

UMP ITEM TRADING APPLICATION
IN ANDROID MOBILE APPLICATION

LOH HIN YE

BACHELOR OF COMPUTER SCIENCE
(SOFTWARE ENGINEERING) WITH
HONOURS

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : _____ LOH HIN YE _____

Date of Birth : _____

Title : _____ UMP ITEM TRADE APPLICATION _____

Academic Session : _____ Semester 1 2022/2023 _____

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

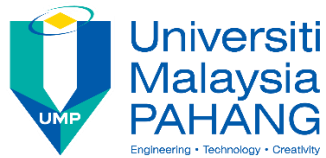
(Student's Signature)

(Supervisor's Signature)

New IC/Passport
Number Date: 31/1/2023

DR.ZAFRIL RIZAL BIN M AZMI
Name of Supervisor
Date: 31/1/2023

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.



SUPERVISOR'S DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of Computer Science (SOFTWARE ENGINEERING) with Honors

A handwritten signature in black ink, appearing to read 'Zafirul Rizal Bin M Azmi', is written above a horizontal line.

(Supervisor's Signature)

Full Name : DR. ZAFRIL RIZAL BIN M AZMI
Position :
Date : 31/1/2023

(Co-supervisor's Signature)

Full Name :
Position :
Date :



STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institution.

A handwritten signature in black ink, appearing to read 'LOH HIN YE', is written over a horizontal line.

(Student's Signature)

Full Name : LOH HIN YE

ID Number : CB19039

Date : 3 JUNE 2022

UMP Item Trading Application

LOH HIN YE

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Computer Science (SOFTWARE ENGINEERING) with Honors

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG

JAN 2023

ACKNOWLEDGEMENTS

First, I would like to say thank you to Dr. Zafril Rizal Bin M Azmi who accept me to become my Final Year Project's supervisor. Dr. Zafril Rizal Bin M Azmi give me a lot of guidance and always give me feedback during the full development process of my Final Year Project entitled "UMP Item Trading Application".

Next, I would like to thanks to my parent who always support me in physical and mental. They bought me a new laptop immediately without any consider when my laptop is malfunction and having damage problem so I can complete my Final Year Project on time.

Last but not least, I also want to thanks to all my dear friends and my roommate who always give me advice, motivations and positive energy to me while the development is ongoing.

ABSTRAK

Aplikasi mudah alih ialah program yang berjalan pada peranti mudah alih, seperti telefon pintar atau tablet. Walaupun pada hakikatnya aplikasi selalunya merupakan unit perisian kecil dengan keupayaan terhad, ia masih boleh memberikan perkhidmatan dan pengalaman yang hebat kepada pengguna. Beli-belah mudah alih membolehkan pengguna membeli item daripada peranti mudah alih dengan menggunakan aplikasi seperti Shopee, Lazada dan sebagainya. Ia menjadikan proses pembelian atau penjualan produk dipermudahkan. Dengan mengurangkan pembaziran sumber, di dalam UMP terdapat ramai orang mempunyai banyak buku terpakai atau beberapa barangan terpakai yang boleh dipindah milik dan ingin dijual kepada junior untuk mengurangkan pembaziran sumber, dan junior juga boleh menjimatkan wang berdasarkan hanya menggunakan barang terpakai dan tidak perlu membeli barang baru. Semua penjual hanya boleh mengiklankan barangan atau produk mereka di media sosial atau sembang kumpulan belajar seperti saluran telegram, kumpulan WhatsApp dan sebagainya buat masa ini. Iklan di media sosial atau sembang kumpulan belajar akan mengganggu dan mengurangkan pengalaman pembelajaran pelajar. Objektif kajian ini adalah untuk mengkaji aplikasi mudah alih sedia ada berkaitan perdagangan dan reka bentuk mudah alih serta membangunkan aplikasi sebagai platform untuk pelajar ump membeli dan menjual barangan menggunakan Android Studio. Ia juga untuk menilai kefungsiannya aplikasi mudah alih yang dibangunkan yang mampu pelajar berdagang barang. Skop pengguna kajian ini adalah untuk pelajar sarjana muda UMP dan di kawasan Pekan. Selain itu, skop sistem adalah untuk meliputi transaksi antara penjual dan pembeli. Bagi skop pembangunan, kajian ini mengandungi elemen multimedia seperti grafik dan teks serta menggunakan Android Studio untuk membangunkan aplikasi. Makna projek ini ialah pelajar boleh mempunyai platform untuk membuat transaksi barangan dengan pelajar lain. Ia juga boleh memberi peluang kepada pelajar yang perlu memulakan perniagaan di platform ini dan untuk pembeli, ia juga boleh menjadi lebih mudah dan mempunyai harga yang lebih murah untuk barangan. Bagi UMP, ia boleh mempunyai satu platform baharu yang dapat membantu pelajar UMP meningkatkan kehidupan seharian dan menambah baik pengalaman di UMP.

ABSTRACT

A mobile application is a programme that runs on a mobile device, such as a smartphone or tablet. Despite the fact that applications are often small software units with limited capabilities, they can still deliver great service and experience to users. Mobile shopping allows users to purchase items from mobile devices by using an application like Shopee, Lazada, and so on. It's made the process of purchasing or selling products be simplified. With reducing the waste of resources, inside UMP there have a lot of people has many used books or some used items that are transferable and want to sell to juniors to reduce the waste of resources, and juniors also can save money based on just using the used items and no need to purchase new items. All the sellers are only can advertise their items or products on social media or study group chat like telegram channel, WhatsApp group, and so on for now. Advertisement on social media or the study group chat will disturb and reduce students' learning experience. The objective of this project is to study the existing mobile application related to mobile commerce and design and develop an application as a platform for ump students to purchase and sell items using Android Studio. It also to evaluate the functionality of the developed mobile applications that are able students to trade items. The user scope of this study is for the undergraduate UMP students and in Pekan area. Besides, the system scope is to cover transactions between seller and buyer. For the development scope, the study is contains multimedia elements such as graphics and text and using the Android Studio to develop the application. The signification of the project is students can have a platform to make transactions of items with other students. It also can give an opportunity to the students who need to start a business on this platform and for the buyers, it also can be more convenient and have a cheaper price on the items. For UMP, it can have a new platform that can help UMP students to improve their daily life and improve their experience in UMP.

TABLE OF CONTENT

DECLARATION	
TITLE PAGE	
ACKNOWLEDGEMENTS	ii
ABSTRAK	iii
ABSTRACT	iv
TABLE OF CONTENT	v
LIST OF TABLES	viii
LIST OF FIGURES	x
LIST OF SYMBOLS	xiii
LIST OF ABBREVIATIONS	xiv
CHAPTER 1 INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Statements	2
1.3 Objectives	3
1.4 Scope	4
1.5 Significance of Project	4
1.6 Report Organization	4
CHAPTER 2 LITERATURE REVIEW	6
2.1 Introduction	6
2.2 Review of Existing System	6
2.2.1 Application I – Mudah.my	6
2.2.2 Application II – Carausell	10
2.2.3 Application III – Lelong.my	13

2.3	Comparison of Existing System	18
2.4	Summary of Review Existing System	24
CHAPTER 3 METHODOLOGY		26
3.1	Introduction	26
3.2	Methodology	27
	3.2.1 Rapid Application Development (RAD)	27
	3.2.2 Model-View-Presenter (MVP)	29
3.3	Project Requirement	31
	3.3.1 Functional Requirement	31
	3.3.2 Non-functional Requirement	32
	3.3.3 Constraints	34
	3.3.4 Limitations	35
3.4	Proposed Design	36
	3.4.1 Context Diagram	36
	3.4.2 Use Case Diagram and Description	37
	3.4.3 Activity Diagram	53
3.5	Data Design	58
	3.5.1 Entity Relationship Diagram (ERD)	58
	3.5.2 Data Dictionary	58
3.6	Proof of Initial Concept	61
	3.6.1 Manage Registration	61
	3.6.2 Manage Profile	63
	3.6.3 Manage Goods	66
	3.6.4 Manage Order	71
	3.6.5 Manage User	75
	3.6.6 Storyboard	77
3.7	Testing Plan	77
3.8	Potential Use of The Proposed Solution	80
CHAPTER 4		81
4.1	RESULT AND DISCUSSION	81

4.2	Introduction	81
4.3	Result	81
4.3.1	Installation of IDE (Android Studio)	81
4.3.2	Implementation of database	84
4.3.3	Implementation of Application	85
4.3.4	Implementation of Integrated Development Environment (IDE)	102
4.4	Discussion	109
4.5	Chapter Summary	138
CHAPTER 5		139
CONCLUSION		139
5.1	Objective Revisited	139
5.2	Limitation	139
5.3	Future Work	140
REFERENCES		141
APPENDIX A SRS		143

LIST OF TABLES

Table 2.1	Comparison of the existing system	18
Table 2.2	Function Comparison	22
Table 3.1	RAD Methodology Phases	27
Table 3.2	MVP architectural pattern role and responsibility	30
Table 3.3	Non-functional Requirement	32
Table 3.4	Constraints	34
Table 3.5	Limitation	35
Table 3.6	Manage Registration	39
Table 3.7	Manage Profile	42
Table 3.8	Manage Goods	44
Table 3.9	Manage Order	48
Table 3.10	Manage User	51
Table 3.11	Data Dictionary-User	58
Table 3.12	Data Dictionary-Item	59
Table 3.13	Data Dictionary-Watchlist	60
Table 3.14	UAT form	77
Table 4.1	Test Case Login Page	110
Table 4.2	Test Case Register Page	112
Table 4.3	Test Case Forget Password Page	116
Table 4.4	Test Case Main Page	117
Table 4.5	Test Case Sell Option Page	119
Table 4.6	Test Case Upload Item Page	120
Table 4.7	Test Case Item Page	122
Table 4.8	Test Case Edit Item Page	123

Table 4.9	Test Case Donate Item Page	125
Table 4.10	Test Case Watchlist Page	126
Table 4.11	Test Case Profile Page	127
Table 4.12	Test Case Edit Profile Page	128
Table 4.13	Test Case Search Page	130
Table 4.14	Test Case Item Detail Page	132
Table 4.15	Test Case Admin Page	133
Table 4.16	Test Case Edit User Page	133
Table 4.17	Test Case Add New User Page	134

LIST OF FIGURES

Figure 1.1	Bicycle piles	3
Figure 2.1	Logo of Mudah.my	7
Figure 2.2	Interface of Mudah.my	8
Figure 2.3	Searching item interface	9
Figure 2.4	Interface of the item description	10
Figure 2.5	Logo of Carousell	11
Figure 2.6	Login Interface	12
Figure 2.7	Main Interface	13
Figure 2.8	Logo of Lelong.my	14
Figure 2.9	Main page of Lelong.my	15
Figure 2.10	Products detail interface	16
Figure 2.11	Price Listed of Lelong.my WebStore	17
Figure 3.1	Rapid Application Development (RAD) Methodology	27
Figure 3.2	Model-View-Presenter Model	29
Figure 3.3	Context Diagram	36
Figure 3.4	Use Case Diagram	38
Figure 3.5	Use Case Diagram of Registration	39
Figure 3.6	Use Case Diagram of Manage Profile	42
Figure 3.7	Use Case Diagram of Manage Goods	44
Figure 3.8	Use Case Diagram of Manage Order	48
Figure 3.9	Use Case Diagram of Manage User	51
Figure 3.10	Activity Diagram of Manage Registration	53
Figure 3.11	Activity Diagram of Manage Profile	54
Figure 3.12	Activity Diagram of Manage Goods	55

Figure 3.13	Activity Diagram of Manage Order	56
Figure 3.14	Activity Diagram of Manage User	57
Figure 3.15	Entity Relationship Diagram (ERD)	58
Figure 3.16	Login interface	61
Figure 3.17	Forgot Password interface	
Figure 3.18	Register interface	62
Figure 3.19	Main interface	
Figure 3.20	Profile interface	65
Figure 3.21	Edit Profile interface	
Figure 3.22	Sell Option interface	68
Figure 3.23	Upload Item interface	
Figure 3.24	Item interface	69
Figure 3.25	Edit item interface	
Figure 3.26	Donate Page	70
Figure 3.27	Search interface	73
Figure 3.28	Item Detail interface	
Figure 3.29	Watch List interface	74
Figure 3.30	Admin Page Interface	75
Figure 3.31	Add new user Interface	76
Figure 3.32	Edit user interface	
Figure 3.33	Storyboard	77
Figure 4.1	Android Studio IDE	82
Figure 4.2	Android Studio	83
Figure 4.3	Firebase – Realtime Database	84
Figure 4.4	Firebase – Authentication Database	84
Figure 4.5	Firebase – Storage Database	85
Figure 4.6	Login Interface	86
Figure 4.7	Forgot Password Interface	87
Figure 4.8	Register Interface	88
Figure 4.9	Main Interface	89
Figure 4.10	User Profile Interface	90
Figure 4.11	Edit Profile Interface	91
Figure 4.12	Seller Option Interface	92

Figure 4.13	Sell Page Interface	93
Figure 4.14	Donate Item Interface	94
Figure 4.15	User's Item Interface	95
Figure 4.16	Edit Item Interface	96
Figure 4.17	Searching Interface	97
Figure 4.18	Item Detail Interface	98
Figure 4.19	Watch List Interface	99
Figure 4.20	Admin Page Interface	100
Figure 4.21	Add New User Interface	101
Figure 4.22	Edit User Page	102
Figure 4.23	Item Adapter	103
Figure 4.24	Watchlist Adapter	105
Figure 4.25	User Adapter	106
Figure 4.26	Calling function	107
Figure 4.27	SMS function	108
Figure 4.28	email function	109

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

ERD	Entity Relationship Diagram
ITA	Item Trading Application
SDD	Software Design Document
SRS	Software Requirement Specification
UAT	User Acceptance Test
UMP	Universiti Malaysia Pahang

CHAPTER 1

INTRODUCTION

1.1 Introduction

A mobile application is a software programme that runs on a mobile device like a smartphone or tablet. Even though applications are often little software units with restricted capabilities, they can still provide excellent service and experience to users. There have two categories in the application which are native and web applications. Besides that, it also has some types of application which is gaming applications, productivity applications, lifestyle and entertainment applications, and mobile commerce (m-commerce).

The use of m-commerce activities is increasing. According to market research firm Statista, m-commerce sales in the U.S. were estimated at \$207.2 billion in 2017. M-commerce provides users to purchase or sell items or services through wireless handheld devices like smartphones and tablets. Compare with e-commerce, m-commerce enables users to access the online shopping platform for trading propose without requiring a desktop computer or a laptop computer to access it. It was a lot of industries that were affected by m-commerce including financial services, telecommunications, service, and retail and the most influential by m-commerce is information services.

Mobile shopping allows users to purchase items from mobile devices by using an application like Shopee, Lazada, and so on. It's made the process of purchasing or selling products be simplified. Application commerce, which is a transaction that takes place over a native application, is a subcategory of mobile shopping. In this project, a mobile application is for trading proposals, it can greatly reduce the waste of resources and save money for the juniors where the juniors can purchase a transferable item that is used or

something that is not used by seniors and the project also allow students to sell or purchase freely in the application. This mobile application is open for the student of UMP to upload their items for sale and allow students UMP to purchase. Therefore, the student can easily trade items between UMPians.

1.2 Problem Statements

With reducing the waste of resources, inside UMP there have a lot of people has many used books or some used items that are transferable and want to sell to juniors to reduce the waste of resources, and juniors also can save money based on just using the used items and no need to purchase new items. All the sellers are only can advertise their items or products on social media or study group chat like telegram channel, WhatsApp group, and so on for now. Advertisement on social media or the study group chat will disturb and reduce students' learning experience. Because of long-term advertisements were appearing on these social media or study groups will make a worse experience for users who do not need these products or items. With this project, all the sellers can target the customer that really needs their items or products.

