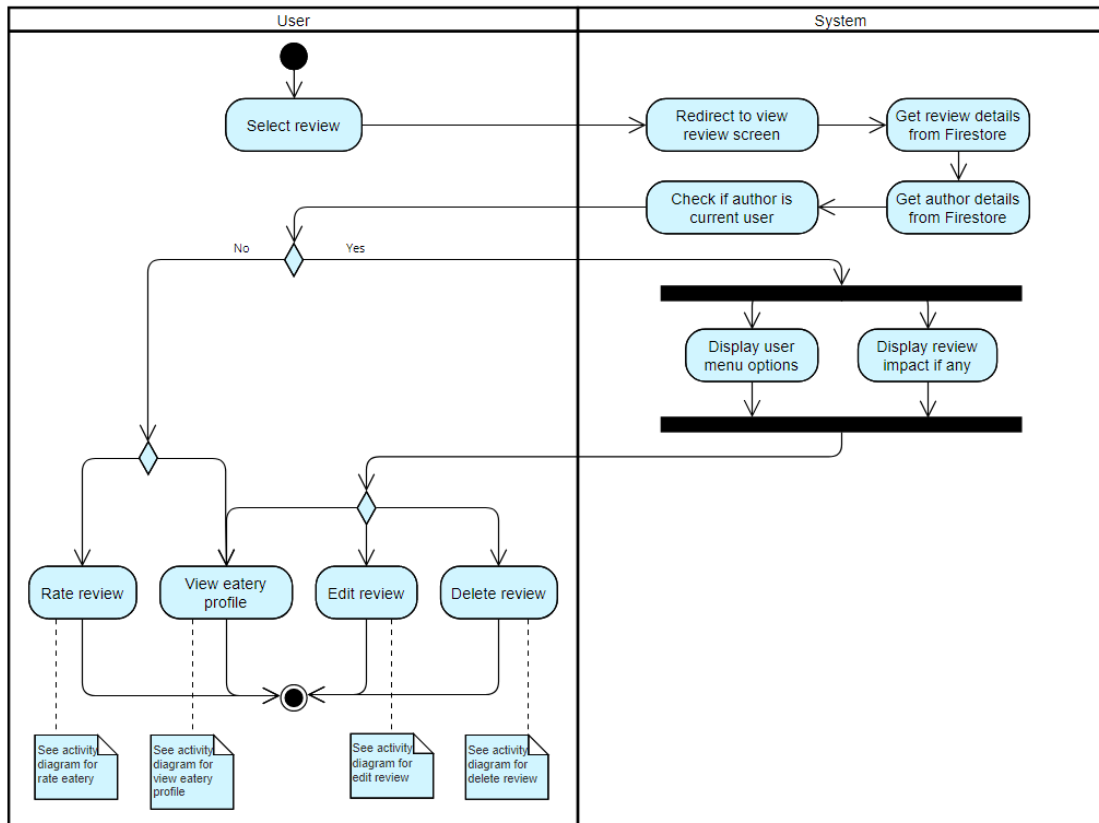


Figure 4.5.15 View Review Activity Diagram

xvi) Rate review

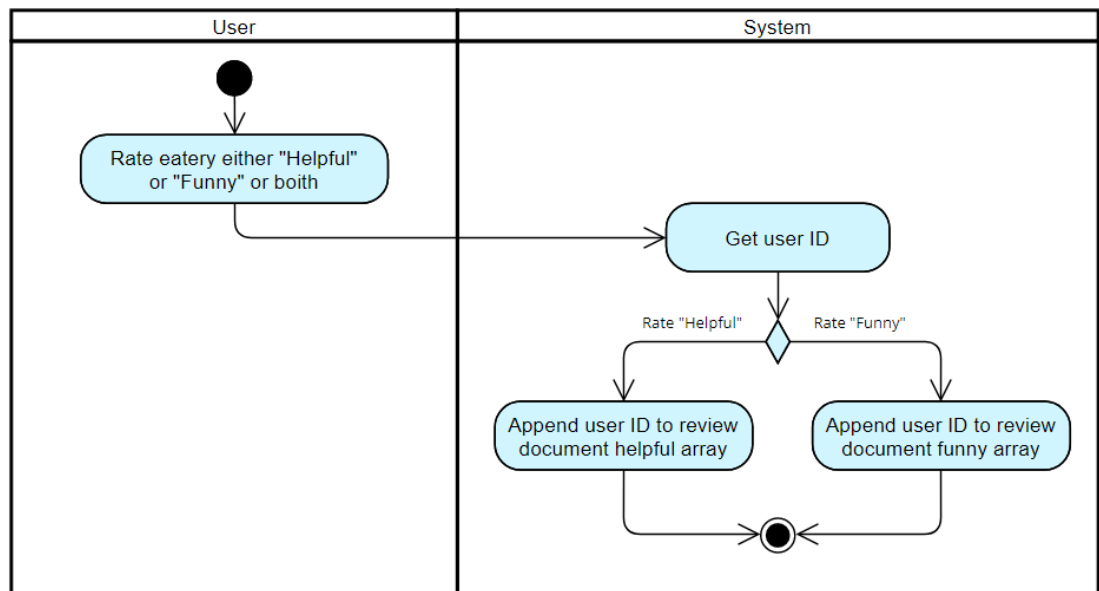


Figure 4.5.16 Rate Review Activity Diagram

xvii) Edit review

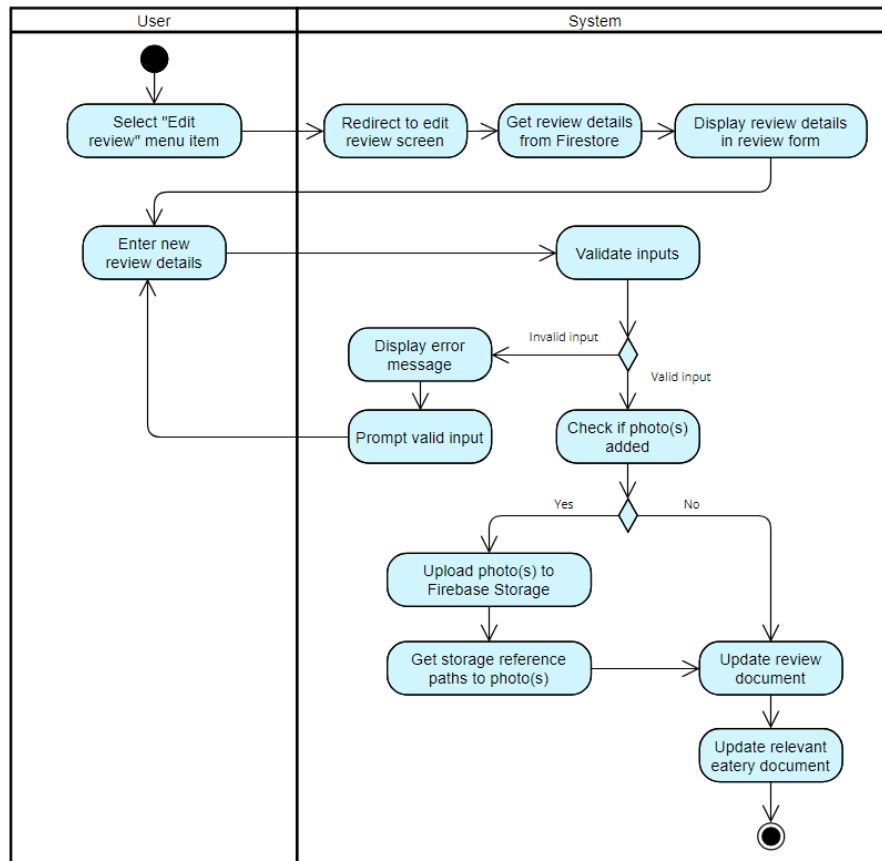


Figure 4.5.17 Edit Review Activity Diagram

xviii) Delete review

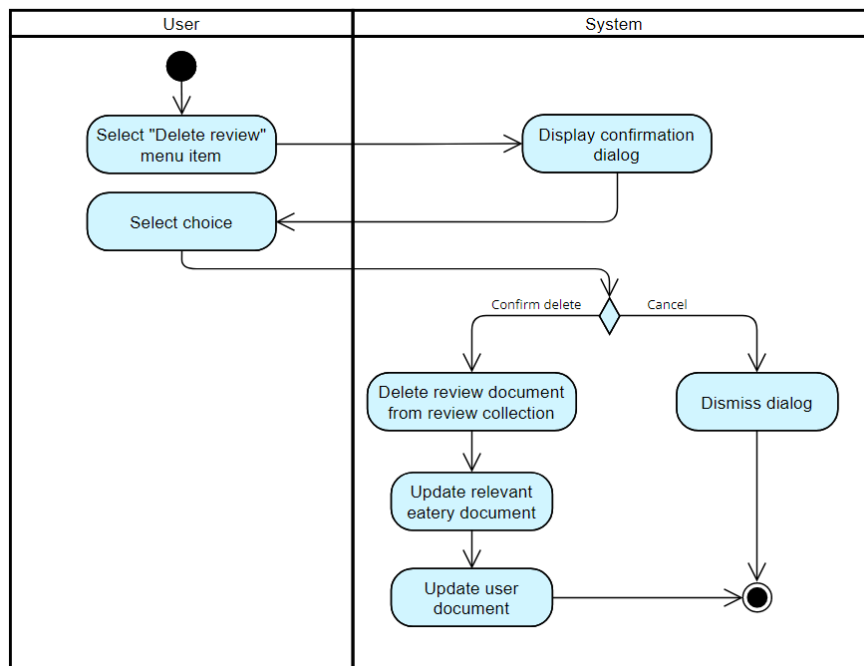


Figure 4.5.18 Delete Review Activity Diagram

xix) View user profile

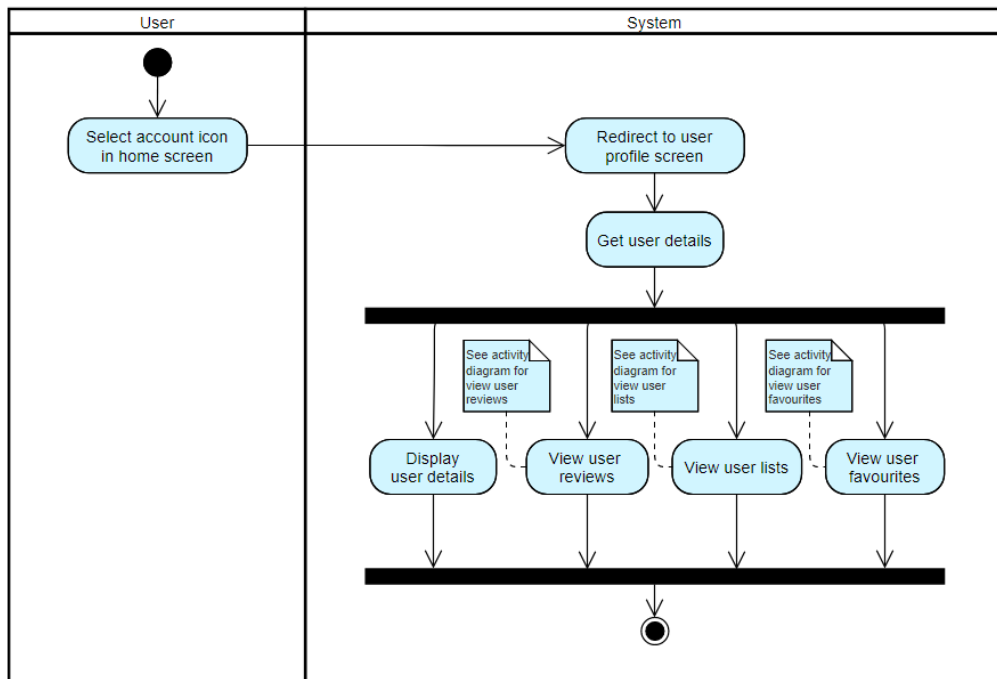


Figure 4.5.19 View User Profile Activity Diagram

xx) View user reviews

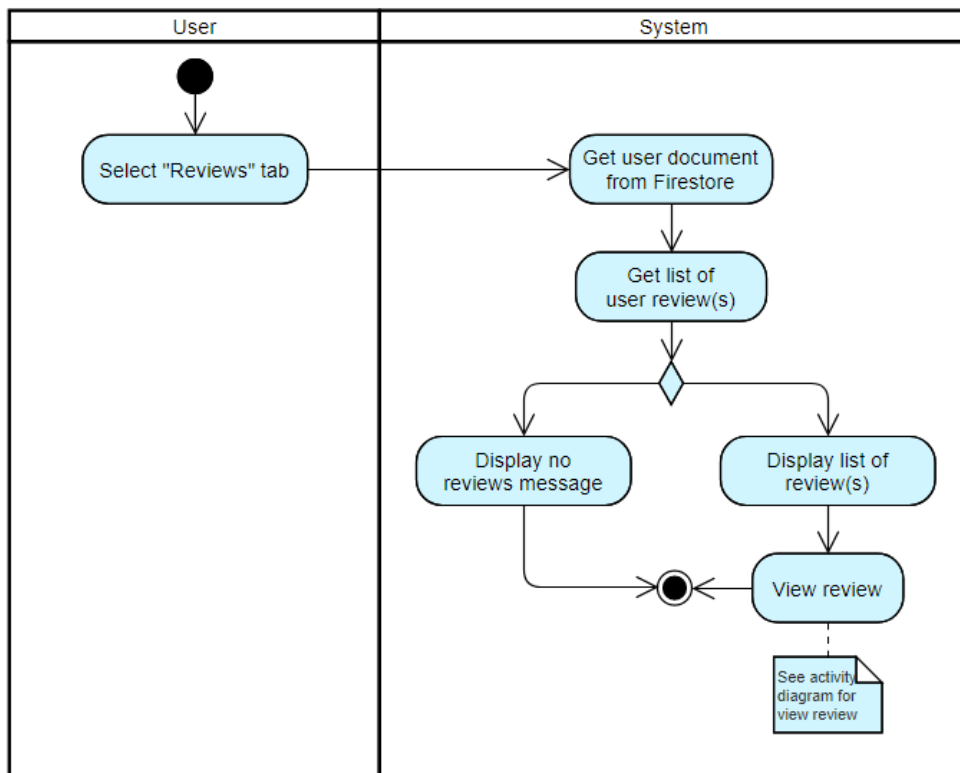


Figure 4.5.20 View User Reviews Activity Diagram

xxi) View user favourites

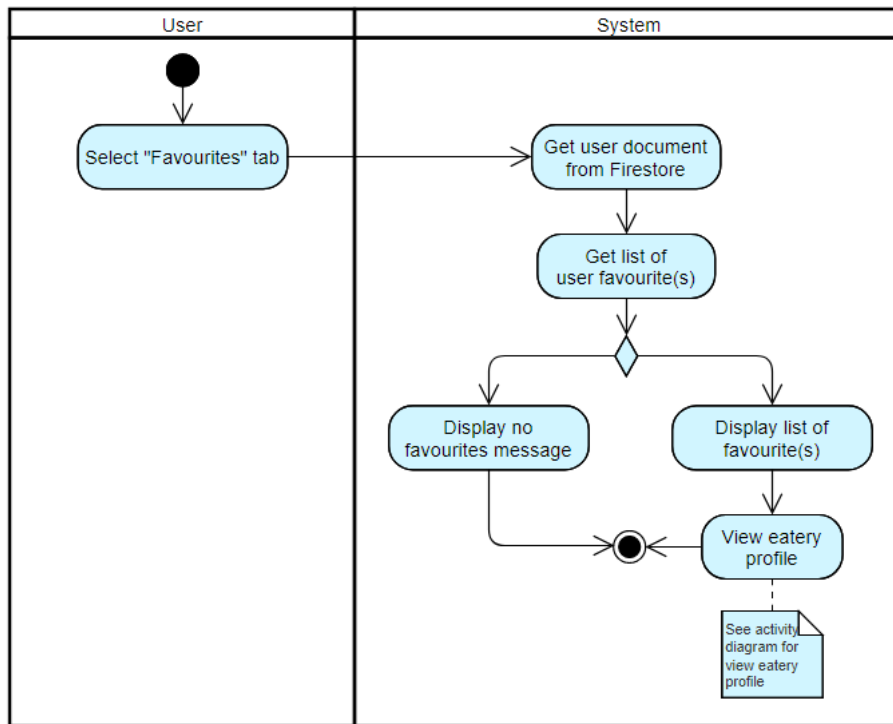


Figure 4.5.21 View User Favourites Activity Diagram

xxii) Unfavourite favourites

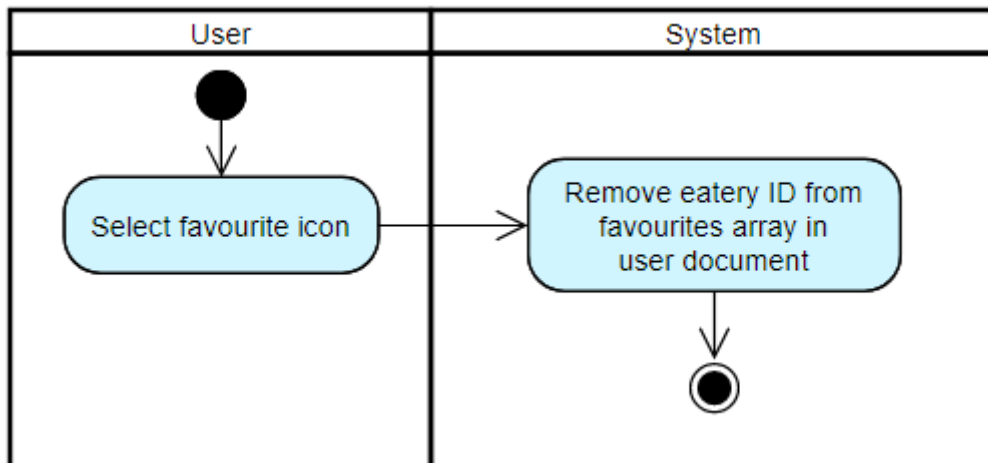


Figure 4.5.22 Unfavourite Eatery Activity Diagram

xxiii) View user lists

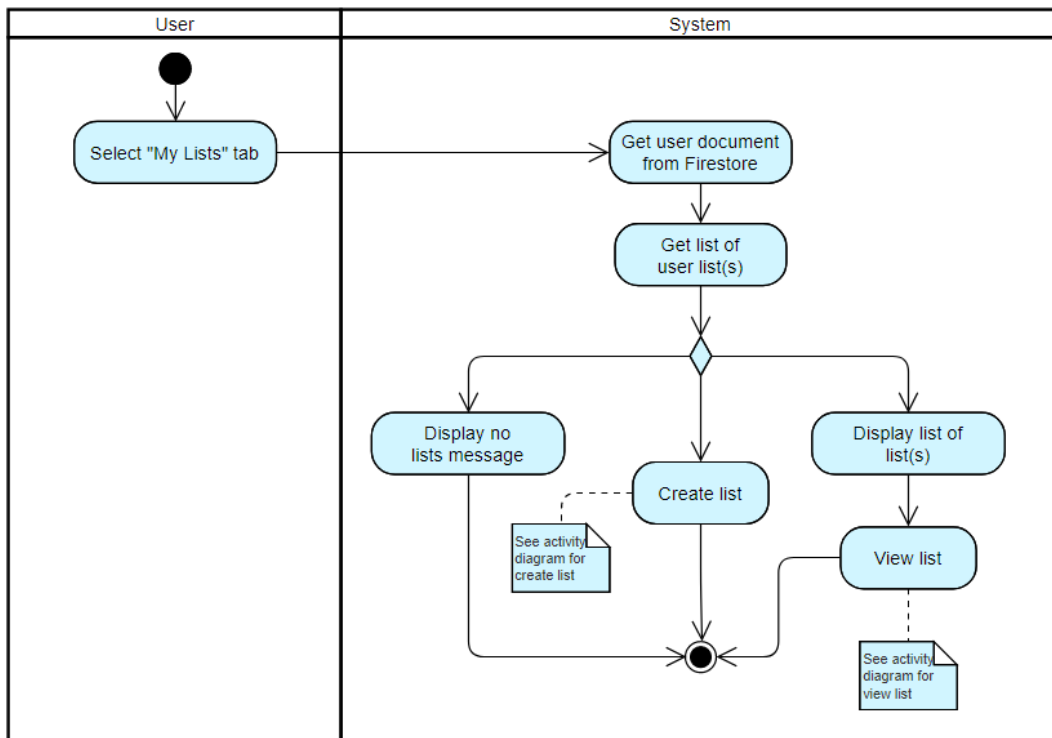


Figure 4.5.23 View User Lists Activity Diagram

xxiv) Create list

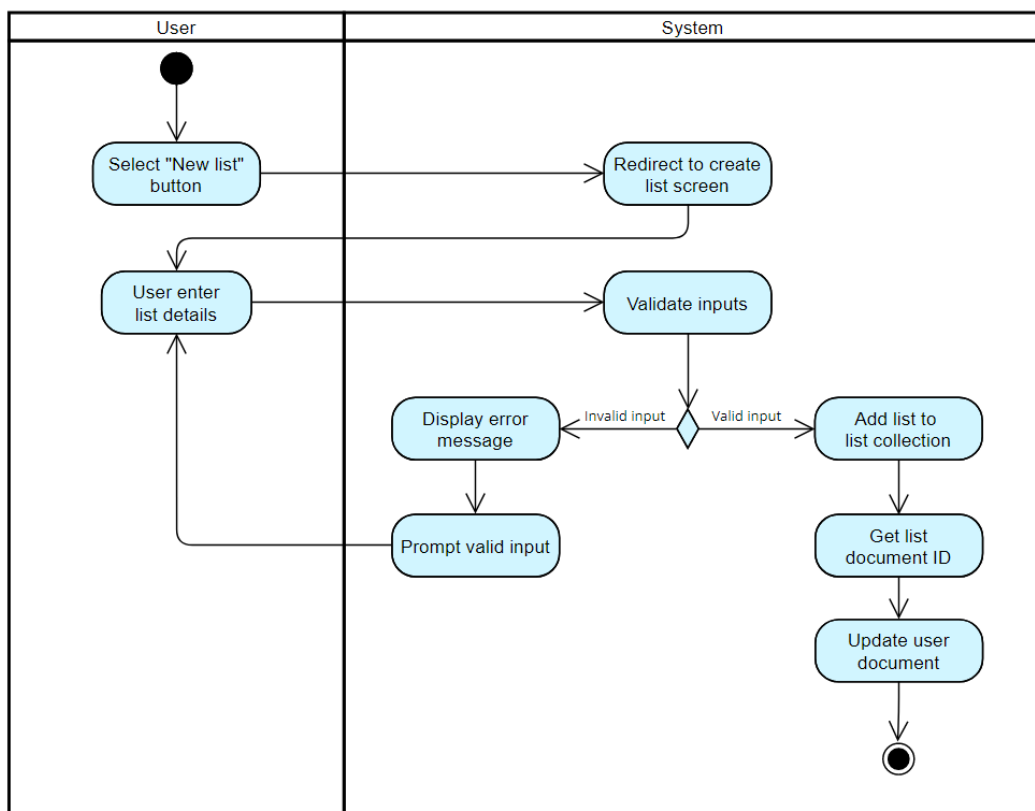


Figure 4.5.24 Create List Activity Diagram

xxv) View list

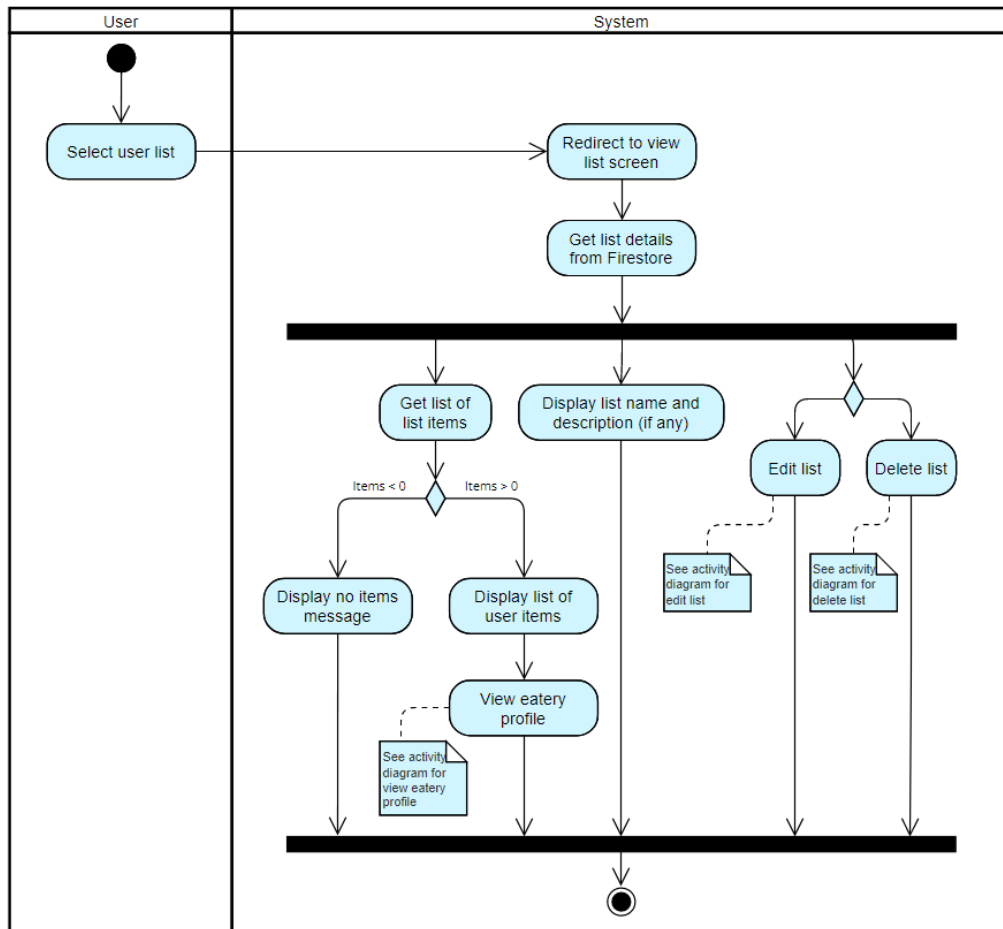


Figure 4.5.25 View List Activity Diagram

xxvi) Edit list

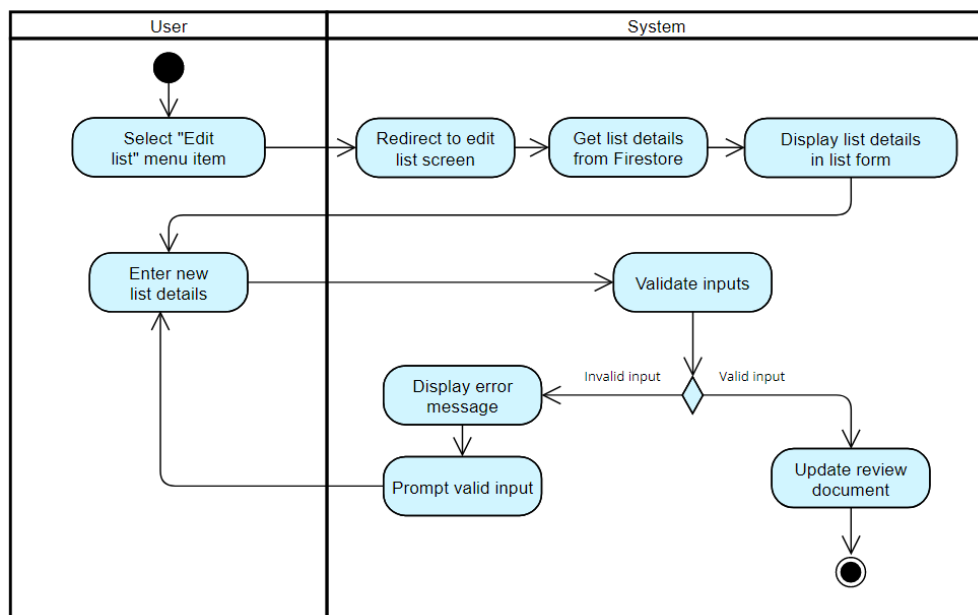


Figure 4.5.26 Edit List Activity Diagram

xxvii) Delete list

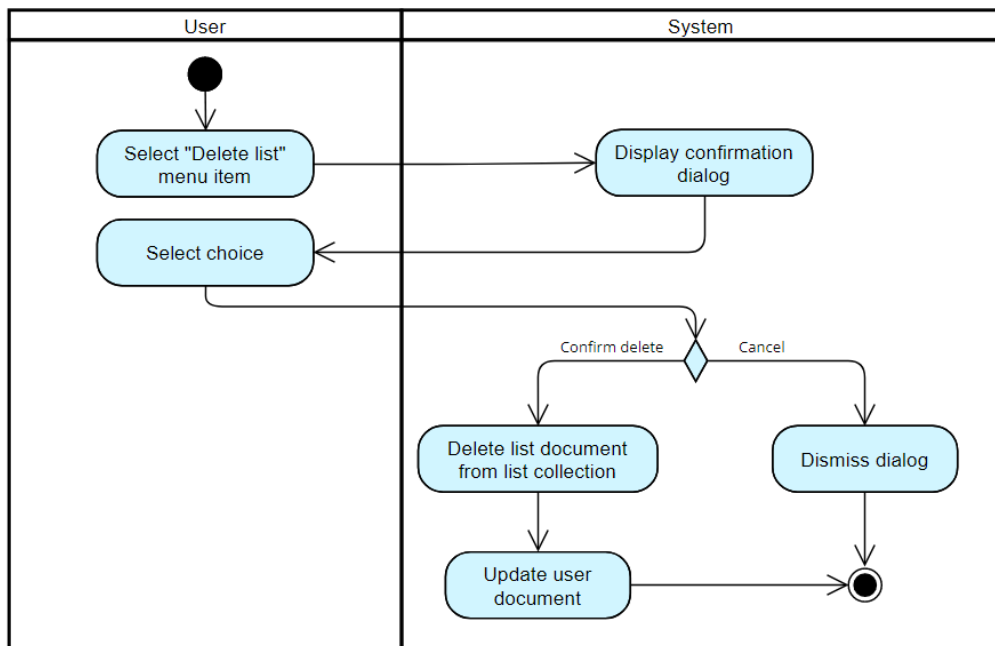


Figure 4.5.27 Delete List Activity Diagram

xxviii) Manage account

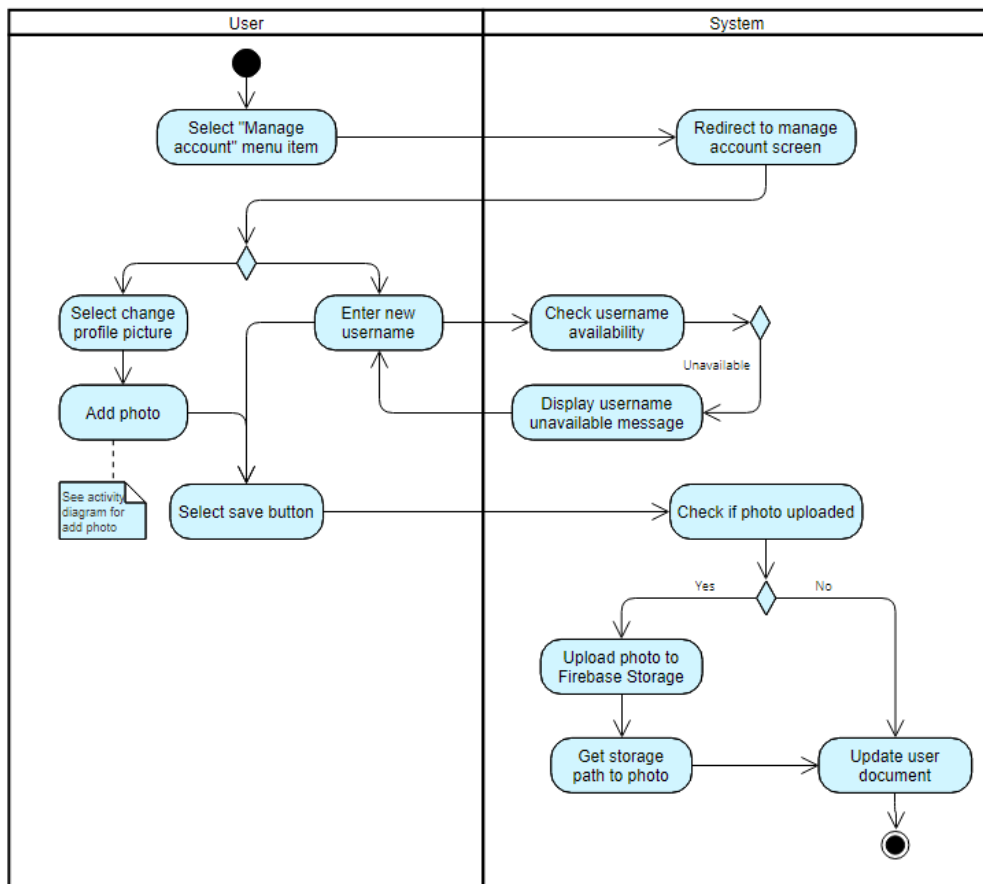


Figure 4.5.28 Manage Account Activity Diagram

4.6 Database Design

Collection	Document Fields (Type)	Description
user	email (string)	The email of the user
	username (string)	The username of the user
	picture (string)	Firestore storage path to user profile picture
	favourites (array)	A list of Place IDs of places which users' favour
	reviews (array)	A list of <i>review</i> document IDs – reviews created by user
	lists (array)	A list of <i>list</i> document IDs of user lists
eatery	name (string)	The name of the eatery
	rating (number)	The average overall rating of the eatery
	totalDineIn (number)	The total overall rating for dine in reviews
	totalTakeaway (number)	The total overall rating for takeaway reviews
	dineInCount (number)	The total number of dine in reviews
	takeawayCount (number)	The total number of takeaway reviews
	reviews (array)	A list of <i>review</i> document IDs
	totalOverall (number)	The total overall rating for all reviews
review	userID (string)	The <i>user</i> document ID of the author
	eateryID (string)	The <i>eatery</i> document ID which the review is about
	eateryName (string)	The name of the eatery
	dining (number)	The dining type the review is about: 1 – dine in, 2 – takeaway
	overall (number)	The overall rating of the eatery
	cleanliness (number)	The cleanliness rating of the eatery
	fbQuality (number)	The F&B quality rating of the eatery
	service (number)	The service rating of the eatery
	text (string)	The text review by the author
	photos (array)	A list of Firestore storage paths to the review's photos
	tags (array)	A list of strings of tags
	createdAt (timestamp)	When the review was created or last updated
	modifiedAt (timestamp)	When the review was last updated
	helpful (number)	The number of users who found the review useful
funny (number)	The number of users who found the review funny	
list	name (string)	The name of the list
	description (string)	The description of the list
	userID (string)	The <i>user</i> document ID of the author
	eateries (array)	A list of references to “eatery” documents

Table 4.6.1 Document Structures

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 Technologies and Tools Involved

5.1.1 Hardware Setup

The specifications of the device used for the development of this project are as follows:

Model	Lenovo Legion: Legion 5 15ARH05
Processor	AMD Ryzen™ 5 4600H Processor (6 Cores / 12 Threads, 3.00 GHz, up to 4.00 GHz with Max Boost, 3 MB Cache L2 / 8 MB Cache L3)
Memory	16 GB SO-DIMM DDR4 3200 MHz
System type	64-bit operating system, x64-based processor
Operating System	Windows 11

Table 5.1.1.1 Laptop Specification

5.1.2 Software Setup

i) Android Studio