There are currently 6.4 billion smartphone users on the planet (Statista, 2021). With a global population of 7.9 billion people, this amounts to a smartphone penetration rate of over 80%. More than half of internet users (55.4%) use their mobile phones to shop for things online, and seven out of ten (69.4%) say they use shopping apps on their smartphones or tablets (DataReportal, 2021). It's important to solve because it will make our UMP students purchase or sell items easily to other UMP students. It will make our UMP student's life be more convenient and it also can reduce the waste of resources. It can greatly reduce the waste of resources and save the juniors money because they do not need them to purchase new items.

Nowadays, UMP students are forced to purchase new items although a lot of students still can accept using second-hand or used items temporarily. In commerce, sellers are able to upload their items for sale, users are able to search items, purchase

items, filter items, and sort items. Those functions are the common functions that can get in many m-commerce. Users can easily use those functions to have a great user experience in this project. It's important to ensure your consumers' mobile shopping experience is seamless. With this mobile application, students can easily purchase used items from other students to make the student's life be more convenient.



Figure 1.1 Bicycle piles

Figure 1.1 shows the bicycle piles which is one of the resources that are wasted in UMP. Inside UMP, there have a lot of bicycles was set aside in Gambang area. Some of the senior was graduated and don't want to take the bicycles back to their hometown or their new work place. Therefore, the application can solve this problem which able junior to find the bicycles that senior want to sell or the senior want to donate.

1.3 Objectives

- I. To study the existing mobile applications related to mobile commerce.
- II. To design and develop an application as a platform for ump students to purchase and sell items using Android Studio.
- III. To evaluate the functionality of the developed mobile applications that are able students to trade items.

1.4 Scope

- User Scope:
 - I. Undergraduate UMP students.
 - II. Students in the Pekan area.
- System Scope:
 - I. Covered transactions between seller and buyer
- Development Scope:
 - I. Contains multimedia elements such as graphics and text
 - II. Using Android Studio

1.5 Significance of Project

- I. UMP Students
 - Students can have a platform to make transactions of items with other students. It also can give an opportunity to the students who need to start a business on this platform and for the buyers, it also can be more convenient and have a cheaper price on the items.
- II. University Malaysia Pahang
 - For UMP, it can have a new platform that can help UMP students to improve their daily life and improve their experience in UMP.

1.6 Report Organization

The thesis is divided into five chapters: an introduction, a literature review, a methodology, a result and discussion, and a conclusion. Each chapter consists of their respective criteria to be explained for the further understanding of the development of the project.

Chapter one explained the introduction to the project, which are the problem statements, the objectives, the scope and significance of the project as well as the thesis organization. This chapter also consists of three relevant objectives, a categorization of the scope involves in this project, and the thesis organization that explains the content of each chapter.

Chapter two discusses the literature review of three existing mobile applications with descriptions and comparisons of the mobile application in m-commerce. Three related works will be analyzed for their functions and their strengths and weaknesses will be compared to see if there is any enhancement based on the analysis.

Chapter three discusses the methodology used by this project. This chapter will describe the methodology and requirements of the project. Use-case diagram, use-case description, activity diagram, entity relationship diagram, data dictionary, proof of initial concept, storyboard, and testing plan of the system will be defined in this chapter.

Chapter four discusses of the development and testing of the functionalities of the proposed android application. In this chapter, it will discuss the implementation of database, implementation of application and the code of the project. Lastly, the User Acceptance Test (UAT) report has also been stated in this chapter.

Chapter five discusses the conclusion of the project which content the objective revisited, limitation and the future work of the project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Chapter 2 is about the review of the existing application in m-commerce. Three existing applications for trading items were explained in detail and compared. The comparisons will focus on the Graphical User Interface (GUI), Operating System (OS), size of the application, functions provided, language provided, type of connection, and the advantages and disadvantages of the mobile application. Comparisons between the three existing mobile applications recommend the mobile application's weaknesses, strengths, and effectiveness. The comparison is done to ensure that the mobile application that will be developed in this project is a better version.

2.2 Review of Existing System

There are several m-commerce applications in the market. In this section, three existing mobile applications in m-commerce will be reviewed. These three applications are Mudah.my, Carousell, Lelong.my.

2.2.1 Application I – Mudah.my

Mudah.my is an application that able users to sell and buy second-hand goods. It is a mobile application that is available on Google Play Store and App Store and it also provided a website for users. Users can install the application for free for both operating systems which are android in Google Play Store and iOS in App Store. Figure 2.1 shows the logo of Mudah. my. Mudah. my was founded in 2007 as a joint venture between

Singapore Press Holdings Ltd and Schibsted ASA called Mudah. my Sdn. Bhd. (previously 701 Search Sdn. Bhd.). The brand name Mudah. my is derived from the Malay word "Mudah," which means "simple" or "convenient".

In this application, users are required to log in to use the main function which is to sell and buy items. Users can sign up for an account or sign in to the existed account via Facebook, Google, or email. Some sub-functions like searching items, contact sellers, filtering by categories, filtering by prices, and so on can be used normally without signing in to the account. Figure 2.2 shows the interface of the main page of the application. In this interface, it shows the button that able users to sell and search for items in the application. Figure 2.3 shows the searching interface and Figure 2.4 shows the interface of the item description. Figure 2.4 also able users contact to sellers. After users log in to an existed account, the application will allow users to sell items and users can set the listed items as a favourite. Users are also required to log in to the existing account to use the chatting function. This application lets users contact sellers via email, call, and text.



Figure 2.1 Logo of Mudah.my

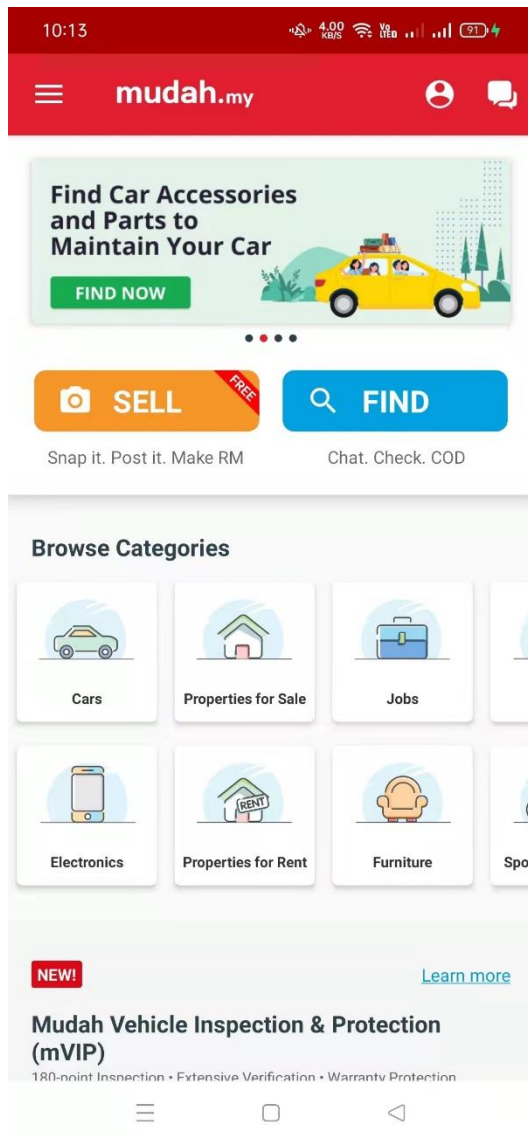


Figure 2.2 Interface of Mudah.my

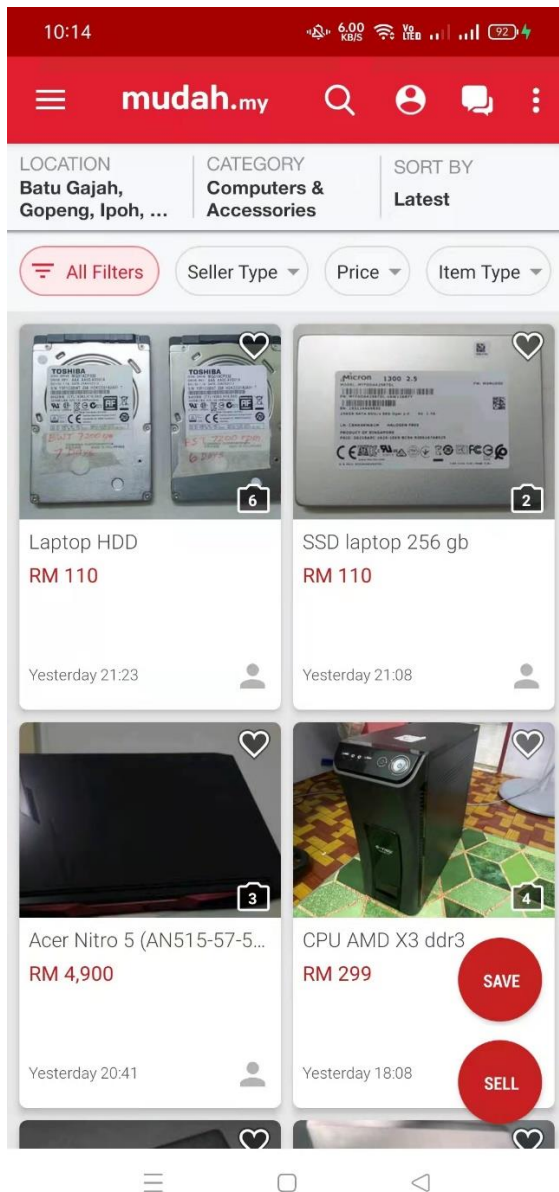


Figure 2.3 Searching item interface

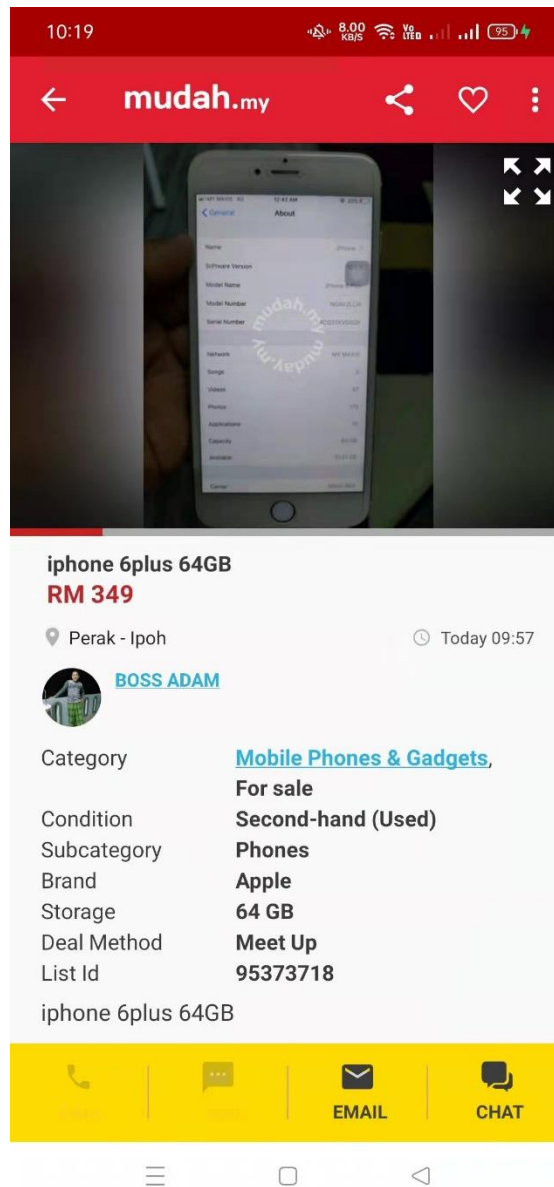


Figure 2.4 Interface of the item description

2.2.2 Application II – Carausell

Carousell is a C2C (consumer-to-consumer) and B2C (business-to-consumer) marketplace in Singapore where you can buy and sell new and used goods. It is a smartphone application that can be found on the Google Play Store and the App Store, as well as a website for users. Users can install the application for free for both operating systems which are android in Google Play Store and iOS in App Store. Quek Siu Rui,

Lucas Ngoo, and Marcus Tan co-founded Carousell in Singapore on May 14, 2012. Figure 2.5 shows the logo of Carousell.

In this application, users are required to log in to the registered account via Facebook, Google, or email, as shown in Figure 2.6. After login, users can use the function in Carousell which like searching items, contacting sellers, making offers, filtering functions, and so on. Figure 2.7 shows the interface of the main page of the application. In this interface, users can search items by using the search box at the top of the interface and it also provided a group function as a social media platform that able sellers to promote their products and able buyer to learn more about the product. In this application, buyers can argue the price with the sellers. Carousell also allows users to set the listed items as a favourite to make the users easily compare prices between the items. Carousell also provided an activities function that will promote the activities or promotions that are issued by Carousell like free shipping promotion with T&C.



Figure 2.5 Logo of Carousell


Login




Sell and declutter

Make money, while freeing up space



 Continue with Google

Sign up for an account

 Continue with Facebook

By signing up, you agree to our [Terms of Service](#) and [Privacy Policy](#)



Figure 2.6 Login Interface



Figure 2.7 Main Interface

2.2.3 Application III – Lelong.my

Lelong.my is a Malaysian-based online marketplace that offers buyers a comprehensive and economical e-commerce platform with over 10,000 sellers offering a diverse range of products and services. It is a mobile application that is available on Google Play Store and App Store and it also provided a website for users. Figure 2.8 shows the logo of the Lelong.my and users can install the application for free for both operating systems which are android in Google Play Store and iOS in App Store. Autoworld's web development was started in 1998 by K.S. Wei and Richard Tan, owners of eBay's Malaysian auction site Lelong.com.my.

In this application, users are required to use the main function which is to buy items. Users can sign up for an account or sign in to the existed account via Facebook, Google, Apple, or email. Some sub-functions like searching items, contact sellers, filtering by categories, filtering by prices, changing the display method, and so on. Figure 2.9 shows the main page of the Lelong.my. In this interface, it shows the button that able users to check the cart. Figure 2.10 shows the interface of the item description and its able users to add the items to the watch list and also allows users to contact the sellers by clicking the PM seller button. For the selling function, Lelong.my required sellers to register and purchase a webstore to continue the selling functions. Figure 2.11 shows the price list of the WebStore of the Lelong.my.



Figure 2.8 Logo of Lelong.my

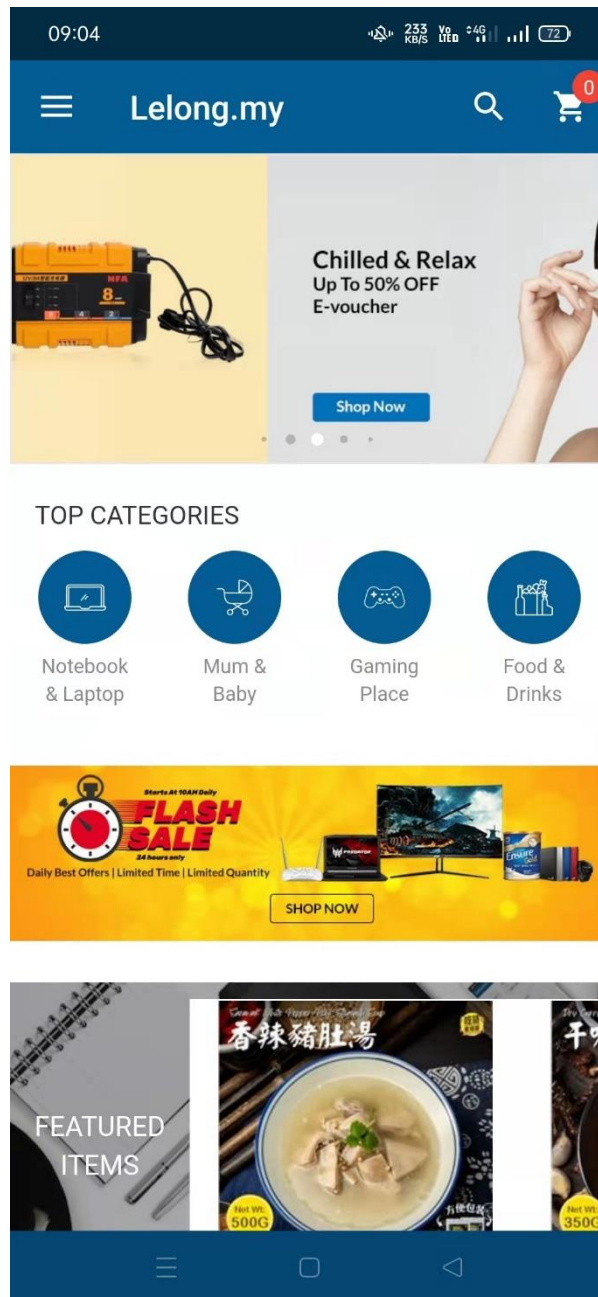


Figure 2.9 Main page of Lelong.my



Figure 2.10 Products detail interface

	Sign Up for FREE (Apr'22 - Jun'22)	MOST SIGN UP
WebStore	WebStore PLUS	WebStore PRO
WebStore + Store Front + Netpay Payment Service	Marketplace Store Front + Netpay Payment Service + Additional Plug-ins*	Own Branded Store Front + Marketplace Store Front + Netpay Payment Service + Additional Plug-ins* + Own Domain Name* (www.yourbrand.my)
For Malaysia Seller	For Malaysia Seller	For Malaysia Seller
RM 1.09 /day RM398/year (Save RM1,500)	FREE For 2 years RM498/year LIMITED TIME RM998.00/year (Save RM3,798)	RM 2.73 /day RM998/year LIMITED TIME RM1298.00/year (Save RM4,800)
	Sign Up for FREE	SIGN UP

Figure 2.11 Price Listed of Lelong.my WebStore

2.3 Comparison of Existing System

Table 2.1 shows the comparison of the existing system which is Mudah.my, Carousell, and Lelong.my. Those three applications is the application that related with my application. Table 2.1 will show the comparison of Graphical User Interface (GUI), Mobile OS, Connection Type, Language Provided, Size of Application, Functions, advantages and disadvantages of those applications.