The project is developed using Android Studio, the official IDE for Google's Android OS. It includes various SDK tools needed for building Android applications. For this project, Android Studio Bumblebee is used, and the version details are shown in the figure below. Additionally, the programming language used for the project is Java. The IDE is also equipped with an AVD Manager, an interface for creating and managing AVDs. For this project, a virtual Google Pixel 4a with Android API 32 was mainly used for the development and testing of the application. Additionally, a variety of external binaries and library modules is implemented in the project build as dependencies for supporting the use of APIs as well as miscellaneous application functions.



Figure 5.1.2.1 Android Studio Version Details

ii) Google Cloud Platform



Figure 5.1.2.2 Google Cloud Logo

Google Cloud Platform offers a variety of hosted computing, storage, and application development services. For this project, several of the APIs offered by the platform is used to develop the application and thus, the platform was required to implement most of the application's back-end functions. A cloud project for EatMore was created where the APIs for the Google Places API were enabled. The relevant credentials and API key were also created.

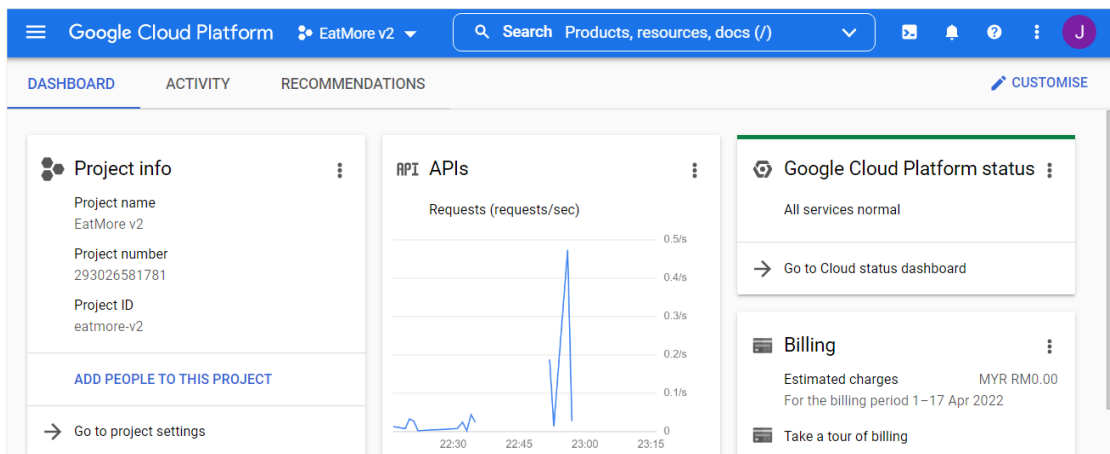


Figure 5.1.2.3 EatMore Cloud Console Dashboard

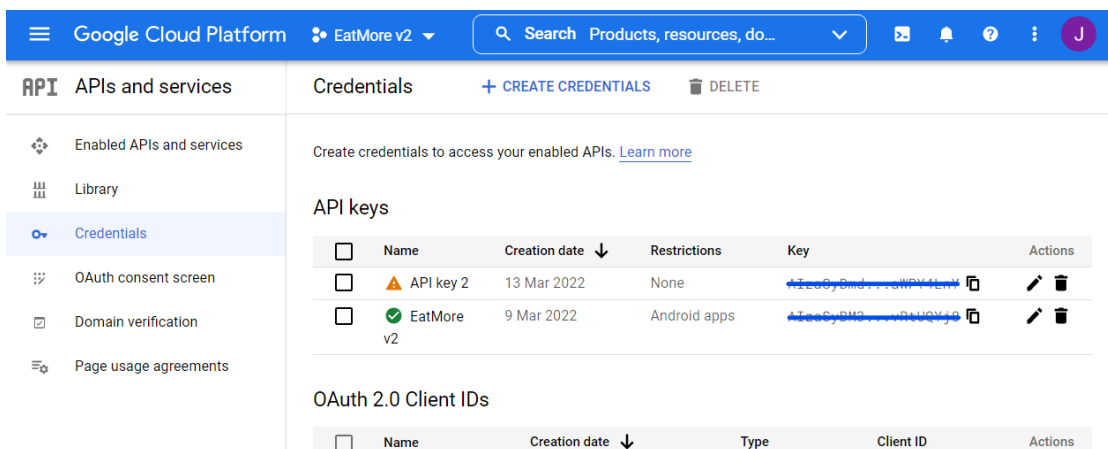


Figure 5.1.2.4 EatMore API Keys

iii) Google Maps Platform



Figure 5.1.2.5 Google Maps Platform Logo

Google Maps Platform is a suite of APIs and SDKs that developers can use to integrate Google Maps into mobile and web applications, as well as to obtain data from the platform. Google Maps Platform is integrated with the Google Cloud Platform console and as such, is managed through the project console. For the location-based application functions, the Google Places API, a service that returns information about places will be implemented [16]. It is one of the many services offered on the Google Maps Platform. The following features from Places API are used:

1. Place Autocomplete

In response to user queries, the autocomplete service returns a list of place predictions as the user types. This service is used in the Get User Location feature for users to input their location manually.

2. Text Search

Text Search returns information about a list of places based on user queries. This service is used in the Search function where different parameters can be included in the query which is used for the search filter function.

3. Nearby Search

This service returns information about places within a specified area. This service is used for suggesting places in Kampar. Compared to Text Search, this service does not accept ambiguous text.

4. Place Details

Place Details service returns information about a particular business or point of interest. This is used in the View Eatery Profile as well as for showing list of eateries in user lists and favourites. As the fields return can be specified, two different methods were developed to respond to the different needs as seen in the figures below.

```

public void getBriefPlaceDetails(String placeID, BriefPlaceDetailsResponseListener listener) {
    final List<Place.Field> placeFields = Arrays.asList(
        Place.Field.NAME,
        Place.Field.ADDRESS,
        Place.Field.OPENING_HOURS,
        Place.Field.PHOTO_METADATAS);
}

```

Figure 5.1.2.6 Place Details Method 1

```

public void getFullPlaceDetails(String placeID, FullPlaceDetailsResponseListener listener) {

    final List<Place.Field> placeFields = Arrays.asList(
        Place.Field.ADDRESS,
        Place.Field.LAT_LNG,
        Place.Field.NAME,
        Place.Field.OPENING_HOURS,
        Place.Field.PHONE_NUMBER,
        Place.Field.PHOTO_METADATAS,
        Place.Field.PRICE_LEVEL,
        Place.Field.WEBSITE_URI);
}

```

Figure 5.1.2.7 Place Details Method 2

5. Place Photos

This service returns a place photo to be displayed in the application. This is used in most of the application functions where the eatery photo is displayed, namely the home screen, search results, list of eateries in favourites and user lists, as well as eatery profiles.

iv) Firebase



Figure 5.1.2.8 Firebase Logo

Google's Firebase technology supports developers in creating mobile and web applications. Firebase tools and services is integrated into the application for database and user authentication. The tools and services used are described below.

1. Cloud Firestore



Figure 5.1.2.9 Cloud Firestore Logo

Cloud Firestore, a NoSQL document database handles the storage and retrieval of application data. Additionally, the agile approach perfectly complements the use of NoSQL databases which will be utilized in this project. Due to the nature of both agile methods and NoSQL databases, the scalability of the application is increased which allows for a more flexible development process.

2. Firebase Authentication



Figure 5.1.2.10 Firebase Authentication Logo

Besides that, Firebase Authentication which provides authentication using email and passwords, phone numbers and popular federated identity providers such as Google and Facebook. For this project, Firebase Authentication is used for authenticating users using email and passwords. The password reset feature is also supported by this.

3. Cloud Storage



Figure 5.1.2.11 Cloud Storage Logo

Cloud Storage is used to store images from user profile pictures and review photos. The API is simple and provides secure file uploads and downloads.

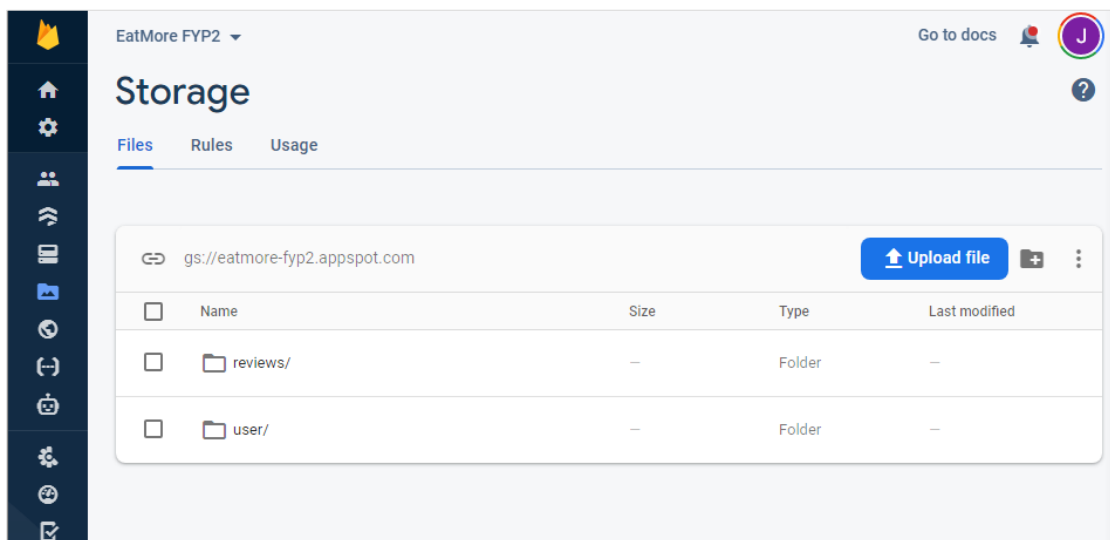


Figure 5.1.2.12 EatMore Cloud Storage Console

v) FirebaseUI for Android and Glide

FirebaseUI is an open-source library that allows developers to easily integrate common UI elements with Firebase APIs. The Firebase UI Storage was used to download and display images which works along with Glide, an image loading and caching library for Android.

vi) Design and Planning Tools

1. Visual Paradigm Online

Visual Paradigm Online is an online drawing software that provides plenty of tools and templates for designing a variety of diagrams. It was used for designing the various system design diagrams for the mobile application.

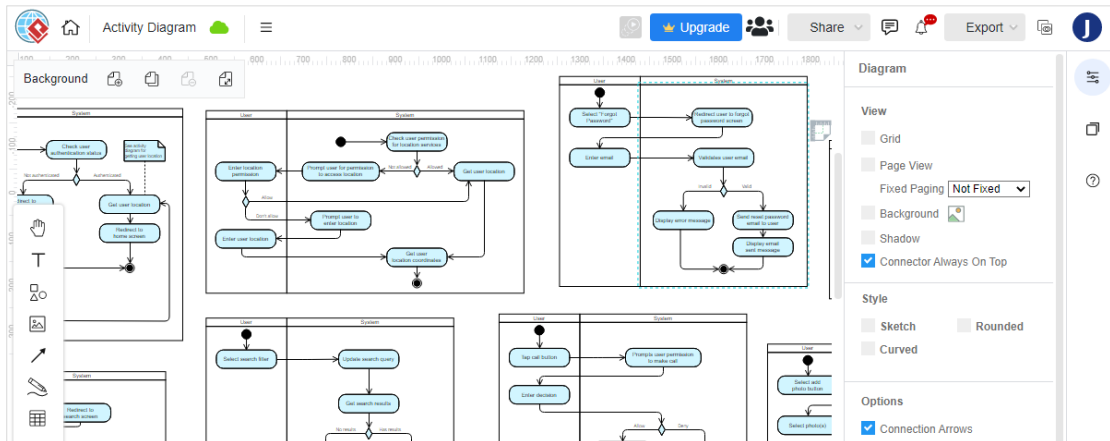


Figure 5.1.2.13 Visual Paradigm Online File

2. Figma

Figma is a web-based graphics editing and UI design platform that is used for planning and designing the overall design theme and UIs. For this project, illustrations and vector design ideas are mainly sourced from Figma's community content as well as through SVG exports.

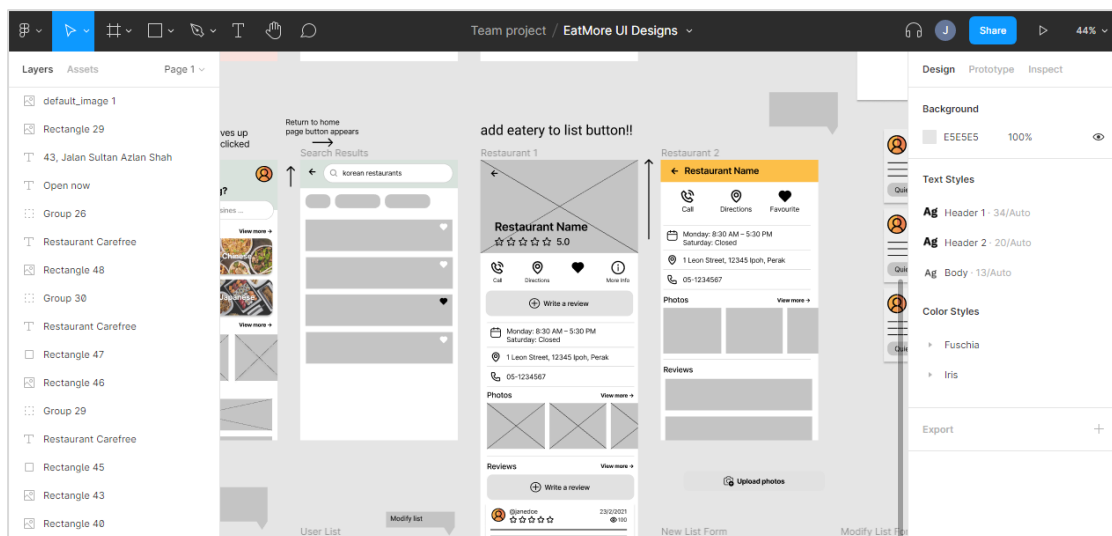


Figure 5.1.2.14 EatMore Figma Project

3. Canva

For the system block diagram and poster of the project, the designs were made using Canva, a graphic design platform which also offers many templates and design ideas.

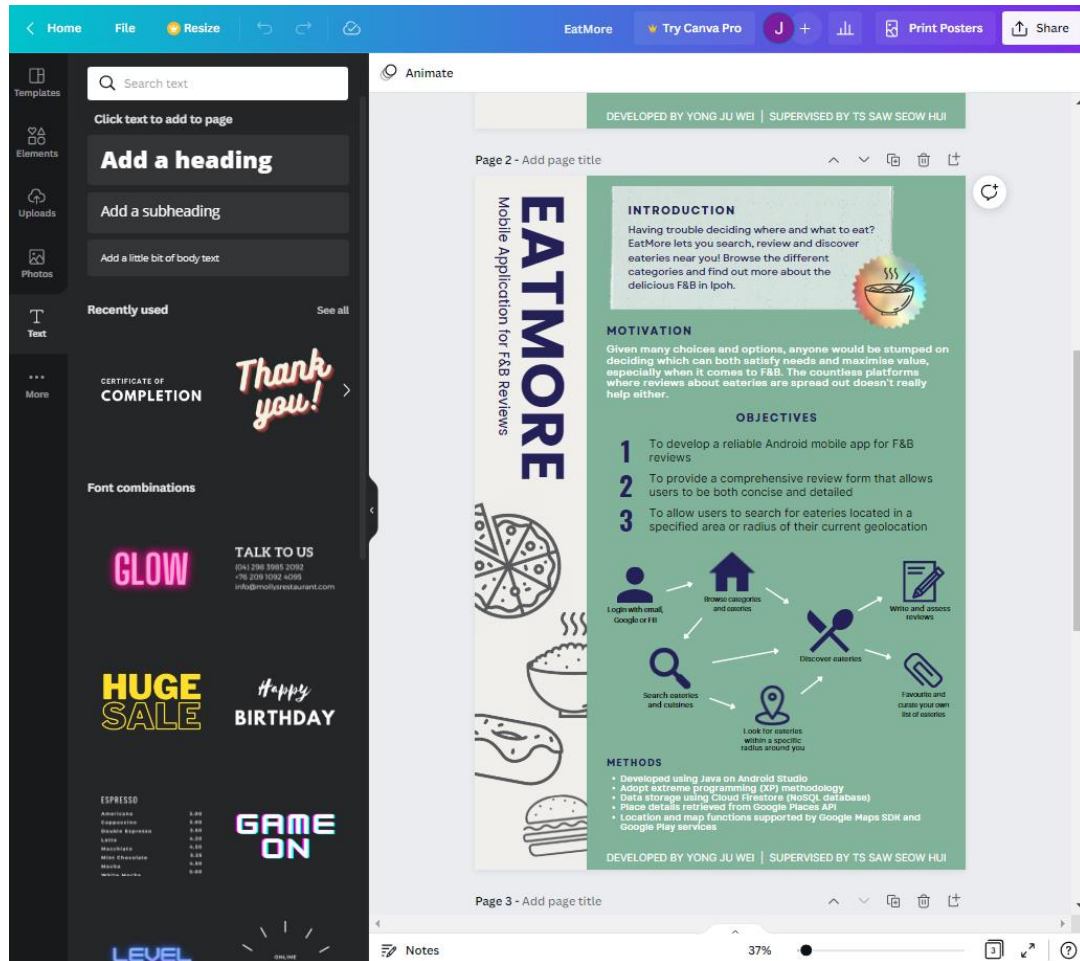


Figure 5.1.2.15 EatMore Canva Project

4. TeamGantt



Figure 5.1.2.16 TeamGantt Logo

TeamGantt is an online project planning software that is used to plan and track the project timeline.