Table 2.1 Comparison of the existing system

Application Name	Mudah.my	Carousell	Lelong.my
Graphical User Interface (GUI)	The interface is simple.	The user interface is appealing and well-organized, with a pleasing backdrop colour.	The interface is dull but it is simple and well organized.
Mobile OS	Android, iOS	Android, iOS	Android, iOS
Connection Type	Online	Online	Online

Language Provided	English	English	English, Bahasa Malayu, Mandarin
Size of Application	16.58MB	23.75MB	19.40MB
Functions	<ul style="list-style-type: none"> • Provided buttons “SELL” and “FIND” on the main page for users to choose the actions that users want to do which are sell items or search items. • Provided button that allows users to browse the items by filtering with categories. • Provided a button that allows users to log in or sign up for an account. • Provided buttons to filter the items which included price, type, condition, seller type, location, and so on. 	<ul style="list-style-type: none"> • Provided buttons to filter items by using categories on the main page. • Provided a search box for searching items. • Provided a button to view the liked items. • Provided a button to join groups as a community. • Provided a button to check the activities. • Provided a button for selling items. • Provided a button to display a QR code of users’ accounts and was able to scan other QR codes. • Provided a button to check or change the setting. 	<ul style="list-style-type: none"> • Provided buttons to filter items by using categories on the main page. • Provided a button for searching items. • Provided a button to view the shopping cart. • Provided a button that allows users to log in or sign up for an account. • Provided buttons to filter the items which included price range, auction type, condition, seller type, location, brand, and so on. • Provided a button to sort the items according to the latest items,

	<ul style="list-style-type: none"> • Provided a button to sort the items according to the latest items, lower price to higher price, and higher price to lower price. • Provided a button that users can add the items to the “My Favourite” list. • Provided a button that allows users to sell their items. • Provided buttons “CALL”, “SMS”, “EMAIL”, and “CHAT” to contact the seller. 	<ul style="list-style-type: none"> • Provided a button to filter and sort the items by location, sorting, product details, product conditions, price, and transaction methods. • Provided a button that able users to contact the sellers. • Provided a button that able users to add the items to the liked items. • Provided a button that able users to buy or argue prices with the sellers. • Provided a button that allows users to log in or sign up for an account. 	<p>lower price to higher price, higher price to lower price, best-selling, Wishlist, and Relevance.</p> <ul style="list-style-type: none"> • Provided a button to change the display method of the items. • Provided a button link to the Lelong webpage which able users to register for a WebStore.
--	--	--	---

Advantages	<ul style="list-style-type: none"> • Provided beginner tutorial on how to use the application. • Provided multiway for contact to the sellers. • Provided an easy way of selling items. • Provided multi-categories in this application. 	<ul style="list-style-type: none"> • Provided beginner tutorial on how to use the application. • Provided a function to argue the price of the items with the seller. • Provided an easy way of selling items. • Provided groups as a community. • Provided a QR code and scanning QR code functions. • Provided multi-categories in this application. 	<ul style="list-style-type: none"> • Provided functions that able users to change the display methods. • Provided multi-sorting to sort the items. • Provided multiple languages in the application.
Disadvantages	<ul style="list-style-type: none"> • Lack of resources in some types of categories. • Lack of languages in the application. 	<ul style="list-style-type: none"> • Lack of languages in the application. 	<ul style="list-style-type: none"> • Lack of categories. • Needed to register and buy a WebStore before starting to sell items.

Table 2.2 Function Comparison

Function	Mudah.my	Carousell	Lelong.my
Search function	✓	✓	✓
Filter items:			
1. Price	✓	✓	✓
2. Conditions	✓	✓	✓
3. Seller Type	✓	✓	✓
4. Location	✓	✓	✓
5. Transaction Methods	x	✓	x
6. Categories	✓	✓	✓
7. Brand	✓	x	✓
Sell Item function	✓	✓	✓
Buy Item function	✓	✓	✓
Log in and sign up function	✓	✓	✓
Sorting Item			
1. Latest	✓	✓	✓

2. lower price to higher price	✓	✓	✓
3. higher price to lower price	✓	✓	✓
4. best-selling	x	x	✓
Watchlist function	✓	✓	✓
Contact Seller Function			
1. Chat	✓	✓	✓
2. SMS	✓	x	x
3. Email	✓	x	x
4. Call	✓	x	x
Argue Price Function	x	✓	x
Generate QR Code Function	x	✓	x
Group Function	x	✓	x
Changing Display Method function	x	x	✓

2.4 Summary of Review Existing System

From the comparison of three applications which are Mudah.my, Carousell and Lelong.my, there are advantages and disadvantages in every application.

In terms of Graphical Users Interface (GUI), are Mudah.my and Carousell applications have good interfaces compared to Lelong.my application. The GUI of the Mudah.my is simple and simple is good. The GUI of Carousell is beautiful with attractive background colour and well organized but the GUI of Lelong.my is dull but it is simple and well organized. All three applications have required the internet to launch. Users can download those applications through Google Play Store and App Store. All three applications have the English language in application and Lelong.my have extra languages which are Bahasa Malaysia and Mandarin. The size of the applications of all three applications is lower than 25 MB.

All three applications provided buy and selling functions on their applications for free except Lelong.my which requires users to register and purchase a WebStore before starting to sell items. All three applications are able users to filter items by categories, price range, condition, location, and so on. Besides, those applications also able users to sort items according to the latest items, lower price to higher price, higher price to lower price, best-selling, and so on.

These three applications have their strength which is Mudah.my and Carousell have provided beginner tutorials on how to use the application, multiple categories, and provided an easy way of selling items. Mudah.my provided multiway for contact to the sellers. Carousell has provided a function that can let users argue prices with the seller, provided groups as a community, and provided a QR code and scanning QR code functions. For the Lelong.my, it provided a button that able users to change the display methods and it provided multiple languages in the application.

These three applications have their weakness which Mudah.my and Carousell were lack in system language in applications. There were just provided the English language in the application. Besides Lelong.my were provided other languages like

Bahasa Malaysia and Mandarin. Mudah.my lack in the resources in some types of categories and Lelong.my lack in the categories. The weakness of the Lelong.my was it needed users to register and purchase a WebStore on their website before starting to sell items.

For the UMP Item Trading Application, the system will include some of the features from those existing application which include providing multiple categories for user to filter, providing a multiple way that let user contact sellers, the buyers can argue price with the sellers freely through by using the contact method provided by the system, and for the unique function that UMP Item Trading Application provided is able seller to donate the used item or product like the used bicycles that can let the users to receive the item freely. Based on the donation function, it can resolve the waste of resources which the seller can donate the used item or item that there no need and donate to junior or someone that needed the item. Therefore, it can resolve the problem of the waste of resources in UMP. For the proposed system, the application should able on Google Play Store in Android. The advantages of the proposed system is free to download and user friendly and clear interface for user to use.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will describe the Software Development Life Cycle (SDLC) and the Model-View-Presenter (MVP) architectural pattern used in this project. The SDLC is a six-phase process that includes gathering and analyzing requirements, designing, implementing and coding, testing, deployment, and maintenance. Each phase will have its objective, which is to ensure that the system or application is of high quality. There are plenty of SDLC models available for project developers to follow, and the SDLC model chosen should be appropriate for the development timeframe, application size, and resources available for the project. MVP is an architectural pattern that comprises three connecting layers which are Model, View, and Presenter. This pattern is derived from the classic MVC (Model-View-Controller) pattern. MVP provides for more decoupling of components, designed to allow each layer of the application to work independently.

The Rapid Application Development (RAD) model was chosen as the methodology for building this project. The justification for choosing this model will be explained in detail in the Methodology section. While the architectural pattern, MVP used in the development of the application will also be explained in detail.

3.2 Methodology

3.2.1 Rapid Application Development (RAD)

Rapid Application Development (RAD) is the software development methodology that was utilized to create this project. It prioritizes the rapid release of prototypes and iterations based on user feedback and software usage.

Rapid Application Development (RAD)

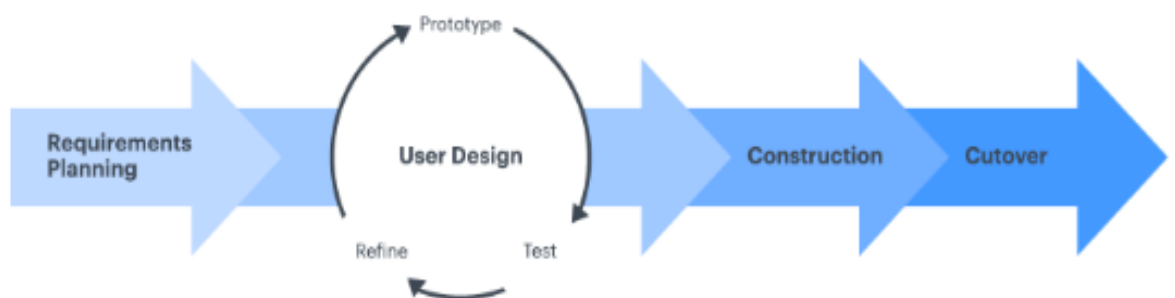


Figure 3.1 Rapid Application Development (RAD) Methodology

RAD uses the agile method as the project management strategy within the process of software development. There are four main phases in RAD methodology which includes:

Table 3.1 RAD Methodology Phases

Phases	Explanation
Requirements Planning	<p>At the beginning of RAD, it specifies the details requirement of the software project.</p> <p>Software Requirement Specification (SRS) document is developed in Requirements Planning phase. SRS is a document that describes what the software will do and how it is expected to perform. The system's functional</p>

	<p>requirement, non-functional requirement, constraints, and limitation had been defined in 3.3 Project Requirement. At this phase, the requirements should be gathered and defined. Some of the examples are the what is the problem need to be solved in this project, how can this project improve the user's experience, how can this project improve the system performance, what is the estimation of time, cost, resources, risks of this project. The requirements will be documented in Software Requirement Specification (SRS) in details.</p>
User Design	<ul style="list-style-type: none"> • This is the development phase that creates prototypes with different features and functionality of the system. • When the prototype is approved, is then ready to start software implementation. <p>All the related diagrams such as context diagram, use case diagram, data flow diagram and others will be designed and presented in Software Requirement Specification (SRS) document. Further detail architecture design will be discussed in Software Design Description (SDD) document. The proposed design had been created which can let the user clear on the design of the application.</p>
Construction	<p>This phase focus on application development, coding, integration, and system testing.</p> <p>The prototype of the system had been developed in construction phase. Coding, integration, and system testing is doing in this phase. In this phase, the product will be integrated and tested to ensure the final product is bug-free and it meet all the documented requirement of the system. A series of tests is conducted to check bugs, ensure the flow of the system is working as stated in the document and meets the requirements. The User Acceptance Test</p>

	(UAT) was planned as the system testing plan in this system.
Cutover	It is a final task in the SDLC implementation phase. The final product will be released and launched to the customer. The development team will demonstrate the system flows and ways to use the system to the customer.

3.2.2 Model-View-Presenter (MVP)

The MVP is the architectural pattern used to organize the code in the proposed project. This architectural pattern is often utilized in mobile application code development. This pattern allows the code to be separated and organized in model, view, and presenter. The three major layers which are Model, View, and Presenter will communicate with one another in an organized manner. Figure 3.2 shows how the MVP architecture works and communicates with each other within its layer.

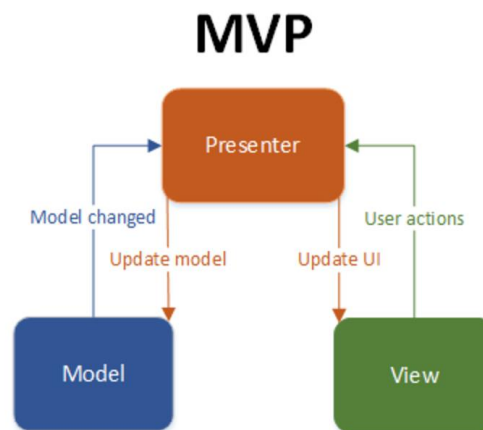


Figure 3.2 Model-View-Presenter Model

Each layer in the MVP architectural pattern will have its role and responsibility as given. Table 3.2 shows the role and responsibility of each application layer.

Table 3.2 MVP architectural pattern role and responsibility

Layer	Responsible
Model	<p>A data layer is an object or class that holds information about the data in the application.</p> <p>A layer for storing data. It's in charge of domain logic (real-world business rules) as well as database and network interactions.</p>
View	<ul style="list-style-type: none"> • It represents the UI that users will see and interact with. • It involved visualization and input detection. • Each view will only have one presenter to communicate with. <p>The user interacts with the system through the UI (User Interface) layer. It visualises the data and maintains track of the user's actions in order to alert the Presenter.</p>
Presenter	<p>It acts as the middleman between the Model and View layer. It tells the View to change based on the underlying data and updates the Model data.</p> <p>Fetch data from the model and use UI logic to determine what should be displayed. It keeps track of the View's status and reacts to the user's input notifications from the View.</p>

3.3 Project Requirement

The Project Requirement will define the functional requirements, non-functional requirements, constraints, and limitations of the application.

3.3.1 Functional Requirement

Mobile application functional requirements define what must be included in a system or product, as well as what activities users can perform to interact with the software. They are the ones who make the system's decisions. The functional requirement in this application will be stated below:

1. System shall able users to register and log in to the application easily.
2. System shall not allow user register two account using one email address.
3. System must allow user to reset password.
4. System shall allow users to view the product.
5. System shall provide user-friendly product filtering and sorting.
6. System shall provide detailed items description.
7. System shall able users to view their profiles.
8. System shall able users to edit their profile information.
9. System shall provide seller to edit their item information.
10. System shall able users to add items to the Watchlist.
11. System shall able users to remove item from the Watchlist.
12. System shall able users to contact the seller in multiple ways.
13. System must able users to sell their items by uploading pictures and details of the items.
14. System must able users to search for items.
15. System should provide multiple categories.
16. System should able admin to view user information.
17. System should able admin to add new user.
18. System should able admin to edit user information.
19. System should able admin to block the user information.