5.2 Project Timeline

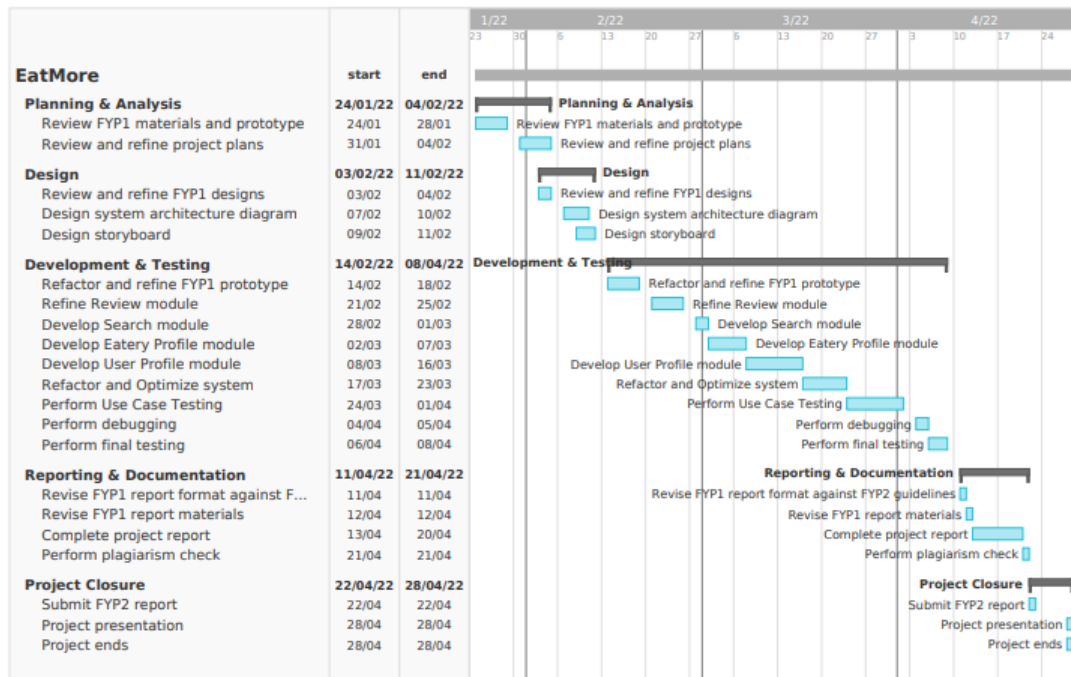


Figure 5.2.1 Project Timeline

5.3 System UI

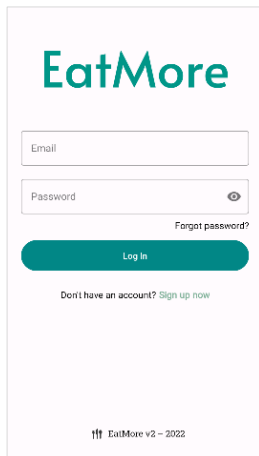


Figure 5.3.1 Log In

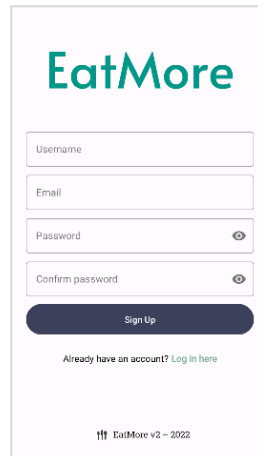


Figure 5.3.2 Sign Up

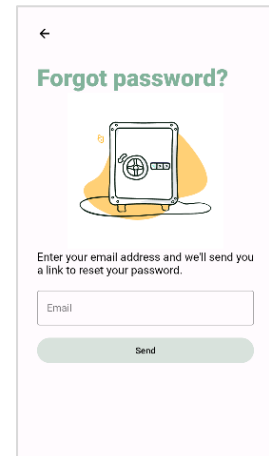


Figure 5.3.3 Forgot Password

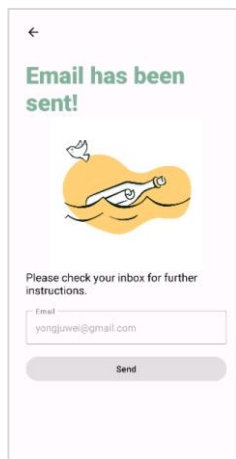


Figure 5.3.4 Forgot Password (Email Sent)

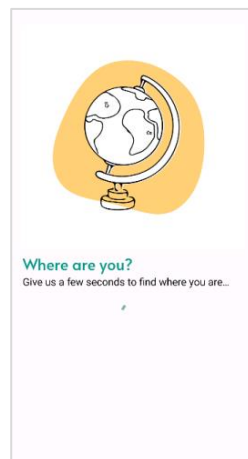


Figure 5.3.5 Get Device Location

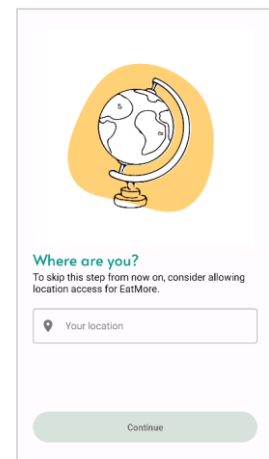


Figure 5.3.6 Input Manual Location

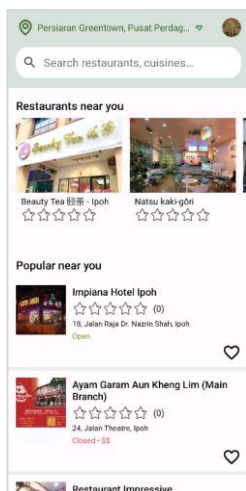


Figure 5.3.7 Home

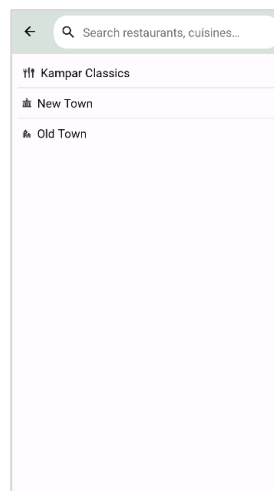


Figure 5.3.8 Search

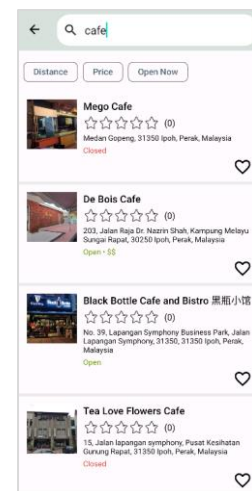


Figure 5.3.9 Search Results

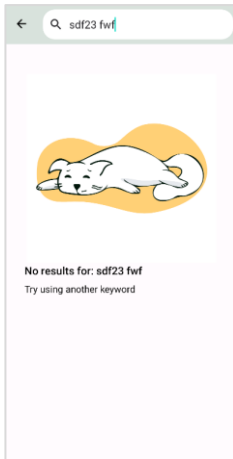


Figure 5.3.10 No Search Results

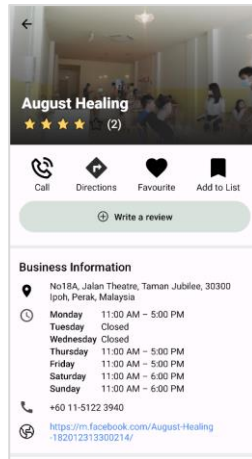


Figure 5.3.11 Eatery Profile General Section

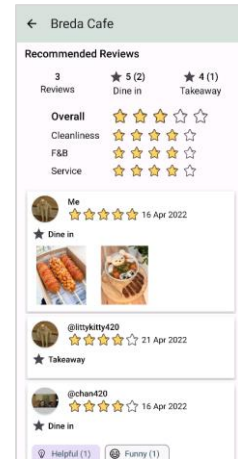


Figure 5.3.12 Eatery Profile Review Section

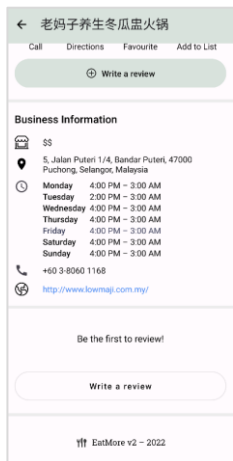


Figure 5.3.13 Eatery Profile with No Reviews

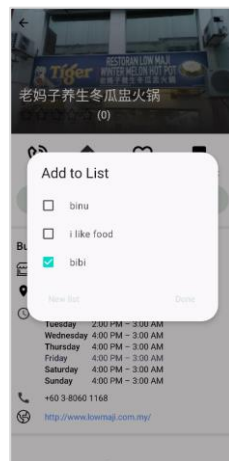


Figure 5.3.14 Add to List Dialog

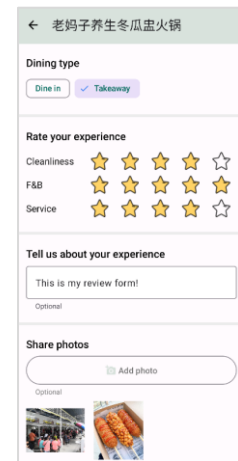


Figure 5.3.15 Review Form

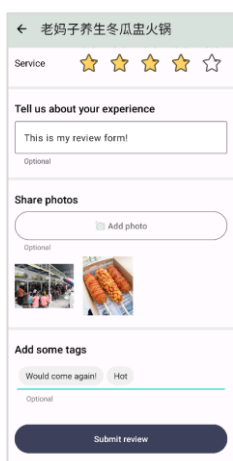


Figure 5.3.16 Review Form

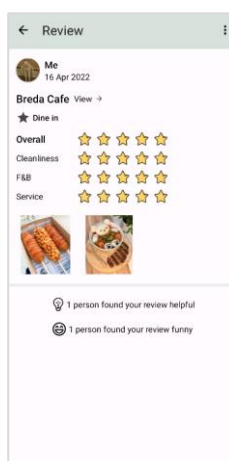


Figure 5.3.17 View Own Review

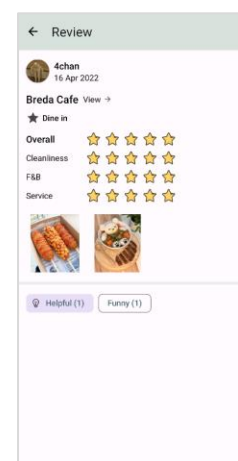


Figure 5.3.18 View Other User Review



Figure 5.3.19 Create List

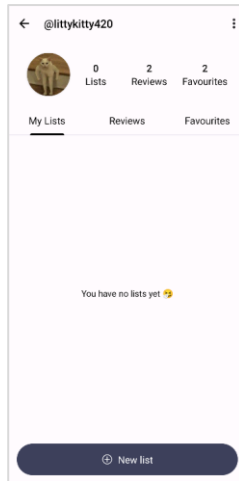


Figure 5.3.20 User Lists
(No lists)

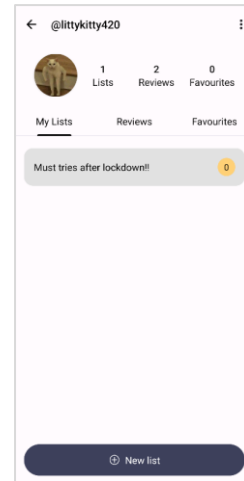


Figure 5.3.21 User Lists

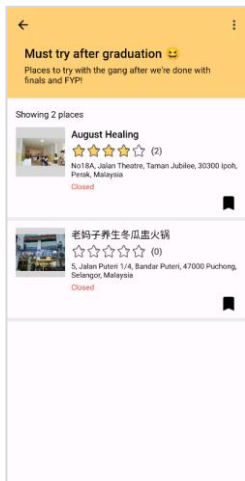


Figure 5.3.22 View List

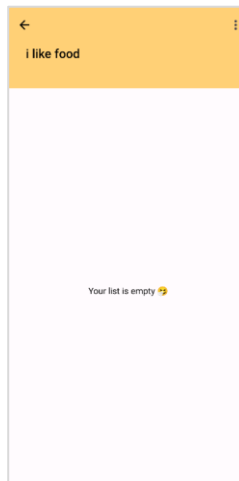


Figure 5.3.23 View List
(No items)

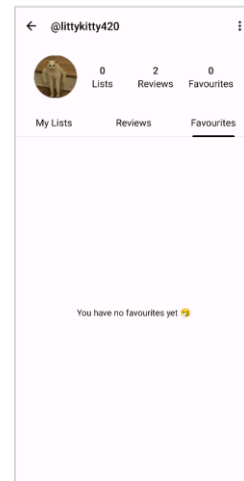


Figure 5.3.24 Favourites
(No favourites)

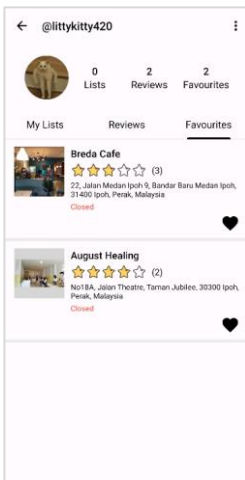


Figure 5.3.25 User
Favourites

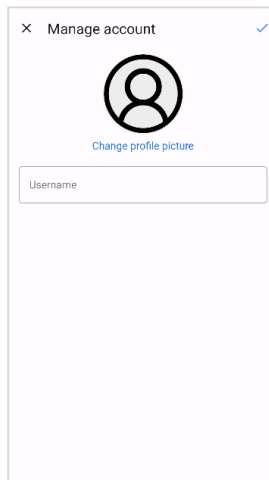


Figure 5.3.26 Manage
Account

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION

6.1 System Testing and Performance Metrics



Figure 6.1.1 Black Box Testing

Black box testing, also known as Behavioural Testing, is a software testing technique that involves testing software applications without knowing the internal code structure, implementation details and internal paths. For this project, use case testing which is a type of functional black box testing is mainly performed to test the functional requirements of the system. It facilitates testing communications amongst integrated modules.

6.1.1 Testing Setup and Result

Use Case	UC001 – Log in		
Test Case	Expected Output	Actual Output	Result
User logs in with valid credentials	User is authenticated and redirected to get user location screen	User is authenticated and redirected to get user location screen	Pass
User logs in with invalid credentials	Error message displayed	Error message displayed	Pass
User logs in with no credentials	Error message displayed	Error message displayed	Pass
User taps “Forgot password” button	User is redirected to forgot password screen	User is redirected to forgot password screen	Pass
User taps “Sign up now” button	User is redirected to registration screen	User is redirected to registration screen	Pass

Table 6.1.1.1 Log In Use Case Testing

Use Case	UC002 – Log out		
Test Case	Expected Output	Actual Output	Result
User taps “Log out” menu item in user profile screen	User is logged out and redirected to login screen	User is logged out and redirected to login screen	Pass

Table 6.1.1.2 Log Out Use Case Testing

Use Case	UC003 – Sign up		
Test Case	Expected Output	Actual Output	Result
User signs up with valid inputs	<ol style="list-style-type: none"> 1. User is authenticated 2. User is redirected to get user location screen 3. User document is created in database 	<ol style="list-style-type: none"> 1. User is authenticated 2. User is redirected to get user location screen 3. User document is created in database 	Pass
User signs up with email that is already registered	Error message displayed	Error message displayed	Pass
User inputs taken username	Username taken error message displayed	Username taken error message displayed	Pass
User inputs taken username and then change to available username	Error message removed	Error message removed	Pass
User signs up with taken username	Username taken error message displayed	Username taken error message displayed	Pass
User signs up with invalid email	Error message displayed	Error message displayed	Pass
User signs up with password length less than 6 characters	Error message displayed	Error message displayed	Pass
User password different from confirm password input	Error message displayed	Error message displayed	Pass
User taps “Log in here” button	User is redirected to login screen	User is redirected to login screen	Pass

Table 6.1.1.3 Sign Up Use Case Testing

Use Case	UC004 – Forgot Password		
Test Case	Expected Output	Actual Output	Result
User enters registered email	<ol style="list-style-type: none"> 1. Email sent message displayed 2. Input field disabled 3. Email sent to user 	<ol style="list-style-type: none"> 1. Email sent message displayed 2. Input field disabled 3. Email sent to user 	Pass
User enters invalid email	Error message displayed	Error message displayed	Pass
User taps on navigation icon	User is redirected to login screen	User is redirected to login screen	Pass

Table 6.1.1.4 Forgot Password Use Case Testing

Use Case	UC005 – Get user location, UC006 – Get device location		
Test Case	Expected Output	Actual Output	Result
User logs in on the application for the first time	System requests user permission to access location	System requests user permission to access location	Pass
User logs in and previously denied permission to access location	System displays prompt for user to enter manual location	System displays prompt for user to enter manual location	Pass
User allows permission to access location	<ol style="list-style-type: none"> System displays progress bar System gets user location User is redirected to home screen 	<ol style="list-style-type: none"> System displays progress bar System gets user location User is redirected to home screen 	Pass
User denies permission to access location	System prompts user for manual location	System prompts user for manual location	Pass

Table 6.1.1.5 Get User Location and Get Device Location Use Case Testing

Use Case	UC007 – Input location manually		
Test Case	Expected Output	Actual Output	Result
User taps input field to enter location	Place Autocomplete search is displayed	Place Autocomplete search is displayed	Pass
User selects a place from Place Autocomplete	<ol style="list-style-type: none"> System gets place address and coordinates Address of selected place is displayed in location input field 	<ol style="list-style-type: none"> System gets place address and coordinates Address of selected place is displayed in location input field 	Pass
User taps “Continue” button after input a location	User is redirected to home screen	User is redirected to home screen	Pass
User taps “Continue” button without input a location	Error message is displayed	Error message is displayed	Pass

Table 6.1.1.6 Input Location Manually Use Case Testing

Use Case	UC008 – Search eatery, UC009 – Sort & filter search results		
Test Case	Expected Output	Actual Output	Result
User enters a search query	Suggestion layout hidden	Suggestion layout hidden	Pass
No search results found	No results found message displayed	No results found message displayed	Pass
There are search results	<ol style="list-style-type: none"> Search filters are displayed 	<ol style="list-style-type: none"> Search filters are displayed 	Pass

	2. Results are displayed in RecyclerView	2. Results are displayed in RecyclerView	
User applies search filter	Search results are filtered based on filter choice	Search results are filtered based on filter choice	Pass
User resets search filter	Original search results are shown	Original search results are shown	Pass
User taps on search result item	User is redirected to the profile screen for selected item	User is redirected to the profile screen for selected item	Pass

Table 6.1.1.7 Search and Sort & Filter Search Results Use Case Testing

Use Case	UC010 – View eatery profile		
Test Case	Expected Output	Actual Output	Result
User selects an eatery to view profile	<ol style="list-style-type: none"> User is redirected to eatery profile Progress bar hidden when eatery details loaded 	<ol style="list-style-type: none"> User is redirected to eatery profile Progress bar hidden when eatery details loaded 	Pass
Eatery contact number is available	Call button and eatery contact number is displayed	Call button and eatery contact number is displayed	Pass
Eatery is a user favourite	Filled favourite icon displayed	Filled favourite icon displayed	Pass
Eatery is not a user favourite	Hollow favourite icon displayed	Hollow favourite icon displayed	Pass
Eatery is in a user list	Filled list icon displayed	Filled list icon displayed	Pass
Eatery is not in any user list	Hollow list icon displayed	Hollow list icon displayed	Pass
Eatery opening hours are available	Opening hours are displayed	Opening hours are displayed	Pass
Eatery website is available	Eatery website is displayed	Eatery website is displayed	Pass
Eatery does not have reviews	<ol style="list-style-type: none"> Ratings and review count is 0 No reviews message is displayed 	<ol style="list-style-type: none"> Ratings and review count is 0 No reviews message is displayed 	Pass
Eatery has reviews	<ol style="list-style-type: none"> Ratings and count for overall, dine in and takeaway is displayed Overall rating for overall, cleanliness, F&B and service is displayed 	<ol style="list-style-type: none"> Ratings and count for overall, dine in and takeaway is displayed Overall rating for overall, cleanliness, F&B and service is displayed 	Pass

	3. Reviews are displayed in RecyclerView	3. Reviews are displayed in RecyclerView	
--	--	--	--

Table 6.1.1.8 View Eatery Profile Use Case Testing

Use Case	UC011 – Call eatery		
Test Case	Expected Output	Actual Output	Result
User taps call button	System requests user permission to make phone call	System request user permission to make phone call	Pass
User taps call button and allows app to make call	System makes phone call to eatery contact number	System makes phone call to eatery contact number	Pass
User taps call button and denies app permission to make call	Permission denied message displayed	Permission denied message denied	Pass
User taps eatery contact number text button	System requests user permission to make phone call	System request user permission to make phone call	Pass
User taps eatery contact number text button and allows app to make call	System makes phone call to eatery contact number	System makes phone call to eatery contact number	Pass
User taps eatery contact number text button and denies app permission to make call	Permission denied message displayed	Permission denied message denied	Pass

Table 6.1.1.9 Call Eatery Use Case Testing

Use Case	UC012 – Get directions to eatery		
Test Case	Expected Output	Actual Output	Result
User taps get direction button	User is redirected to Google Maps with eatery set as input	User is redirected to Google Maps with eatery set as input	Pass
User taps eatery address	User is redirected to Google Maps with eatery set as input	User is redirected to Google Maps with eatery set as input	Pass

Table 6.1.1.10 Get Directions to Eatery Use Case Testing

Use Case	UC013 – Favourite eatery, UC023 – Unfavourite eatery		
Test Case	Expected Output	Actual Output	Result
User taps favourite icon for eatery that is not user favourite	<ol style="list-style-type: none"> Eatery ID is added to user document “favourite” array Filled favourite icon is displayed 	<ol style="list-style-type: none"> Eatery is added to user document “favourite” array Filled favourite icon is displayed 	Pass
User taps favourite button for favourite eatery	<ol style="list-style-type: none"> Eatery ID is removed from user document “favourite” array Hollow favourite icon is displayed 	<ol style="list-style-type: none"> Eatery is removed from user document “favourite” array Hollow favourite icon is displayed 	Pass

Table 6.1.1.11 Favourite and Unfavourite Eatery Use Case Testing

Use Case	UC014 – Add eatery to list		
Test Case	Expected Output	Actual Output	Result
User taps add to list icon and user has lists	<ol style="list-style-type: none"> Add to list dialog displayed User lists are displayed as check options 	<ol style="list-style-type: none"> Add to list dialog displayed User lists are displayed as check options 	Pass
User checks list in add to list dialog	<ol style="list-style-type: none"> Eatery ID is added to list document “eateries” array Filled list icon is displayed 	<ol style="list-style-type: none"> Eatery ID is added to list document “eateries” array Filled list icon is displayed 	Pass
User unchecks list in add to list dialog	Eatery ID is removed from list document “eateries” array	Eatery is removed from list document “eateries” array	Pass
User taps “New list” button	User is redirected to create list screen	User is redirected to create list screen	Pass
User taps “Done” button	Dialog is dismissed	Dialog is dismissed	Pass
User taps area outside dialog	Dialog is dismissed	Dialog is dismissed	Pass

Table 6.1.1.12 Add Eatery To List Use Case Testing

Use Case	UC015 – Write review, UC018 – Edit review, UC030 – Add dining type, UC031 – Rate eatery, UC033 – Add tags, UC034 – Write text review, UC035 – Submit Review		
Test Case	Expected Output	Actual Output	Result

User taps "Write review" button	User is redirected to write review screen	User is redirected to write review screen	Pass
User submits review with only dining type selected	Error message displayed	Error message displayed	Pass
User submits review without input ratings	Error message displayed	Error message displayed	Pass
User submits review without any input	Error message displayed	Error message displayed	Pass
User taps "Submit review" button and all input are valid	<ol style="list-style-type: none"> 1. New review document is added in database 2. Review document ID is added to user and eatery document "reviews" array 3. Eatery document rating and relevant dining type total and count is updated 4. User is redirected to eatery profile screen 	<ol style="list-style-type: none"> 1. New review document is added in database 2. Review document ID is added to user and eatery document "reviews" array 3. Eatery document rating and relevant dining type total and count is updated 4. User is redirected to eatery profile screen 	Pass
User submits review and includes photos	<ol style="list-style-type: none"> 1. Photos are uploaded to Cloud Storage 2. Path to photos in storage is added to review 3. New review document is added in database 4. Review document ID is added to user and eatery document "reviews" array 5. Eatery document rating and relevant dining type total and count is updated 6. User is redirected to eatery profile screen 	<ol style="list-style-type: none"> 1. Photos are uploaded to Cloud Storage 2. Path to photos in storage is added to review 3. New review document is added in database 4. Review document ID is added to user and eatery document "reviews" array 5. Eatery document rating and relevant dining type total and count is updated 6. User is redirected to eatery profile screen 	Pass
User taps "Submit review" button and input are invalid	Error message displayed	Error message displayed	Pass