3.3.2 Non-functional Requirement

Non-functional requirements for mobile applications explain the characteristics and aspects of a solution, such as how and why it will function. The non-functional requirement in this application will be stated below:

Table 3.3 Non-functional Requirement

Non-Functional Requirement	Description
Availability	<ul style="list-style-type: none">• The term "availability" represents the belief that a user will be able to use the system at any particular time. Example:• During a month, the system may be available 95 percent of the time.
Reliability	<ul style="list-style-type: none">• This quality feature describes the likelihood that the system or one of its components will operate without failure for a defined amount of time under specified conditions. Example:• If a system has a 90 percent reliability for a month, that means there's a 90% probability it won't have a critical failure during that month under normal usage conditions.
Extensibility	<ul style="list-style-type: none">• Extensible applications have a long lifespan, avoiding the costly process of acquiring huge, inflexible systems and retiring them as business needs change. Extensibility enables businesses to seize opportunities and respond to threats.

	<p>Example:</p> <ul style="list-style-type: none"> • Requirements for extensibility attribute in case additional functional requirements are required
Performance	<ul style="list-style-type: none"> • The number of concurrent users or transactions the system must handle, as well as the system's response time <p>Example:</p> <ul style="list-style-type: none"> • Within 10 seconds, each request should be processed.
Security	<ul style="list-style-type: none"> • Software's functional security criteria describe a security function that it must be able to perform. A subset of the broader functional criteria are the functional security requirements. <p>Example:</p> <ul style="list-style-type: none"> • Sensitive data transferred over the Internet between the server and the browser must be encrypted by the system.
Usability	<ul style="list-style-type: none"> • Usability refers to the ability to use a particular product • If a feature's usability is excellent, users can quickly figure out what it is and what it can accomplish. <p>Example:</p> <ul style="list-style-type: none"> • Users might predict that pressing a button with a magnifying glass on it will bring up a search bar.
Maintainability	<ul style="list-style-type: none"> • Maintainability refers to the time it takes to repair, update, or adjust a system or one of its components in order to improve performance or other attributes, or to adapt to a changing environment. <p>Example:</p> <ul style="list-style-type: none"> • If a component has an 80 percent maintainability for 24 hours, it means it has an 80 percent chance of being repaired in that time.

3.3.3 Constraints

A constraint is defined when an application feature or application content should be used. Users and user roles, as well as the characteristics of the device on which the mobile application is meant to run and the hardware available on the device, can all be used to limit access.

Table 3.4 Constraints

Constraints	Description
Policies	<ul style="list-style-type: none">• The system requires an active licence, either on an individual basis for smaller businesses or as part of a multi-license agreement for bigger enterprises.• The admin should agree to “EULA” (End-Use License Agreement) or related terms and conditions before using this system.
Execution Rate	<ul style="list-style-type: none">• The system should have better effectiveness like the rate of the response time of the system must be lower than the rate of execution. This is called signal timing requirements.• The execution time allocated to the control task for the detection of a command.• The reaction time allotted to the control task for the processing of the command and effecting the relevant change in the system.
Higher-order language requirements	<ul style="list-style-type: none">• The system should use higher-order language because they are easier to read, write, and maintain.
Signal handshake protocols	<ul style="list-style-type: none">• The system should have good control of data flow so the system can run smoothly without crashing.

	<p>Example:</p> <ul style="list-style-type: none"> • The system is sending data to an instrument that could accept no more data for the time being.
Site security and data privacy constraints	<ul style="list-style-type: none"> • User cannot access and edit the information of storage. • User cannot reach and edit the profile of the admin. • Admin cannot access and edit the personal information of the user.

3.3.4 Limitations

Table 3.5 Limitation

Limitation	Description
Small Screen	<ul style="list-style-type: none"> • Phone screens can only hold a fraction of the content that desktop and laptop screens can. As a result, the screen size of mobile devices has become a significant constraint. • On a 30-inch monitor, the content displayed above the fold requires 5 screenfuls on a little 4-inch screen.
Single Window	<ul style="list-style-type: none"> • The great majority of users only view one window at a time (and consequently one application or website). • They are unable to split the screen (like on a desktop) and simultaneously use two different apps.
Interruptible	<ul style="list-style-type: none"> • An external incident in the outside environment may cause us to stop what we were doing on the little screen and pay attention to the external incident. • On mobile devices, attention is frequently fragmented, and sessions are brief.
Connectivity Issues	<ul style="list-style-type: none"> • M-Commerce requires a high-speed internet connection.

	<ul style="list-style-type: none"> • Slow transmission speeds may be a big stumbling barrier for M-Commerce, as they restrict mobile phone users from accessing the internet.
Fraud Risk	<ul style="list-style-type: none"> • In M-Commerce, there is a significant risk of fraud. • Many marketers are ill-equipped to cope with it.

3.4 Proposed Design

3.4.1 Context Diagram

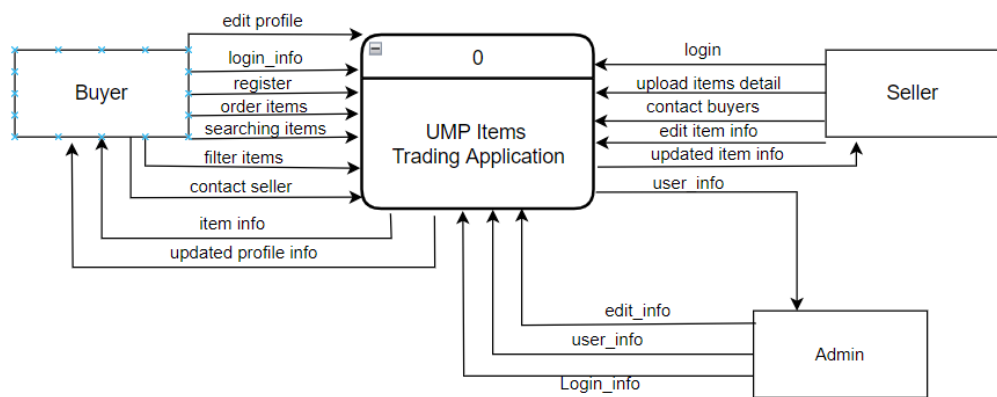


Figure 3.3 Context Diagram

Based on figure 3.3 shows the Context Diagram (Data Flow Diagram Level 0) of the application. In engineering, a system context diagram depicts the border between a system or component of a system and its environment, as well as the entities with whom it interacts. The figure depicts the system at a high level. It resembles a block diagram.

Solution explanation:

- Buyer will register a new account for the application
- Buyer will log in to the application using the username and password that has been registered
- Buyer will order items with the seller by the application

- d) Buyers will enter new profile information to update their profile information
- e) Buyer will enter the item name that they want to search
- f) Buyer will select the filter function to filter the searching items
- g) Buyer will select a method to contact the seller
- h) Buyer will receive the items information after searching and the filtering function is applied
- i) Buyer will receive the updated information of the profile
- j) Seller will enter item detail information for selling the items
- k) Seller will reply to question from the buyer
- l) Seller will enter new information about the items to update the latest item information
- m) Seller will receive the updated items information after the update by the seller
- n) Admin will receive the user information from the database.
- o) Admin will login to the application by using username and password that specific for admin.
- p) Admin will enter the new user information to add user into database.
- q) Admin will enter the new information of the user to update the user information.

3.4.2 Use Case Diagram and Description

A use case diagram shows all the possible interaction of the actor with a system and it is represented in a graphical format. Figure 3.4 below show the use case diagram of the study. There are two actors in this system which are user and admin. Besides, there are five modules in this system which are manage registration, manage profile, manage order, manage goods and manage user.

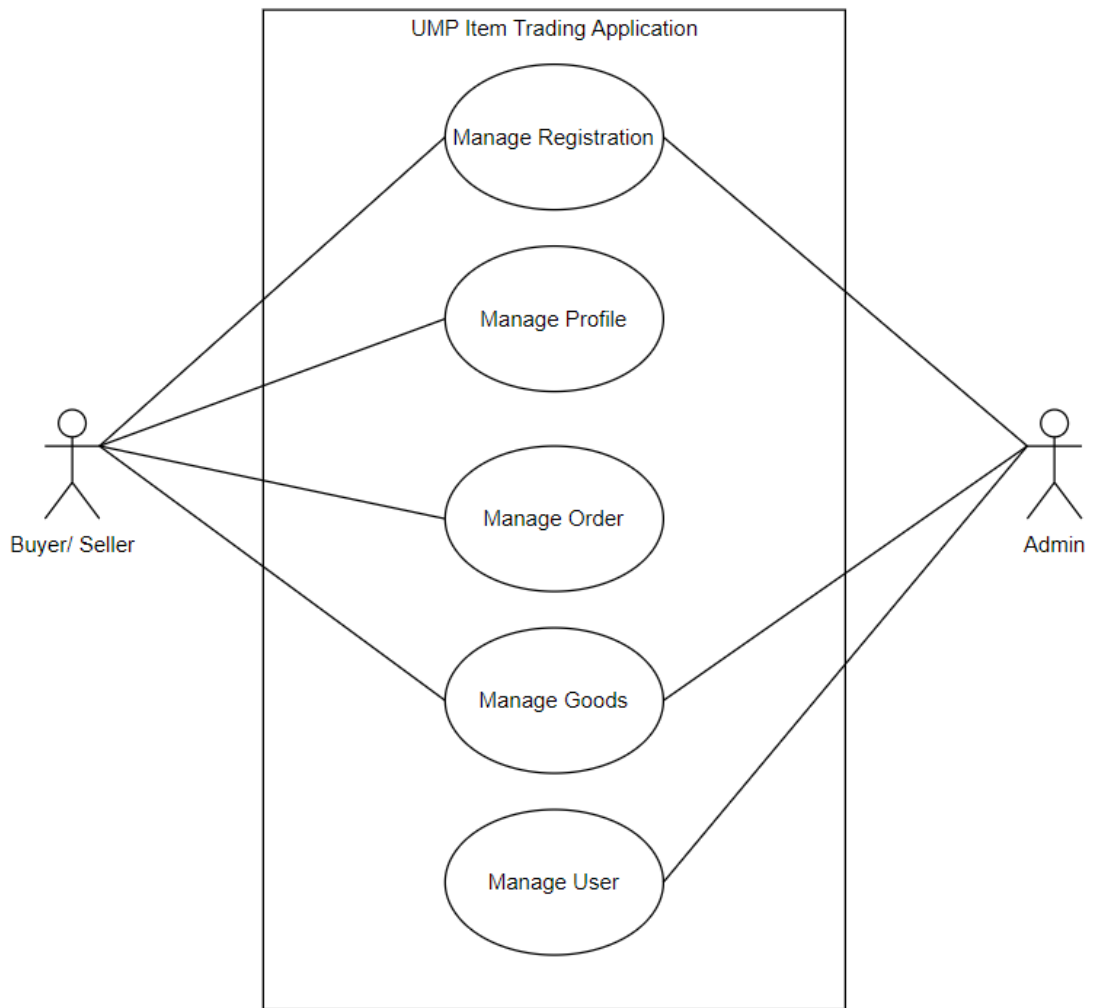


Figure 3.4 Use Case Diagram

3.4.2.1 Manage Registration

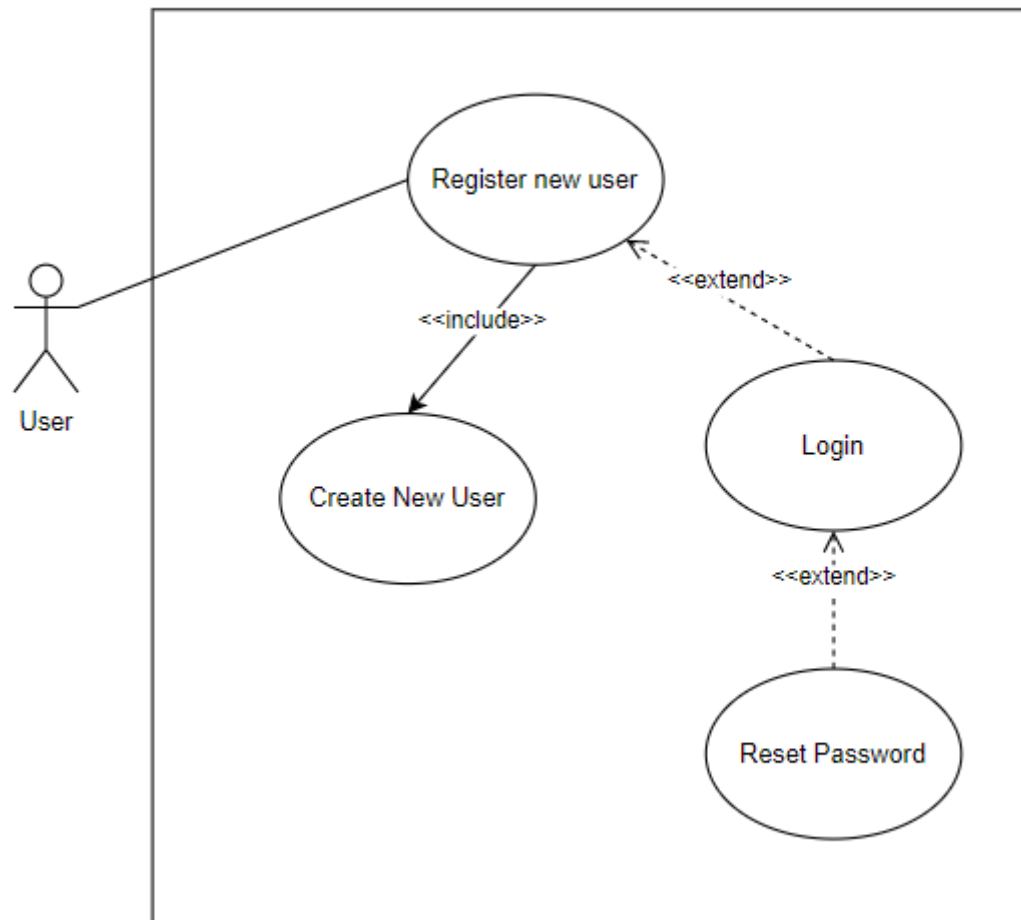


Figure 3.5 Use Case Diagram of Registration

Table 3.6 Manage Registration

Use Case ID	ITA_UCI_1000
Description	This use case describes users registering a new user in the application and users can log in to the application by the registered account.

Actor	User
Pre-Conditions	User must have a valid student id/staff id, password, email, and phone number to register
Basic Flow	<ol style="list-style-type: none"> 1. The use case starts when the user wishes to create an account by clicking the <<Register>> button 2. The system verifies that the user is entering the correct data and the data save in user information <p>[E1 - Incorrect data]</p> <ol style="list-style-type: none"> 3. The system provides a portal to access the user through a secure channel using login credentials. <p>[A1 – Log in]</p> <ol style="list-style-type: none"> 4. Users can log in to the application <p>[A2 – Reset Password]</p> <ol style="list-style-type: none"> 5. Users can reset their password through the link on the login page 6. The use case end.
Alternative Flow	<p>[A1 – Log in]</p> <ol style="list-style-type: none"> 1. The user input the username and password 2. The user clicks the <<Log in>> button after inputting data <p>[E2 - Incorrect username or password]</p> <ol style="list-style-type: none"> 3. Continue step 5 in the basic flow

	<p>[A2 – Reset Password]</p> <ol style="list-style-type: none"> 1. The user inserts the valid email to reset the password 2. Continue step 4 in the basic flow
Exception Flow	<p>[E1 - Incorrect data]</p> <ol style="list-style-type: none"> 1. The system displays an error message showing that the registration process is failed. 2. The user continues to step 2 in the basic flow. <p>[E2 - Incorrect username or password]</p> <ol style="list-style-type: none"> 1. The system displays an error message, showing that the login process is failed. 2. The user continues to step 2 in the basic flow.
Post-Condition	<ul style="list-style-type: none"> • The system provides a portal to access the user through a secure channel using login credentials.
Rules	-
Constraints	<ul style="list-style-type: none"> • User id and password just type in the correct blank space.