User type and taps enter for tags	Chip is created with tag text	Chip is created with tag text	Pass
User backspace tag chip	Chip is removed	Chip is removed	Pass
User wants to edit a review	Initial review data is input and shown in editable review form	Initial review data is input and shown in editable review form	Pass
User edits and saves review with valid inputs	<ol style="list-style-type: none"> Review document is updated in database Eatery document rating and relevant dining type total and count is updated User is redirected to view review screen 	<ol style="list-style-type: none"> Review document is updated in database Eatery document rating and relevant dining type total and count is updated User is redirected to view review screen 	Pass

Table 6.1.1.13 Write, Edit and Submit Review Use Case Testing

Use Case	UC016 – View review		
Test Case	Expected Output	Actual Output	Result
User taps on review	User is redirected to review screen	User is redirected to review screen	Pass
User is the review author	<ol style="list-style-type: none"> Username shown as “Me” Menu items are shown 	<ol style="list-style-type: none"> Username shown as “Me” Menu items are shown 	Pass
User is the author and review have impact	Review impact section displayed	Review impact section displayed	Pass
User is the author and review does not have any impact	Review impact section is hidden	Review impact section is hidden	Pass
User is not the review author	<ol style="list-style-type: none"> Impact chips displayed Menu items are hidden 	<ol style="list-style-type: none"> Impact chips displayed Menu items are hidden 	Pass
User taps on eatery name	User is redirected to eatery profile screen	User is redirected to eatery profile screen	Pass
User taps on “Edit review” menu item	User is redirected to edit review screen	User is redirected to edit review screen	Pass

Table 6.1.1.14 View Review Use Case Testing

Use Case	UC017 – Rate review		
Test Case	Expected Output	Actual Output	Result

User taps “Helpful” chip	1. User ID is added to review document “helpful” array 2. Chip UI updated	1. User ID is added to review document “helpful” array 2. Chip UI updated	Pass
User taps checked “Helpful” chip	1. User ID is removed from review document “helpful” array 2. Chip UI updated	1. User ID is removed from review document “helpful” array 2. Chip UI updated	Pass
User taps “Funny” chip	1. User ID is added to review document “funny” array 2. Chip UI updated	1. User ID is added to review document “funny” array 2. Chip UI updated	Pass
User taps checked “Funny” chip	1. User ID is removed from review document “funny” array 2. Chip UI updated	1. User ID is removed from review document “funny” array 2. Chip UI updated	Pass

Table 6.1.1.15 Rate Review Use Case Testing

Use Case	UC019 – Delete review		
Test Case	Expected Output	Actual Output	Result
User taps on “Delete review” menu item	Delete review dialog is displayed	Delete review dialog is displayed	Pass
User taps “Yes” button	1. Review document is deleted in database 2. Review document ID is removed from eatery and user “reviews” array 3. Eatery document rating and relevant dining type total and count is updated 4. User is redirected to previous screen before view review screen	1. Review document is deleted in database 2. Review document ID is removed from eatery and user “reviews” array 3. Eatery document rating and relevant dining type total and count is updated 4. User is redirected to previous screen before view review screen	Pass
User taps “No” button	Dialog is dismissed	Dialog is dismissed	Pass

Table 6.1.1.16 Delete Review Use Case Testing

Use Case	UC020 – View user profile		
Test Case	Expected Output	Actual Output	Result
User taps on profile picture in home screen	User is redirected to user profile screen	User is redirected to user profile screen	Pass
User taps or swipes to “My Lists” tab	User lists fragment is displayed	User lists fragment is displayed	Pass
User taps or swipes to “Reviews” tab	User reviews fragment is displayed	User reviews fragment is displayed	Pass
User taps or swipes to “Favourites” tab	User favourites fragment is displayed	User favourites fragment is displayed	Pass
User taps navigation icon	User is redirected to home screen	User is redirected to home screen	Pass

Table 6.1.1.17 View User Profile Use Case Testing

Use Case	UC021 – View user reviews		
Test Case	Expected Output	Actual Output	Result
User is viewing “Reviews” tab in user profile and user has reviews	All user reviews are displayed in RecyclerView	All user reviews are displayed in RecyclerView	Pass
User is viewing “Reviews” tab in user profile and user does not have any reviews	No reviews message displayed	No reviews message displayed	Pass
User taps on a review	User is redirected to view review screen	User is redirected to view review screen	Pass

Table 6.1.1.18 View User Reviews Use Case Testing

Use Case	UC022 – View user favourites		
Test Case	Expected Output	Actual Output	Result
User viewing “Favourites” tab in user profile and has favourites	All user favourites are displayed in RecyclerView	All user favourites are displayed in RecyclerView	Pass
User viewing “Favourites” tab in user profile and does not have favourites	No favourites message displayed	No favourites message displayed	Pass
User taps on a favourite	User is redirected to view eatery profile screen	User is redirected to view eatery profile screen	Pass

Table 6.1.1.19 View User Favourities Use Case Testing

Use Case	UC024 – View user lists		
Test Case	Expected Output	Actual Output	Result
User is viewing “My Lists” tab in user profile and user has lists	All user lists are displayed in RecyclerView	All user lists are displayed in RecyclerView	Pass
User is viewing “My Lists” tab in user profile and user does not have any lists	No lists message displayed	No lists message displayed	Pass
User taps on a list	User is redirected to view lists screen	User is redirected to view lists screen	Pass
User taps on “New list” button	User is redirected to create list screen	User is redirected to create list screen	Pass

Table 6.1.1.20 View User Lists Use Case Testing

Use Case	UC025 – Create list, UC027 – Edit list		
Test Case	Expected Output	Actual Output	Result
User taps “Save” button without any input	Error message displayed	Error message displayed	Pass
User input name exceed 50 characters	Error message displayed	Error message displayed	Pass
User input description exceed 50 characters	Error message displayed	Error message displayed	Pass
User taps “Save” button with no input for name	Error message displayed	Error message displayed	Pass
User taps “Save” button with correct input	<ol style="list-style-type: none"> 1. New list document is added to database 2. List document ID is added to user document “lists” array 3. User is redirected back to view list screen 	<ol style="list-style-type: none"> 1. New list document is added to database 2. List document ID is added to user document “lists” array 3. User is redirected back to view list screen 	Pass
User is editing list	Initial list details are input into text fields	Initial list details are input into text fields	Pass
User submits list changes and input is valid	List document is updated in database	List document is updated in database	Pass
User submits list changes and input is invalid	Error message displayed	Error message displayed	Pass

Table 6.1.1.21 Create List and Edit List Use Case Testing

Use Case	UC026 – View list		
Test Case	Expected Output	Actual Output	Result
User view list with no items	Empty list message displayed	Empty list message displayed	Pass
User view list with items	<ol style="list-style-type: none"> List items displayed in RecyclerView List size displayed in TextView 	<ol style="list-style-type: none"> List items displayed in RecyclerView List size displayed in TextView 	Pass
User taps on a list item	User is redirected to view eatery profile screen	User is redirected to view eatery profile screen	Pass
User taps on list icon on list item	<ol style="list-style-type: none"> Eatery document ID is removed from list document “eateries” array List item is removed from list 	<ol style="list-style-type: none"> Eatery document ID is removed from list document “eateries” array List item is removed from list 	Pass
User taps “Edit list” menu item	User is redirected to edit list screen	User is redirected to edit list screen	Pass
User taps “Delete list” menu item	Delete review confirmation dialog displayed	Delete review confirmation dialog displayed	Pass
User taps navigation icon	User is redirected to user profile screen	User is redirected to user profile screen	Pass

Table 6.1.1.22 View List Use Case Testing

Use Case	UC027 – Delete list		
Test Case	Expected Output	Actual Output	Result
User taps “Yes” button	<ol style="list-style-type: none"> List document is deleted in database List document ID is removed from user document “lists” array 	<ol style="list-style-type: none"> List document is deleted in database List document ID is removed from user document “lists” array 	Pass
User taps “No” button	Dialog is dismissed	Dialog is dismissed	Pass
User taps area outside dialog	Dialog is dismissed	Dialog is dismissed	Pass

Table 6.1.1.23 Delete List Use Case Testing

Use Case	UC029 – Manage account		
Test Case	Expected Output	Actual Output	Result
User taps “Manage account” menu item in user profile	User is redirected to manage account screen	User is redirected to manage account screen	Pass
User inputs new username that taken	Error message displayed	Error message displayed	Pass
User saves new username that is available	User document “username” field is updated in database	User document “username” field is updated in database	Pass
User taps “Change profile picture” text button	User is redirected to phone gallery screen	User is redirected to phone gallery screen	Pass
User makes no changes and taps on save icon	User is redirected to user profile screen	User is redirected to user profile screen	Pass
User changed profile picture and taps save icon	<ol style="list-style-type: none"> 1. User profile picture is uploaded to Cloud Storage 2. User document “picture” field is updated with path to storage file 3. Once (1) and (2) is complete, User is redirected to user profile screen 	<ol style="list-style-type: none"> 1. User profile picture is uploaded to Cloud Storage 2. User document “picture” field is updated with path to storage file 3. Once (1) and (2) is complete, User is redirected to user profile screen 	Pass
User taps close icon	User is redirected to user profile screen	User is redirected to user profile screen	Pass

Table 6.1.1.24 Manage Account Use Case Testing

Use Case	UC032 – Add photos		
Test Case	Expected Output	Actual Output	Result
User taps on add photo button	User is redirected to phone gallery screen	User is redirected to phone gallery screen	Pass
User selects photo from gallery	The selected photos displayed in ImageView	The selected photos displayed in ImageView	Pass

Table 6.1.1.25 Add Photos Use Case Testing

6.2 Implementation Issues and Challenges

During the development of this project, one of the challenges that was faced was that the AVD emulator provided by Android Studio was occasionally very glitchy despite having been cold booted or being recreated again. As a device for testing was very integral for the entire development process, it was difficult and time consuming when the emulator would suddenly stop. To deal with this issue, a physical Android device was prepared and setup for testing as a contingency plan.

Other than that, designing a smooth and aesthetically pleasing UI was relatively challenging as well. Due to the many elements that are required for most of the UIs, it was important that the end-user would be able to navigate through everything comfortably. This was especially so for developing the review form as well as the eatery profile form where many information and UI elements were being displayed in one screen. Generally, mobile phones have comparatively much less screen estate compared to computer and tablet screens. Thus, designing simplistic and minimalistic UI layouts that contained many elements was quite challenging. To improve the UI and navigation flow, Google Material UI documentation was heavily relied on for integrating UI components as well as design decisions.

Additionally, another challenge that was encountered was handling responses from synchronous methods. For instance, when getting information about places to be displayed, an API request would be made to both Places API and Firestore for information about the places and reviews and ratings data, respectively. The results returned from both API would then need to be combined and be displayed at the same time. For different methods in the application, both synchronous and asynchronous methods were necessary and thus, difficult to design the code structure around. Eventually, this was issue was also handled by implementing response listeners using Java interfaces as well as Cloud Firestore transactions.

6.3 Objectives Evaluation

The application developed in this project is evaluated based on the project objectives as follows:

Objective	Evaluation	Conclusion
To create a reliable and easy-to-use Android mobile application for F&B reviews that can aid users in deciding where and what to eat.	This objective is met as users can use the application to create reviews that have various input options but only the dining type and ratings which are compulsory inputs. The review form allows users to create informative reviews easily. The application has also passed the use case testing affirming its reliability in terms of being able to perform as a review application.	Achieved
To develop a platform where users can browse and discover the F&B of Kampar, Perak	This objective is met as the application provides users with suggestions of F&B in Kampar. Users can use the application to browse eateries in different areas of Kampar.	Achieved
To provide users with a centralized COP for the F&B in Kampar, Perak	This objective is met as the application developed is a COP that offers browsing features specially for Kampar.	Achieved

Table 6.3.1 Objectives Evaluation

CHAPTER 7 CONCLUSION AND RECOMMENDATION

7.1 Conclusion

EatMore is an Android mobile application for F&B reviews that aims to help users decide where and what to eat. EatMore differs from other COP applications where users can rate a place's cleanliness, F&B and customer service as well as specifying the dining type for the review. This is incorporated in the review form which provides users with both a simple and fast process for creating a review and providing others with reviews that are both informative and a quick read. This application stands out as users would be able to have a better idea of specific major aspects of a place through the reviews at a simple glance, namely in terms of its ratings for dine in or takeaway as well as cleanliness, F&B, and customer service. Overall, EatMore serves as a simple solution to a complex problem.