3.4.2.2 Mangeprofile

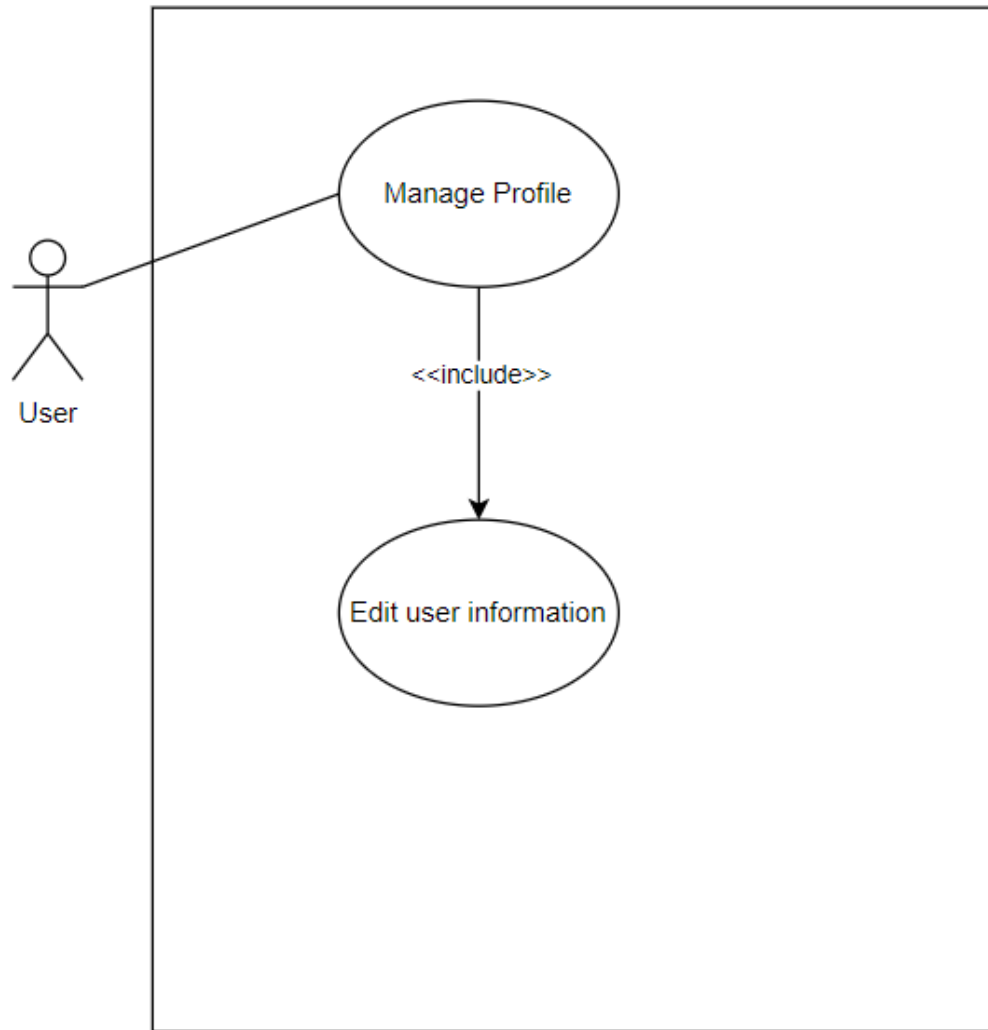


Figure 3.6 Use Case Diagram of Manage Profile

Table 3.7 Manage Profile

Use Case ID	ITA_UCI_2000
Description	This use case describes users manage their profiles to view and edit the profile information.

Actor	User
Pre-Conditions	<ul style="list-style-type: none"> • User must have an account • User needs to log in to the account before viewing and editing the profile
Basic Flow	<ol style="list-style-type: none"> 1. The use case starts after users login into their account 2. User needs to click the person icon to view the profile information 3. Users can view or enter new information for updating the profile <p>[E3 - Incorrect data]</p> <ol style="list-style-type: none"> 4. The use case ends
Alternative Flow	None
Exception Flow	<p>[E3 - Incorrect data]</p> <ol style="list-style-type: none"> 1. The system displays an error message showing that the updating profile process is failed. 2. The user continues to step 2 in the basic flow.
Post-Condition	<ul style="list-style-type: none"> • System will update the new information in the application
Rules	<ul style="list-style-type: none"> • Users can only edit their profile of themselves
Constraints	<ul style="list-style-type: none"> • User unable to edit the username in the profile

3.4.2.3 Manage Goods

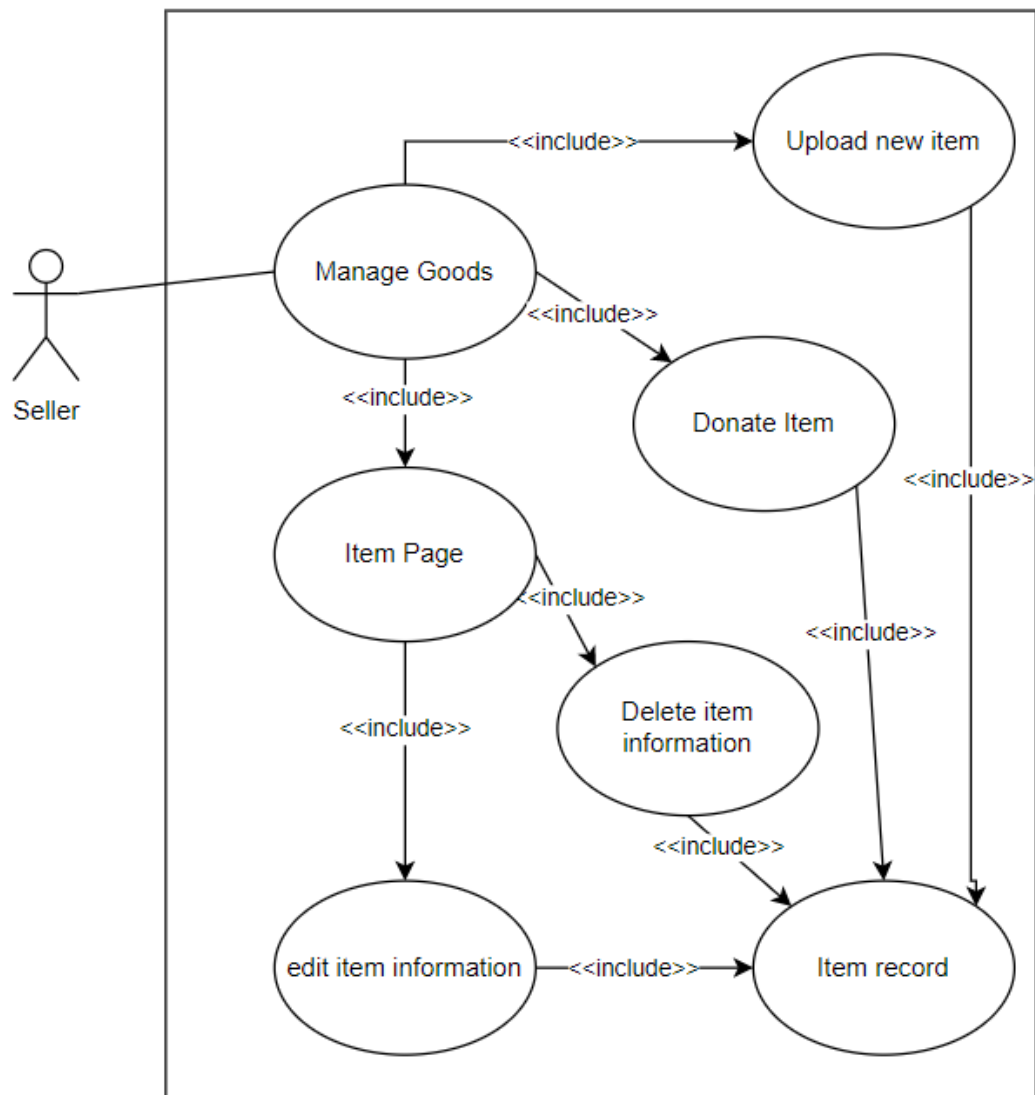


Figure 3.7 Use Case Diagram of Manage Goods

Table 3.8 Manage Goods

Use Case ID	ITA_UCI_3000
Description	This use case describes seller manage their trade items by uploading and editing item information.

Actor	Seller
Pre-Conditions	<ul style="list-style-type: none"> • Seller must have an account • Seller needs to log in to the account before uploading and editing item information
Basic Flow	<ol style="list-style-type: none"> 1. The use case starts after clicking the <<sell>> button on the main page 2. The seller can choose: <ul style="list-style-type: none"> - Click on the <<Upload New Item>> button to upload the item to the application database [A1: Upload new item] - Click on the <<Item Page>> button to link to the item page in the application [A2: Item Page] - Click <<edit>> button to edit an item in the application database [A3: Edit item] - Click on the <<Delete item>> button to delete an item in the application database [A4: Delete item] - Click on the <<Donate Item>> button to add a donation item. [A5: Donate item] 3. The use case ends

Alternative Flow	<p>[A1: Upload new item]</p> <ol style="list-style-type: none"> 1. The use case starts when the user clicks on the <<Upload New Item>> button 2. Seller needs to select a photo or snap a photo as an item picture. 3. Seller requires to insert all required information of the item. 4. Click <<Upload>> to upload the item to the application 5. Continue step 3 in the base flow <p>[A2: Item Page]</p> <ol style="list-style-type: none"> 1. The use case starts when seller click <<Item Page>> button 2. Seller can click: <ul style="list-style-type: none"> - <<Edit>> Button <p style="text-align: center;">[A3: Edit item]</p> <ul style="list-style-type: none"> - <<Delete Button>> <p style="text-align: center;">[A4: Delete item]</p> <ul style="list-style-type: none"> - Seller can modify status of the item <ol style="list-style-type: none"> 3. Continue Step 1 in Alternative 2 <p>[A3: Edit item]</p> <ol style="list-style-type: none"> 1. The use case starts when the user clicks on the <<Edit Item>> button 2. Seller can enter or edit information of the items
------------------	--

	<p>3. Click <<Update>> to update the information on the existing item.</p> <p>4. Continue step 3 in the base flow</p> <p>[A4: Delete item]</p> <p>1. The use case starts when the user clicks on the <<Delete>> button</p> <p>2. Click <<Delete>> to delete the existing item.</p> <p>3. Continue step 3 in the base flow</p> <p>[A5: Donate item]</p> <p>1) The use case starts when the user clicks on the <<Donate Item>> button</p> <p>2) Seller needs to select a photo or snap a photo as an item picture</p> <p>3) Seller requires to insert all required information of the item</p> <p>4) Click <<Upload>> to upload the item to the application</p> <p>5) Continue step 3 in the base flow</p>
Exception Flow	None
Post-Condition	<ul style="list-style-type: none"> • New item was uploaded to application database • New item information was updated by the seller
Rues	<ul style="list-style-type: none"> • Seller only can edit the items that uploaded by themselves
Constraints	-

3.4.2.4 Manage Order

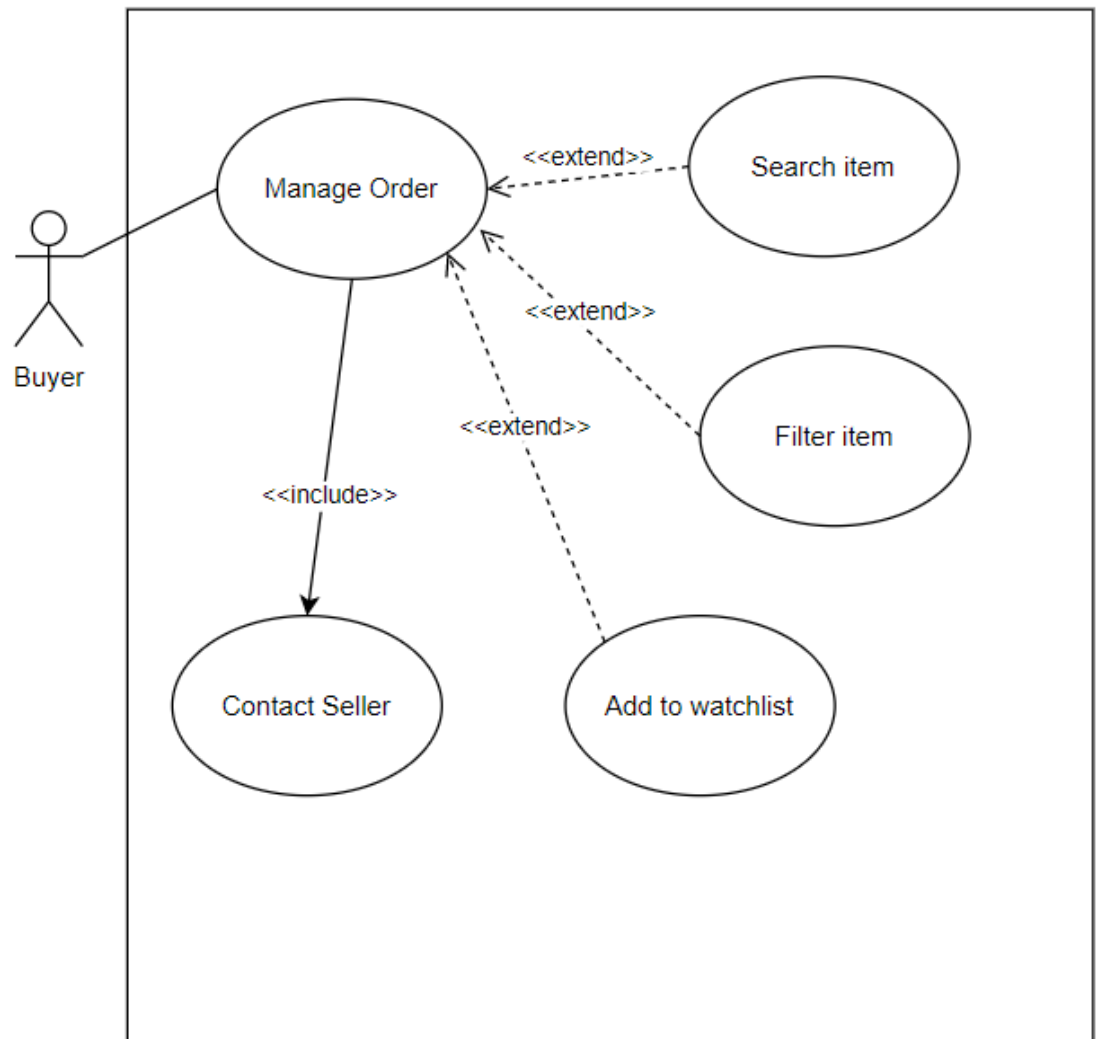


Figure 3.8 Use Case Diagram of Manage Order

Table 3.9 Manage Order

Use Case ID	ITA_UCI_4000
Description	This use case describes buyers manage orders in the application
Actor	Buyer

Pre-Conditions	<ul style="list-style-type: none"> • Buyer must have an account • Buyer needs to log in to the account before manage orders by the buyers.
Basic Flow	<ol style="list-style-type: none"> 1) The user case starts after the user log in to the application 2) The buyer can choose: <ul style="list-style-type: none"> - Search items by entering item names into the search box for searching items. <p style="text-align: center;">[A1: Search item]</p> <ul style="list-style-type: none"> - Filter items by selecting the type of filter prepared by the application <p style="text-align: center;">[A2: Filter item]</p> <ul style="list-style-type: none"> - Add to the watchlist by clicking the icon of the watchlist <p style="text-align: center;">[A3: Add item to watchlist]</p> <ul style="list-style-type: none"> - Contact Seller by selecting the contact method prepared by the application <p style="text-align: center;">[A4: Contact Seller]</p> 3) Use case ends
Alternative Flow	<p>[A1: Search item]</p> <ol style="list-style-type: none"> 1. The use case starts when the user clicks the search on the search box 2. Enter the item name in the search box 3. The application will display the information of the searching item 4. Continues step 2 in the basic flow

	<p>[A2: Filter item]</p> <ol style="list-style-type: none"> 1. The use case starts when the user selects a filtering function like filter by categories, condition, and so on. 2. The application will display the information of the item that is filtered 3. Continues step 2 in the basic flow <p>[A3: Add item to watchlist]</p> <ol style="list-style-type: none"> a. The use case starts when the user clicks the watchlist icon b. The items will add to the watchlist c. Continues step 2 in the basic flow <p>[A4: Contact Seller]</p> <ol style="list-style-type: none"> 1. The use case starts when the user wants to purchase an item from the seller 2. The buyer can select the method to contact the seller using the Call, SMS, email, or Chatbox provided by the application. 3. Continues step 2 in the basic flow
Exception Flow	None
Post-Condition	<ul style="list-style-type: none"> • Buyer able contact to seller • Buyer able to add the item to the watchlist • Buyer able to search and filter items.

Rules	-
Constraints	-

3.4.2.5 Manage User

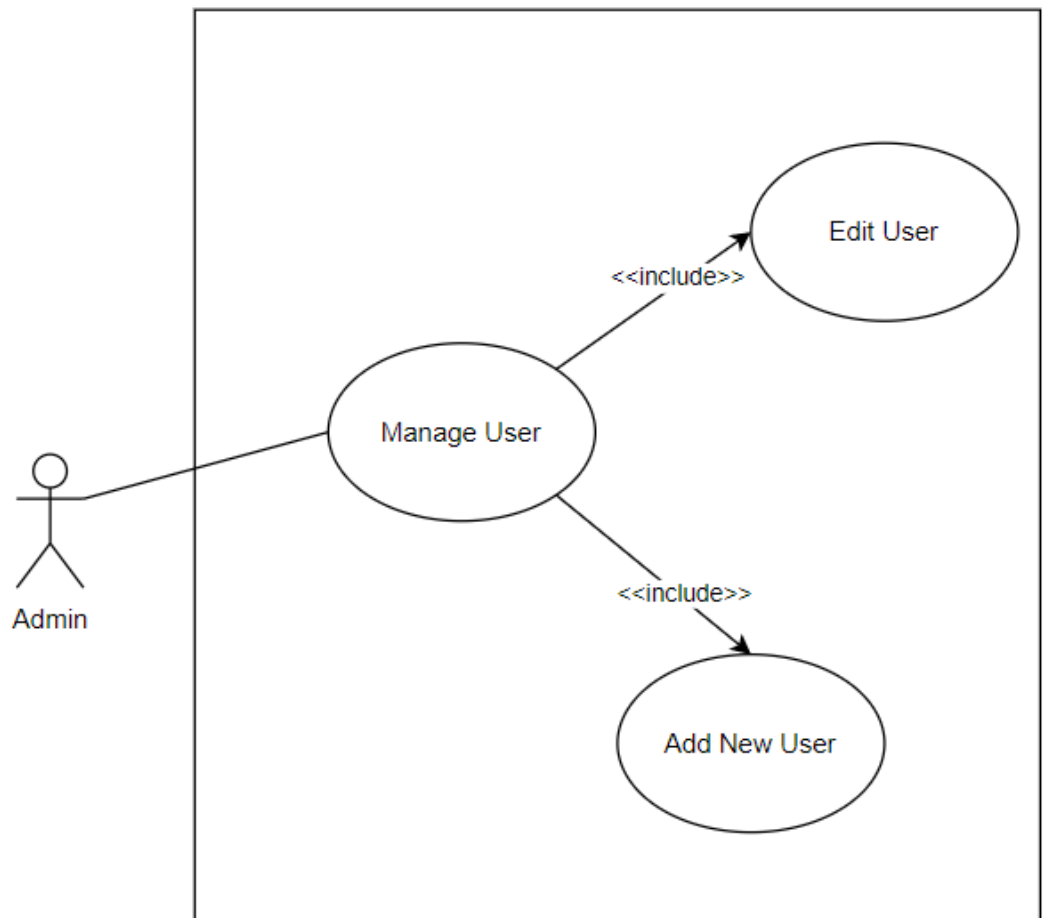


Figure 3.9 Use Case Diagram of Manage User

Table 3.10 Manage User

Use Case ID	ITA_UCI_5000
-------------	--------------

Description	This use case describes admin to manage the user.
Actor	Buyer
Pre-Conditions	<ul style="list-style-type: none"> • Admin had login to the system
Basic Flow	<p>1) The use case start when the admin login to the system.</p> <p>2) The admin able to:</p> <ul style="list-style-type: none"> - Add new user [A1: Add new user] - Edit user [A2: Edit user] <p>3) The use case end</p>
Alternative Flow	<p>[A1: Add new user]</p> <ol style="list-style-type: none"> 1) Admin click the <<Add new user>> button 2) Admin entering the user information 3) Click the <<Add>> button 4) Continues step 2 in basic flow <p>[A2: Edit user]</p> <ol style="list-style-type: none"> 1. Admin click the user that selected 2. Select user status which is block or free 3. Click the <<Edit>> button 4. Continues step 2 in basic flow
Exception Flow	None

Post-Condition	<ul style="list-style-type: none"> • Admin able to add new user • Admin able to edit user information
Rules	-
Constraints	-