To summarize, this application was developed in Java using Android Studio and relies on Firebase as well as Google Places API to perform its features and functions. Furthermore, the project was successfully carried out in which the developed application was able to achieve the project's main and sub-objectives as discussed in the previous chapter.

7.2 Future Work

Although the application as it is now has achieved the objectives and purpose of the project, there are several improvements that can be made and is suggested for future works. Generally, additional work is recommended to be made to improve user engagement for the application. For instance, integrating gamification features into the application's review module would encourage better user engagement to the platform. This is very beneficial for the application as its role as a COP requires user contribution to really help the platform flourish. Besides that, data mining solutions could be integrated into application to improve the suggestions for users based on their favourites as well as lists content. This would be both beneficial to the user and the platform as it would improve the user experience as well as improve the platform's reliability in terms of helping users decide where and what to eat, which is the application's core aim.

REFERENCES

- [1] B. Schwartz, *The Paradox of Choice*. Harper Perennial.
- [2] N. Ali and M. A. Abdullah, "The food consumption and eating behaviour of Malaysian urbanites: Issues and concerns," *Geografia: Malaysian journal of society and space*, vol. 8, no. 6, pp. 157–165, 2012.
- [3] A. Hussey. "Food Delivery in Demand During COVID-19." Kerry. <https://www.kerry.com/insights/kerrydigest/2020/food-delivery-in-demand-during-covid-19> (Accessed Nov. 29, 2020).
- [4] I. Loh. "Covid-19: About 40 Kampar businesses temporarily close to 'Stop the Chain' of infection," *TheStar*, Apr. 19, 2021. [Online]. Available: <https://www.thestar.com.my/news/nation/2021/04/19/covid-19-about-40-kampar-businesses-temporarily-close-to-039stop-the-chain039-of-infection>. (Accessed Mar. 3, 2022).
- [5] "How Online Reviews Influence Sales: Evidence of the Power of Online Reviews to Shape Customer Behavior," Medill Spiegel Research Center. Accessed: Nov 29, 2020. [Online]. Available: https://spiegel.medill.northwestern.edu/wp-content/uploads/sites/2/2021/04/Spiegel_Online-Review_eBook_Jun2017_FINAL.pdf
- [6] J. Anderson. "Why Are Customer Reviews So Important?" Medium. <https://medium.com/revain/why-are-customer-reviews-so-important-185b915d4e5d#:~:text=Reviews%20not%20only%20have%20the,to%20improve d%2> (Accessed Nov. 2, 2020).
- [7] P. Liu and E. C.-Y. Tse, "Exploring factors on customers' restaurant choice: an analysis of restaurant attributes," *British Food Journal*, vol. 120, no. 10, pp. 2289–2303, Sep. 2018, doi: 10.1108/BFJ-10-2017-0561.
- [8] Yelp. Version 22.15.0-26221516. Yelp Inc. Accessed: Mar. 4, 2022. [Mobile app]. Available: <https://play.google.com/store/apps/details?id=com.yelp.android&hl=en>
- [9] D. Johnson. "Yelp Tips vs. Reviews: Know the Difference." 33mileradius.com. <https://www.33mileradius.com/reviews/yelp-tips-vs-reviews-know-difference/#:~:text=about%20Yelp%20Tips%3F-,Yelp%20Tips,Reviews&text=While%20a%20tip%20is%20generally,have%20ha d%> (Accessed Apr. 4, 2022).

- [10] Burpple. Version 7.3.26.4457. Burpple Pte Ltd. Accessed: Mar. 4, 2022. [Mobile app]. Available: <https://play.google.com/store/apps/details?id=com.burpple.app&hl=en&gl=US>
- [11] Zomato. Version 16.4.6. Zomato Media Pvt Ltd. Accessed: Mar. 5, 2022. [Mobile app]. Available: <https://play.google.com/store/apps/details?id=com.application.zomato&hl=en&gl=US>
- [12] S. Sen. “How Foodiebay became Zomato,” BusinessToday.In, May 23, 2012. [Online]. Available: <https://www.businesstoday.in/latest/corporate/story/foodiebay-zomato-30399-2012-05-23>. (Accessed Dec. 1, 2021).
- [13] A. Priyadarshi. “Reviews 2.0 – tags are a new content currency.” zomato.com. <http://zomato.com/blog/reviews-2-0> (Accessed Sep. 20, 2019).
- [14] M. Breja. “Software Engineering | Prototyping Model.” GeeksforGeeks.org. <https://www.geeksforgeeks.org/software-engineering-prototyping-model/> (Accessed Apr. 6, 2022).
- [15] M. Martin. “Prototyping Model in Software Engineering: Methodology, Process, Approach.” Guru99.com. <https://www.guru99.com/software-engineering-prototyping-model.html#5> (Accessed Apr. 6, 2022).
- [16] “Overview.” Google Maps Platform. <https://developers.google.com/maps/documentation/places/web-service/overview> (Accessed July 29, 2021).

APPENDICES

APPENDIX A: WEEKLY REPORTS

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T1	Study week no.: 1
Student Name & ID: Yong Ju Wei, 18ACB04479	
Supervisor: Ts Saw Seow Hui	
Project Title: Mobile Application for F&B Reviews	

1. WORK DONE

Review and revise FYP1 materials and prototype

2. WORK TO BE DONE

Refine project plans and system designs

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T1	Study week no.: 3
Student Name & ID: Yong Ju Wei, 18ACB04479	
Supervisor: Ts Saw Seow Hui	
Project Title: Mobile Application for F&B Reviews	

1. WORK DONE

Plan project timeline, refine systems designs and revise project plans
Edited project report based on FYP2 report guidelines

2. WORK TO BE DONE

Research relevant APIs and SDKs for application functions
Begin development process

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T1	Study week no.: 5
Student Name & ID: Yong Ju Wei, 18ACB04479	
Supervisor: Ts Saw Seow Hui	
Project Title: Mobile Application for F&B Reviews	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Research and test APIs related to application function
Set up project environment for development and implementation
Prepare Android project environment for integration with Firebase and Google Maps API and Android SDK
Refactor FYP1 prototype and developed some basic UI for a few activities
Meet with supervisor to update on current progress

2. WORK TO BE DONE

Continue development process

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T1	Study week no.: 7
Student Name & ID: Yong Ju Wei, 18ACB04479	
Supervisor: Ts Saw Seow Hui	
Project Title: Mobile Application for F&B Reviews	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Refactored and completed user authentication and user location modules
Begun developing eatery profile and user profile flows and functions

2. WORK TO BE DONE

Update documentation and continue developing and refactoring code

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T1	Study week no.: 9
Student Name & ID: Yong Ju Wei, 18ACB04479	
Supervisor: Ts Saw Seow Hui	
Project Title: Mobile Application for F&B Reviews	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Meet with supervisor to update on progress and discuss project direction
Completed user profile module

2. WORK TO BE DONE

Complete all application modules
Update project report

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T1	Study week no.: 11
Student Name & ID: Yong Ju Wei, 18ACB04479	
Supervisor: Ts Saw Seow Hui	
Project Title: Mobile Application for F&B Reviews	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Complete review, and eatery module

2. WORK TO BE DONE

Complete and refactor application modules and perform testing
Finalize and proof-read project chapters 5 and 6

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T1	Study week no.: 13
Student Name & ID: Yong Ju Wei, 18ACB04479	
Supervisor: Ts Saw Seow Hui	
Project Title: Mobile Application for F&B Reviews	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Finalize application development and testing
Review and finalize project report for submission
Discuss and review final project report and mobile application with supervisor

2. WORK TO BE DONE

Prepare presentation materials and produce application demo video
Deliver presentation

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS



Supervisor's signature



Student's signature

POSTER

EATMORE

Mobile Application for F&B Reviews



INTRODUCTION

Having trouble deciding where and what to eat? EatMore lets you search, review and discover eateries near you! Browse the different categories and find out more about the delicious F&B in Kampar.

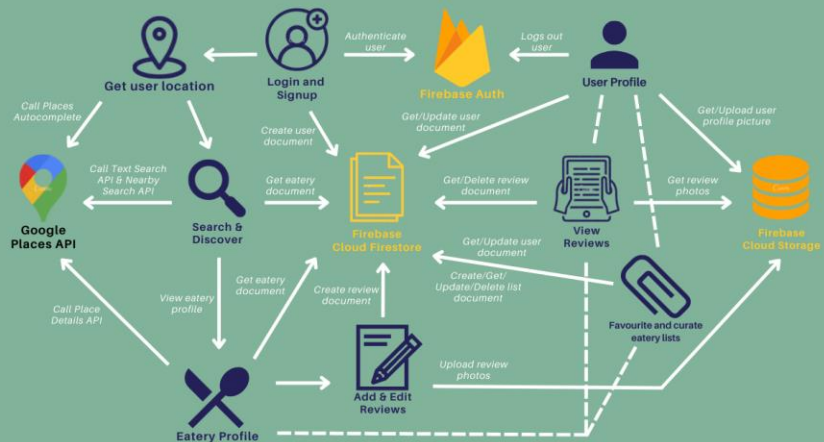


MOTIVATION

Given many choices and options, anyone would be stumped on deciding which can both satisfy needs and maximise value, especially when it comes to F&B. The countless platforms where reviews about eateries are spread out doesn't really help either.

OBJECTIVES

- 1 To develop a reliable and easy-to-use Android mobile app for F&B reviews
- 2 To develop a platform where users can browse and discover the F&B in Kampar, Perak
- 3 To provide users with a centralized COP for the F&B in Kampar, Perak.



METHODS

- Developed using Java on Android Studio
- Adopt extreme programming (XP) methodology
- Data storage using Cloud Firestore (NoSQL database)
- Place details retrieved from Google Places API
- Core API functions supported by Google Cloud, Google Maps SDK and Google Play services

DEVELOPED BY YONG JU WEI | SUPERVISED BY TS SAW SEOW HUI

PLAGIARISM CHECK RESULT

FYP2 Turnitin Plagiarism Check

ORIGINALITY REPORT

4 %	4 %	1 %	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	oraserv.cs.siena.edu Internet Source	1 %
2	hdl.handle.net Internet Source	1 %
3	www.lenovo.com Internet Source	<1 %
4	neilpatel.com Internet Source	<1 %
5	www.ncbi.nlm.nih.gov Internet Source	<1 %
6	www.stitchdata.com Internet Source	<1 %
7	docs.learnupon.com Internet Source	<1 %
8	docs.oracle.com Internet Source	<1 %
9	tajemniczytrojkat.pl Internet Source	<1 %

10	www.researchgate.net Internet Source	<1 %
11	Ahmed Tlili, Fathi Essalmi, Mohamed Jemni, Kinshuk, Nian-Shing Chen. "A Complete Validated Learning Analytics Framework", International Journal of Information and Communication Technology Education, 2018 Publication	<1 %
12	dokumen.pub Internet Source	<1 %
13	firebase.google.com Internet Source	<1 %
14	shodhganga.inflibnet.ac.in Internet Source	<1 %
15	www.hindawi.com Internet Source	<1 %
16	Wervyan Shalannanda, Wilson Rustiandy, Radifa Akbar Abhesa, Dimas Aji Pangestu et al. "Application for Rural Internet Access Services Logistics Travel Duration in Indonesia", 2020 6th International Conference on Wireless and Telematics (ICWT), 2020 Publication	<1 %
17	docplayer.net Internet Source	<1 %
	venturebeat.com	

18	Internet Source	<1 %
19	Armando Padilla. "Beginning Zend Framework", Springer Science and Business Media LLC, 2009 Publication	<1 %
20	Francesco Ricci, Quang Nhat Nguyen, Olga Averjanova. "chapter 5 Exploiting a Map-Based Interface in Conversational Recommender Systems for Mobile Travelers", IGI Global, 2010 Publication	<1 %
21	Ying Bai. "Practical Database Programming with Visual Basic.NET", Wiley, 2012 Publication	<1 %
22	github.com Internet Source	<1 %
23	vector.fnal.gov Internet Source	<1 %

Exclude quotes On

Exclude matches < 8 words

Exclude bibliography On

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



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Yong Ju Wei
ID Number(s)	18ACB04479
Programme / Course	IA
Title of Final Year Project	Mobile Application for F&B Reviews

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: <u> 4 </u> % Similarity by source Internet Sources: <u> 4 </u> % Publications: <u> 1 </u> % Student Papers: <u> 0 </u> %	
Number of individual sources listed of more than 3% similarity: <u> 1 </u>	
Parameters of originality required and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

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

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

Signature of Supervisor

Name: _____
 Ts. SAW SEOW HUI

Date: _____
 4/22/2022

Signature of Co-Supervisor

Name: _____

Date: _____



UNIVERSITI TUNKU ABDUL RAHMAN

**FACULTY OF INFORMATION & COMMUNICATION
TECHNOLOGY (KAMPAR CAMPUS)
CHECKLIST FOR FYP2 THESIS SUBMISSION**

Student Id	18ACB04479
Student Name	Yong Ju Wei
Supervisor Name	Ts Saw Seow Hui

TICK (√)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

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

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

(Signature of Student)

Date: 21 April 2022