3.4.3 Activity Diagram

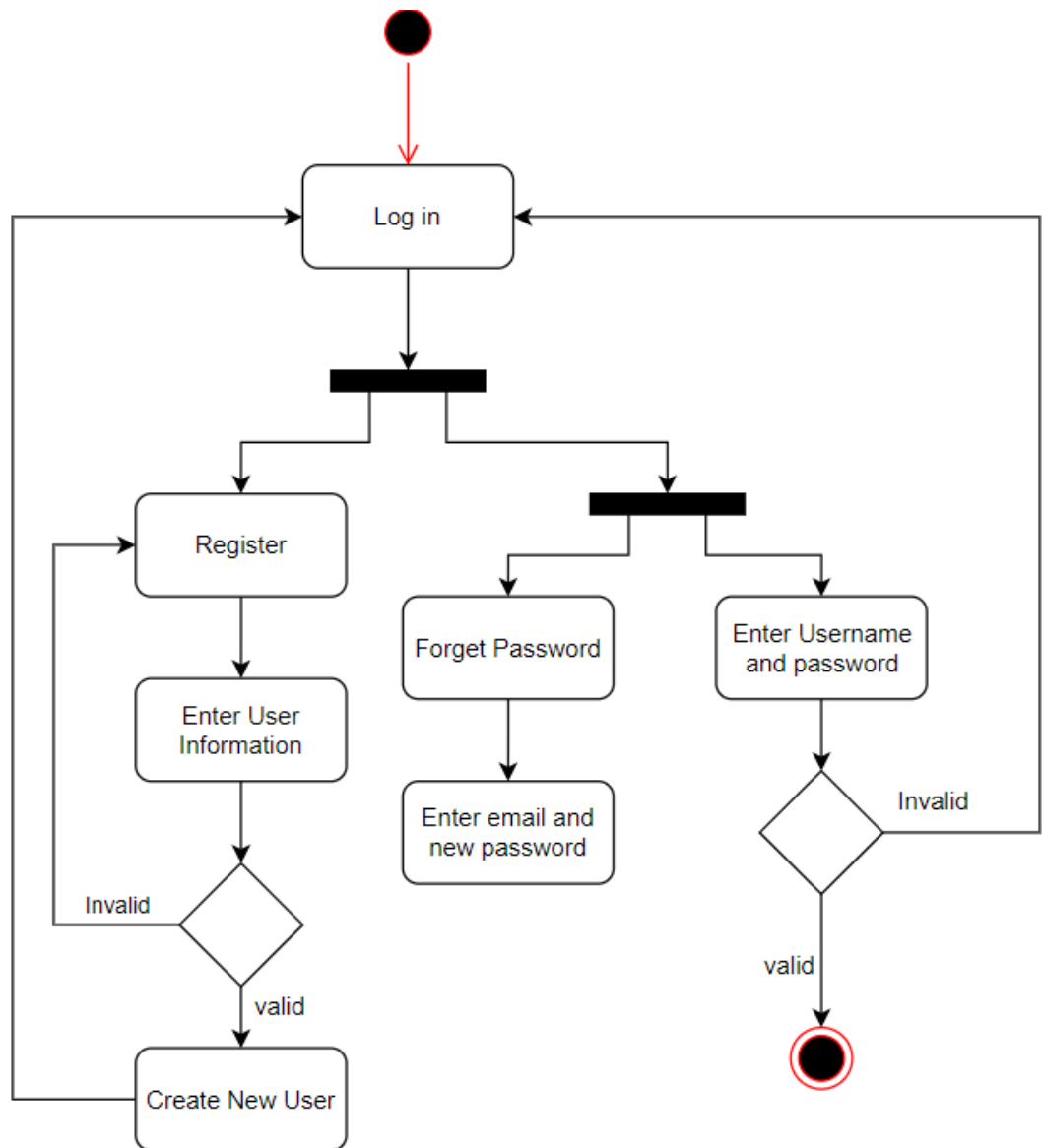


Figure 3.10 Activity Diagram of Manage Registration

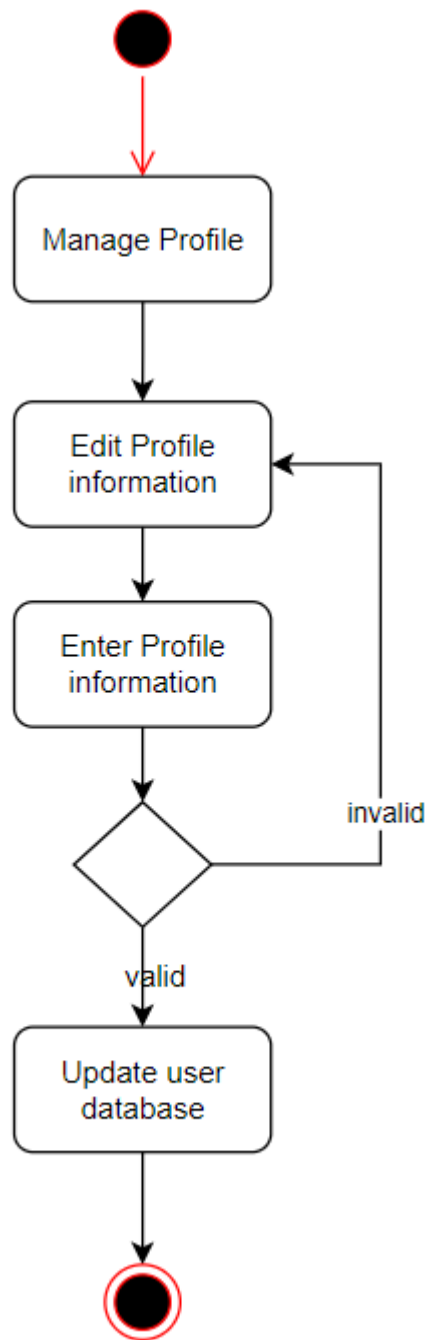


Figure 3.11 Activity Diagram of Manage Profile

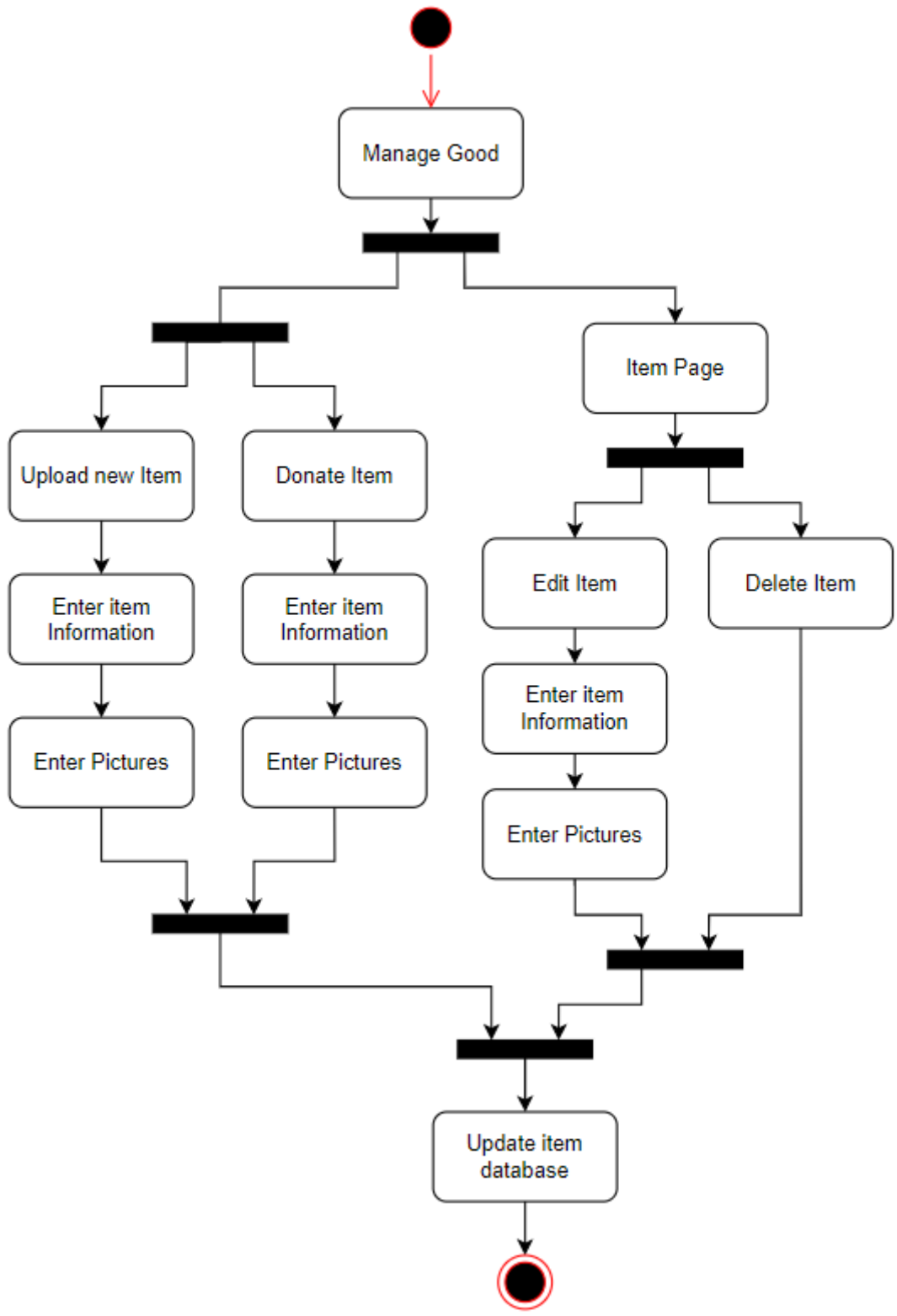


Figure 3.12 Activity Diagram of Manage Goods

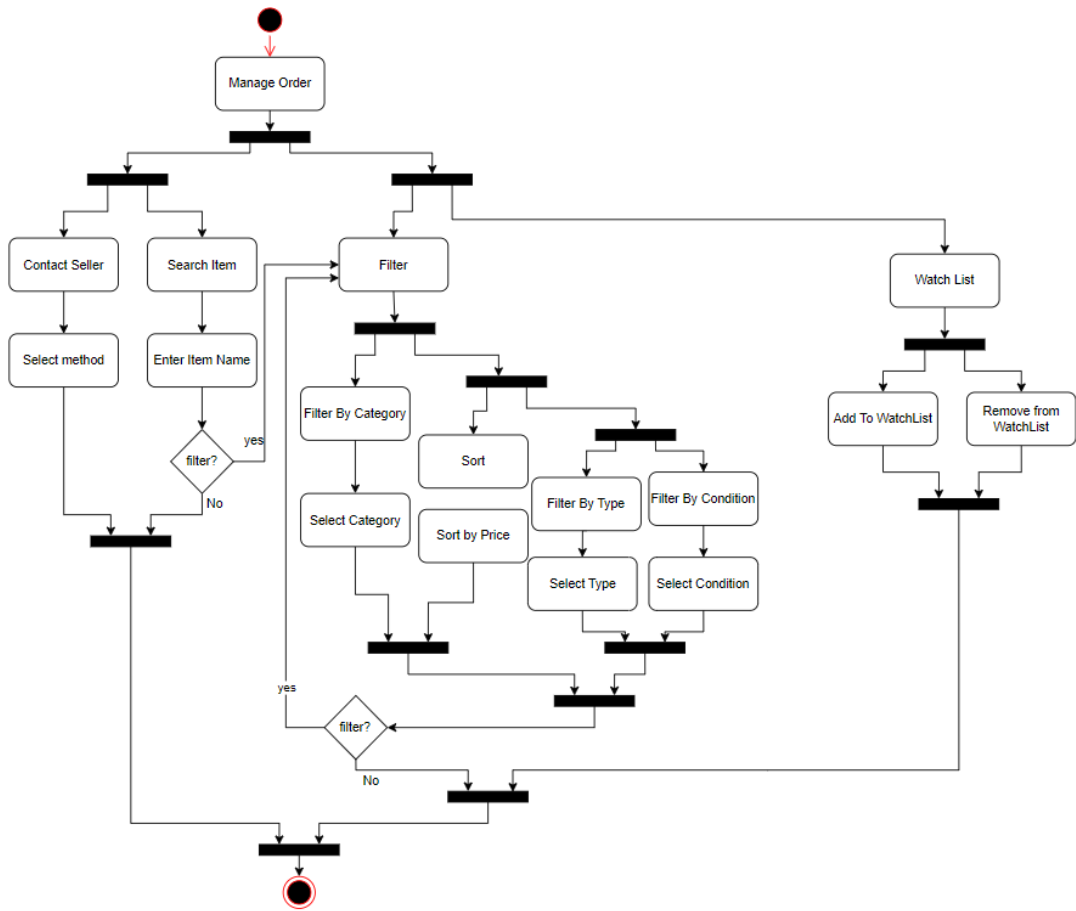


Figure 3.13 Activity Diagram of Manage Order

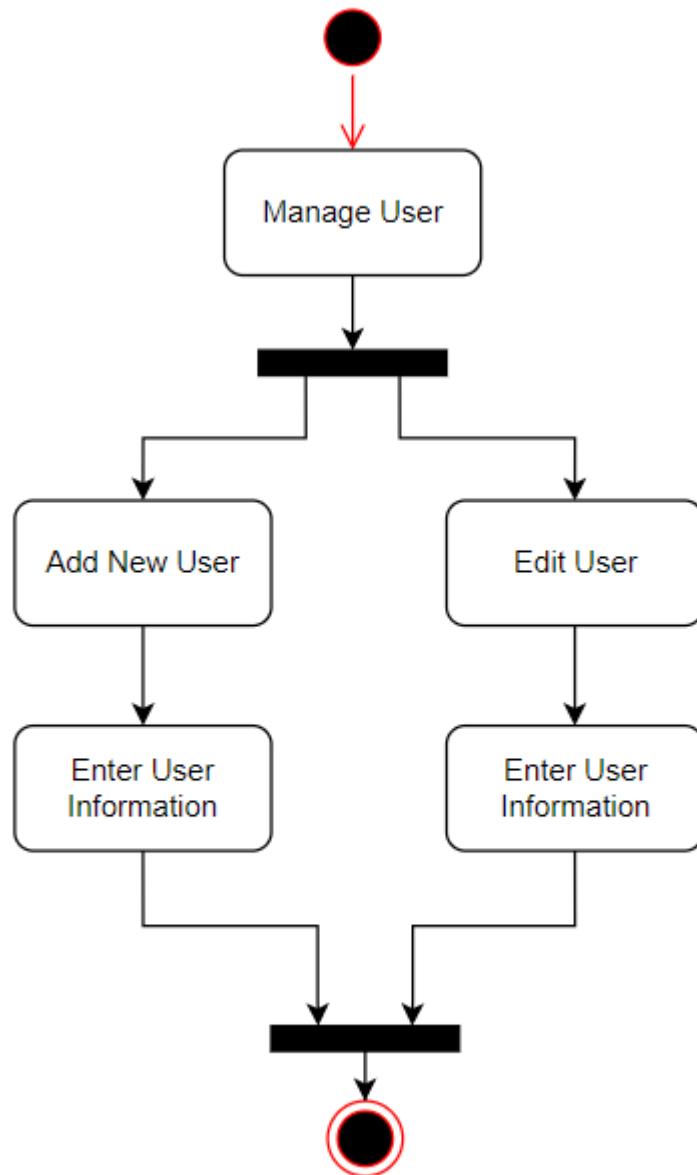


Figure 3.14 Activity Diagram of Manage User

3.5 Data Design

3.5.1 Entity Relationship Diagram (ERD)

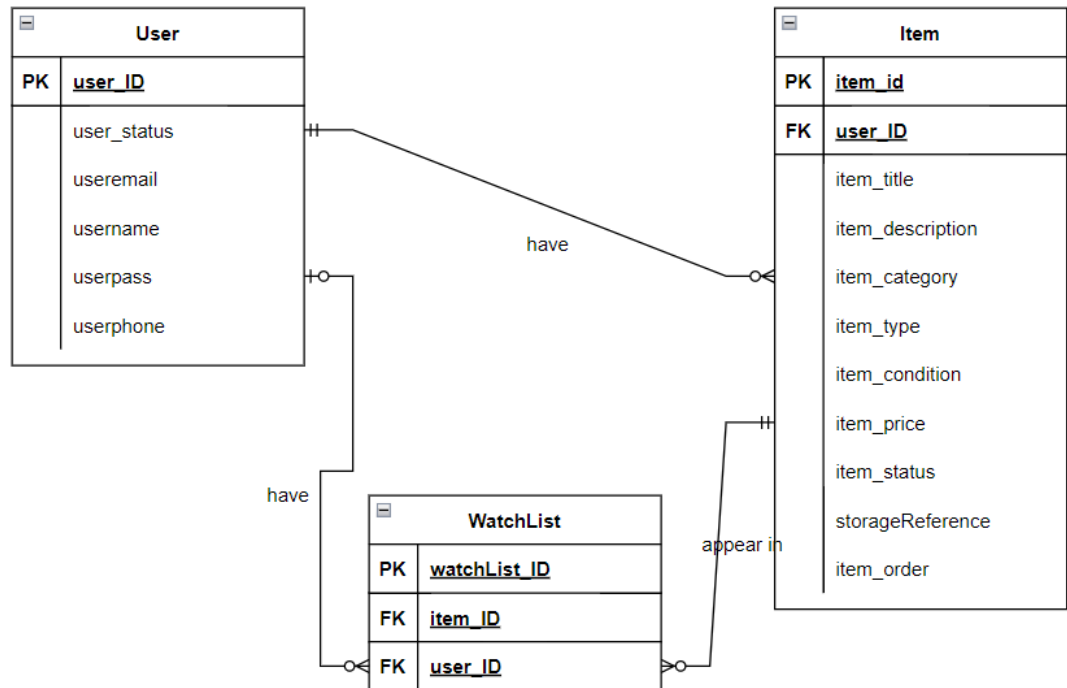


Figure 3.15 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) is a flowchart that illustrates how the entities relate with each other in a system. Figure 3.15 shows the ERD and there have 3 entities in this system which are User, Item, and WatchList.

3.5.2 Data Dictionary

Table 3.11 Data Dictionary-User

Field Name	Description	Data Type	Constraint
user_ID	ID of user	VARCHAR(255)	PK

username	Name of User	VARCHAR(255)	
useremail	Email of user	VARCHAR(255)	
userphone	Phone Number of user	VARCHAR(255)	
user_status	Status of user	VARCHAR(255)	
userpass	Password of user	VARCHAR(255)	

Table 3.12 Data Dictionary-Item

Field Name	Description	Data Type	Constraint
item_ID	ID of item	VARCHAR(255)	PK
user_ID	ID of user	VARCHAR(255)	FK
Item_title	Title of item	VARCHAR(255)	
Item_description	Description of item	VARCHAR(255)	
Item_category	Category of item	VARCHAR(255)	
Item_type	Type of item	VARCHAR(255)	
Item_condition	Condition of item	VARCHAR(255)	

Item_price	Price of item	VARCHAR(255)	
Item_status	Status of item	VARCHAR(255)	
storageReference	Image of item	VARCHAR(255)	
Item_order	Order of item	VARCHAR(255)	

Table 3.13 Data Dictionary-Watchlist

Field Name	Description	Data Type	Constraint
watchList_ID	ID of watch list	VARCHAR(255)	PK
Item_ID	ID of item	VARCHAR(255)	FK
User_ID	ID of user	VARCHAR(255)	FK

3.6 Proof of Initial Concept

3.6.1 Manage Registration

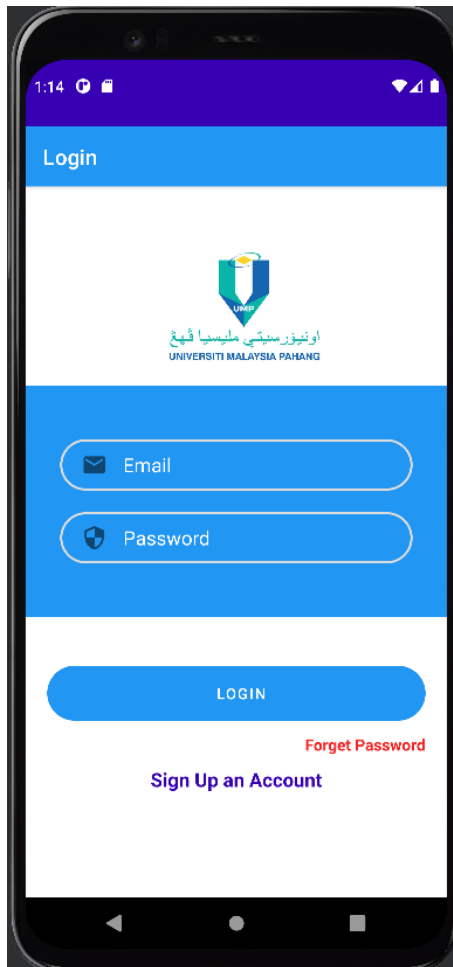


Figure 3.16 Login interface

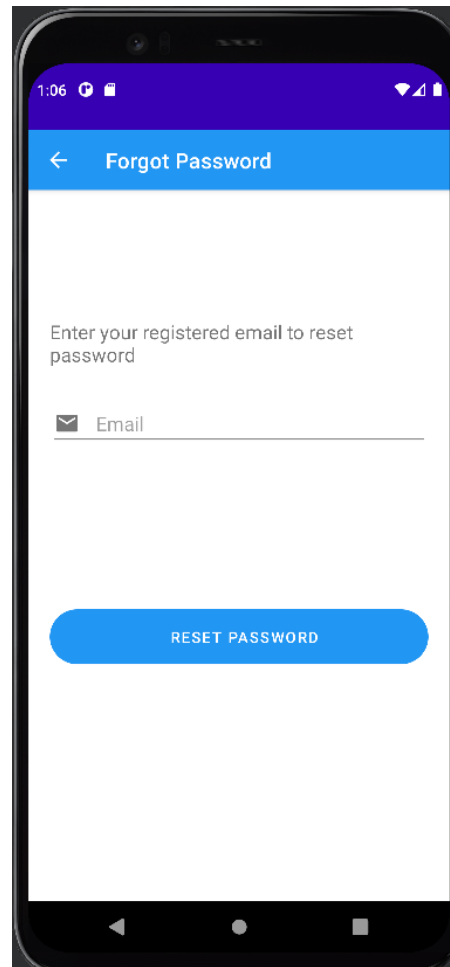


Figure 3.17 Forgot Password interface

Figure 3.16 shows the Login interface which is the first interface that will be seen by the users when entering the application. After clicking the “Forget Password” text, it will redirect to Forgot Password interface shown in Figure 3.17. Users are required to enter the email that had registered and click the “Reset Password” button to reset the password of the user account.

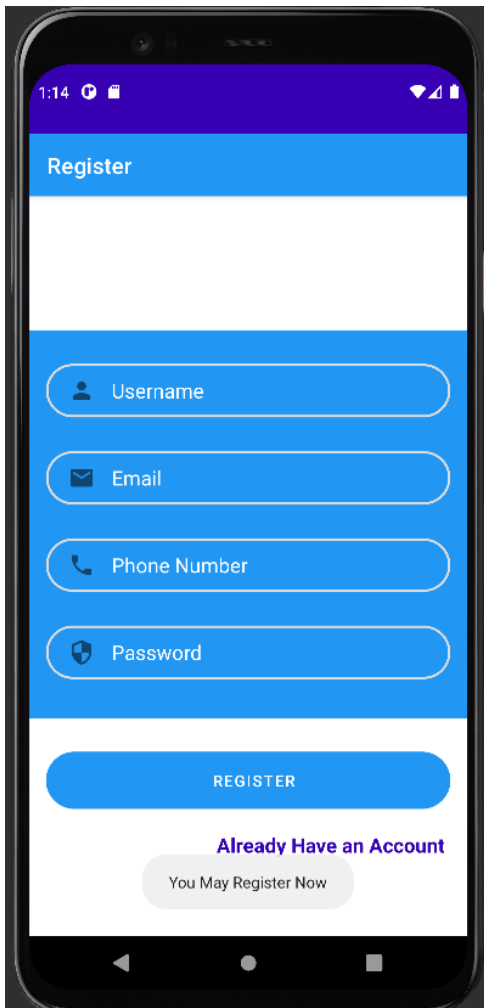


Figure 3.18 Register interface

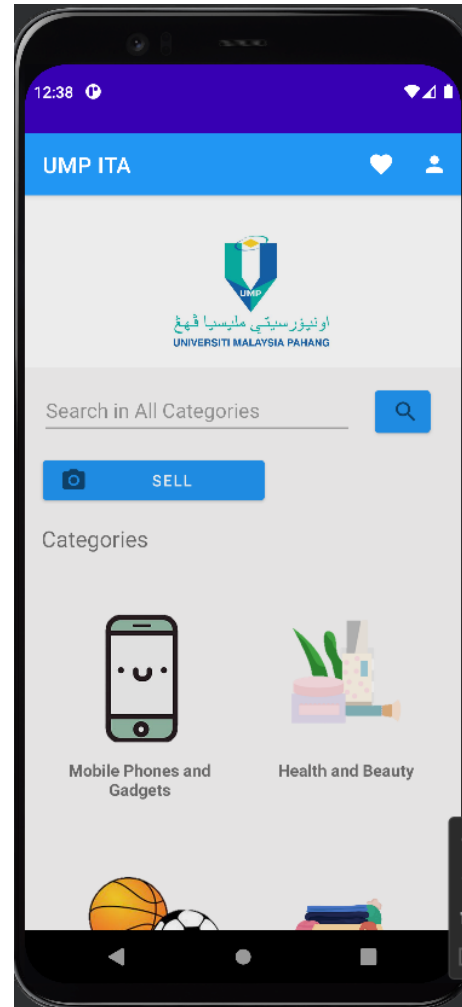


Figure 3.19 Main interface

Figure 3.18 shows the Register interface, the user is required to enter the username, email, phone number, and password to register an account. After entering all the required information, the user can click the Register button to register an account. If the user clicks the Already Have an Account text it will link to Figure 3.16 Login interface. Figure 3.19 shows the main interface after user logged in to the application.

3.6.2 Manage Profile

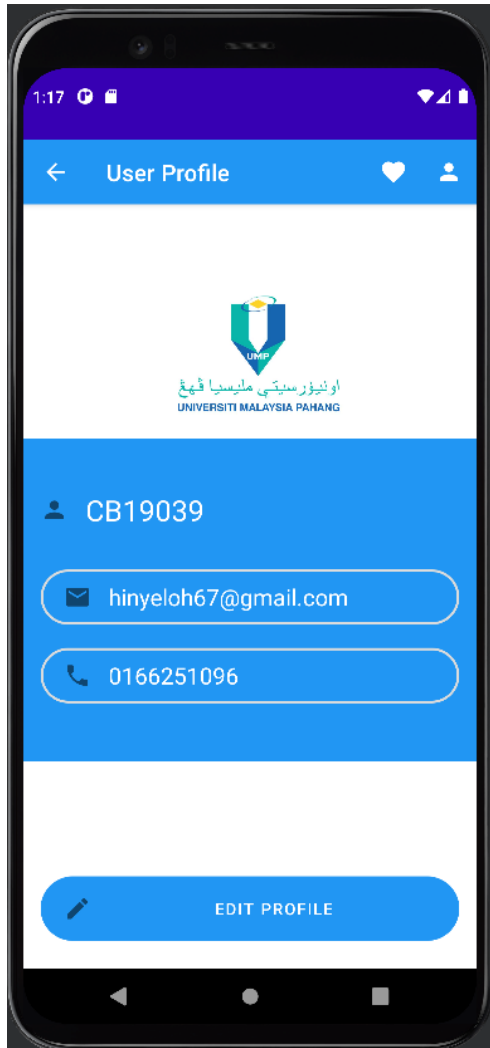


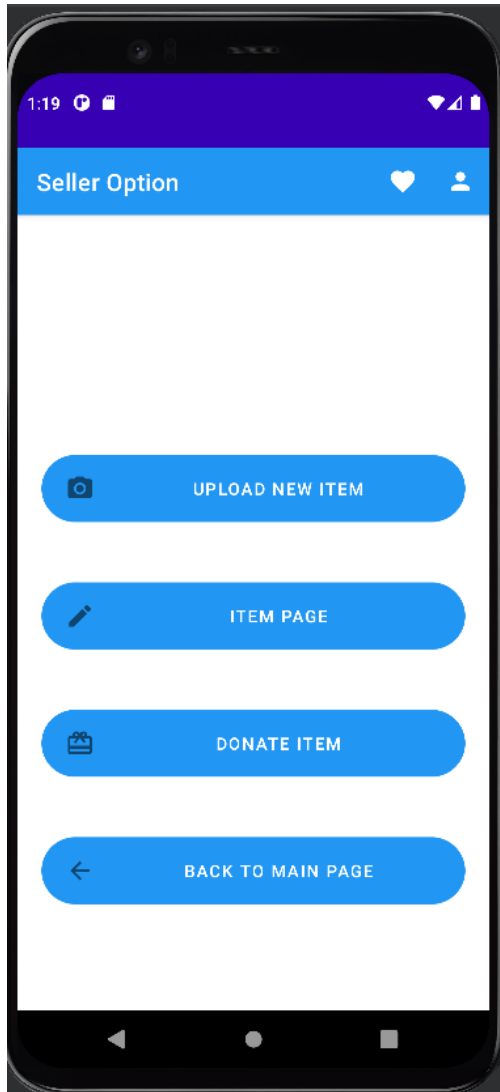


Figure 3.20 Profile interface

Figure 3.21 Edit Profile interface

Figure 3.20 shows the profile interface of the application which will display the user's profile information. If clicking "Edit Profile" button it will directly link to Figure 3.21 Edit Profile interface which able users to edit their profile information. If clicking the "Update" button, it will update the profile information and redirect to Figure 3.20 Profile interface.

3.6.3 Manage Goods



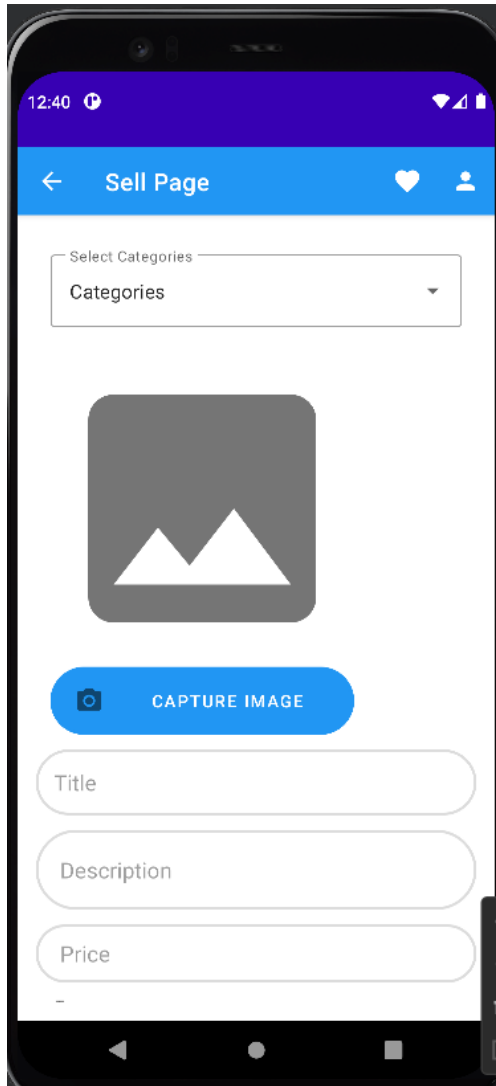


Figure 3.22 Sell Option interface

Figure 3.23 Upload Item interface

Figure 3.22 shows the sell option interface which is allow user to upload new item and view the item page. If clicking the “Upload New Item” button in will redirect to Figure 3.23 Upload Item interface. The user is required to enter item information and capture pictures or select pictures from the gallery. After entering all the information, the user can click the Upload button to upload the item information to the database. If clicking the “Item Page” it will redirect to Figure 3.24 Item Page which able to view the item that uploaded by the seller. If clicking the “Donate Item”, it will redirect to Figure 3.26 Donate Page which able seller to donate their item to someone that needed.

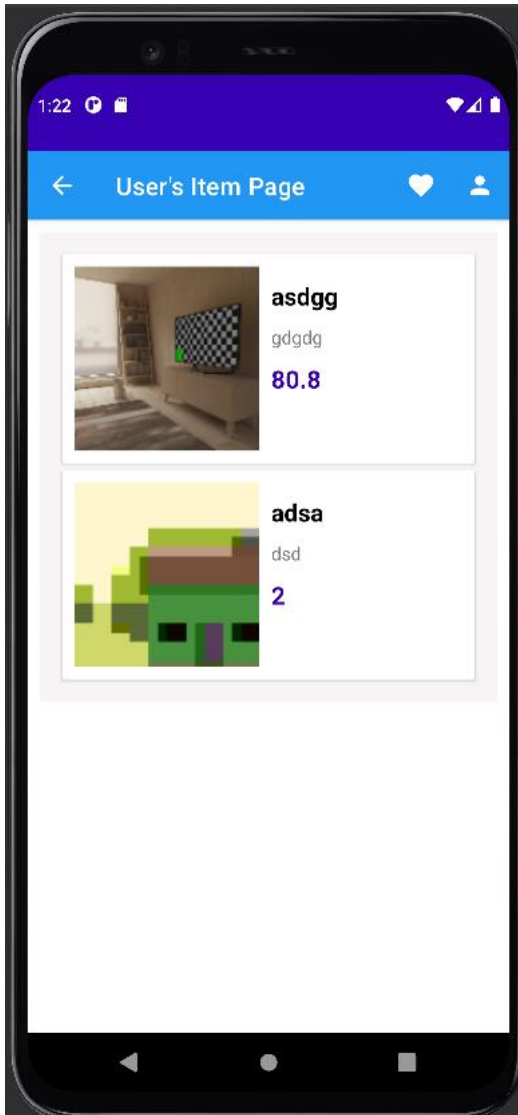


Figure 3.24 Item interface

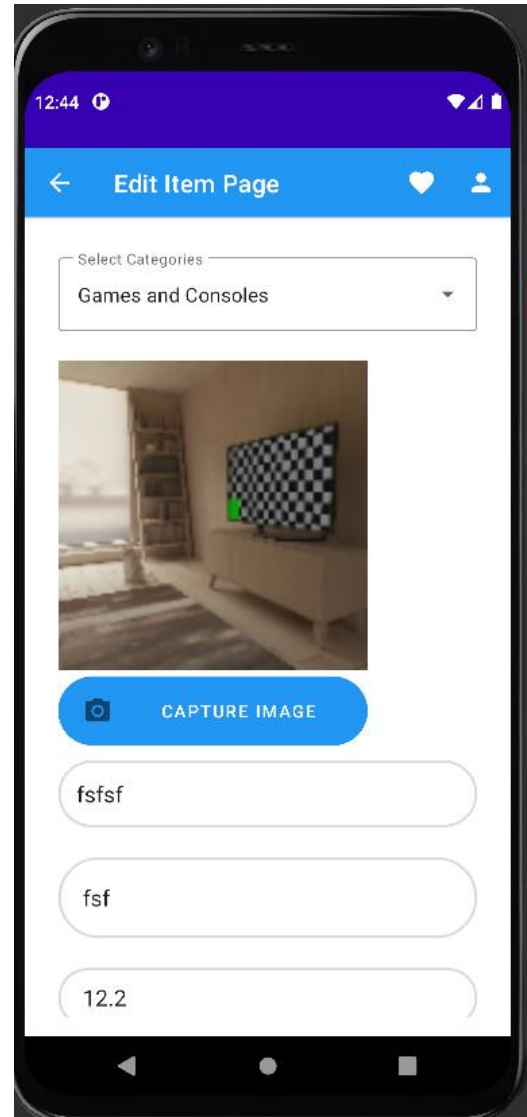


Figure 3.25 Edit item interface

Figure 3.24 shows the item interface which allow users to view the item that had been uploaded by users and users able to delete and edit the item. User can click the item to edit item information and will link to Figure 3.25 Edit item interface. Users can edit the status of the items. Figure 3.25 is required user to enter item information and capture pictures or select pictures from the gallery. After entering all the information, the user can click the Update button to update the item information in the database. Users can delete the item by clicking the Delete button.

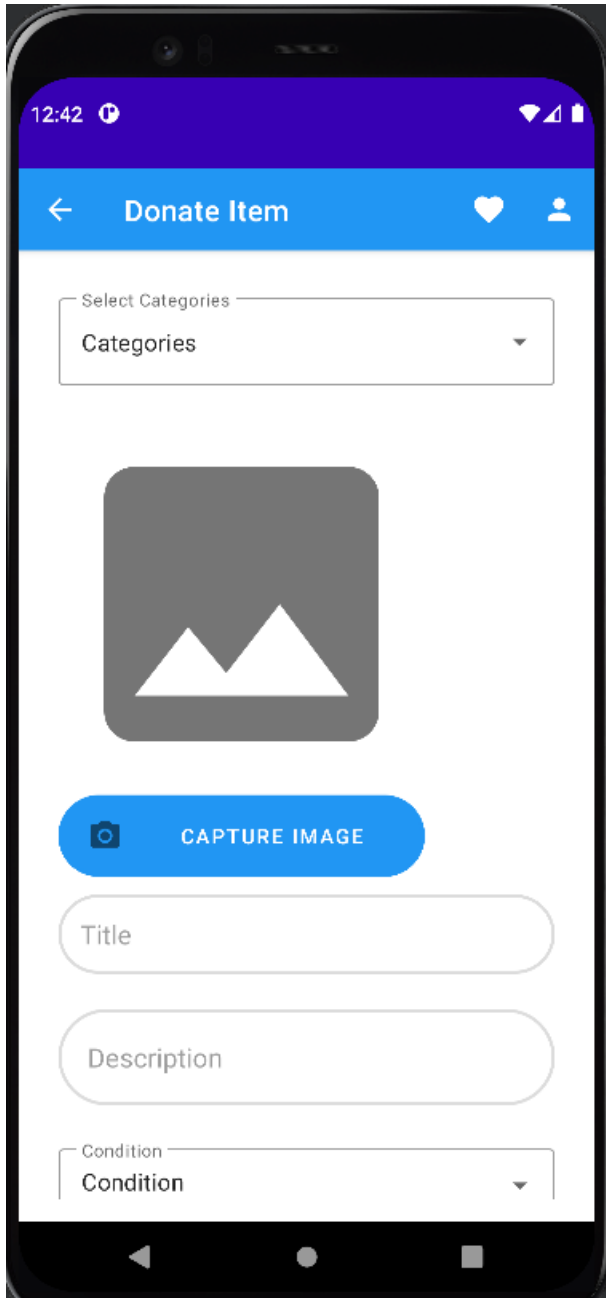
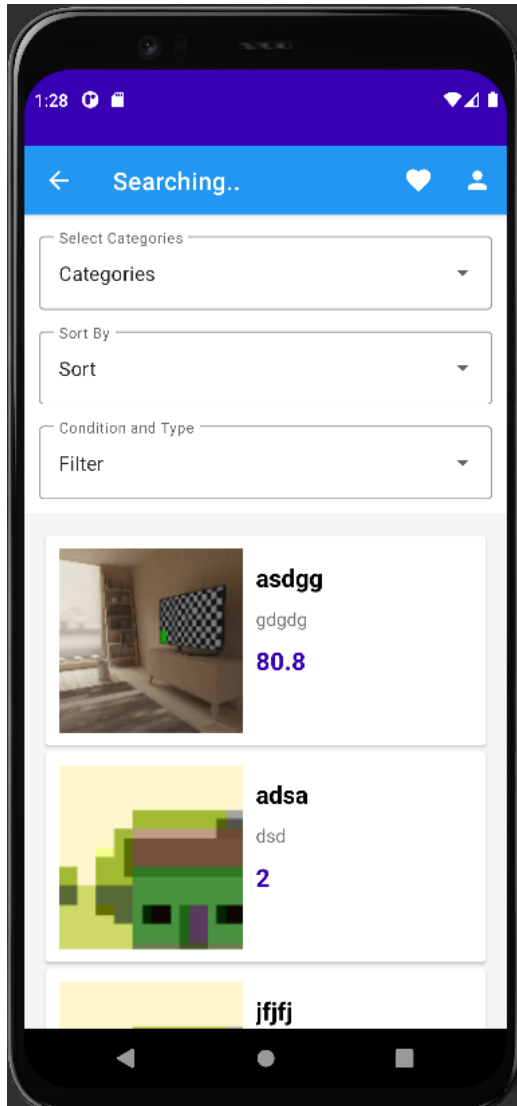


Figure 3.26 Donate Page

Figure 3.26 shows the donate page after seller clicking the “Donate Item” button from Figure 3.22 Sell Option interface. Seller are require to enter the require data to donate the item.

3.6.4 Manage Order



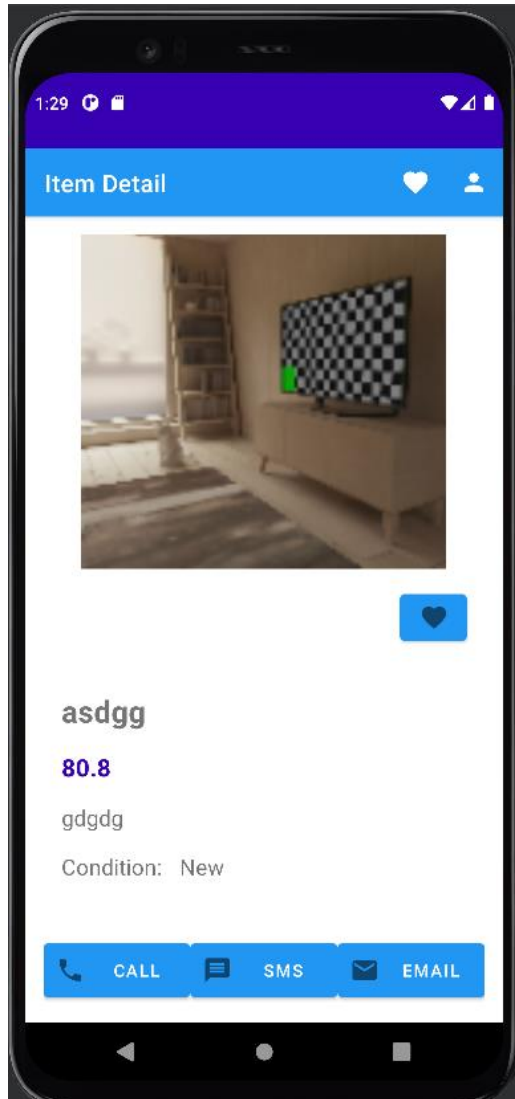


Figure 3.27 Search interface

Figure 3.28 Item Detail interface

Figure 3.27 shows the search interface which are user to filter and sort the item. Users can filter the item by selecting the category, condition, and type. Users also can sort the item by selecting the sort. Users can click the image and it will directly link to Figure 3.28 Item Detail interface. Users can click the star button to add the item to the watchlist and it will display in Figure 3.29 Watch List interface. Figure 3.28 shows the item detail interface which allow users to select the method for contacting sellers like call, SMS, and email. Users can view all the information of the item that selected by user.

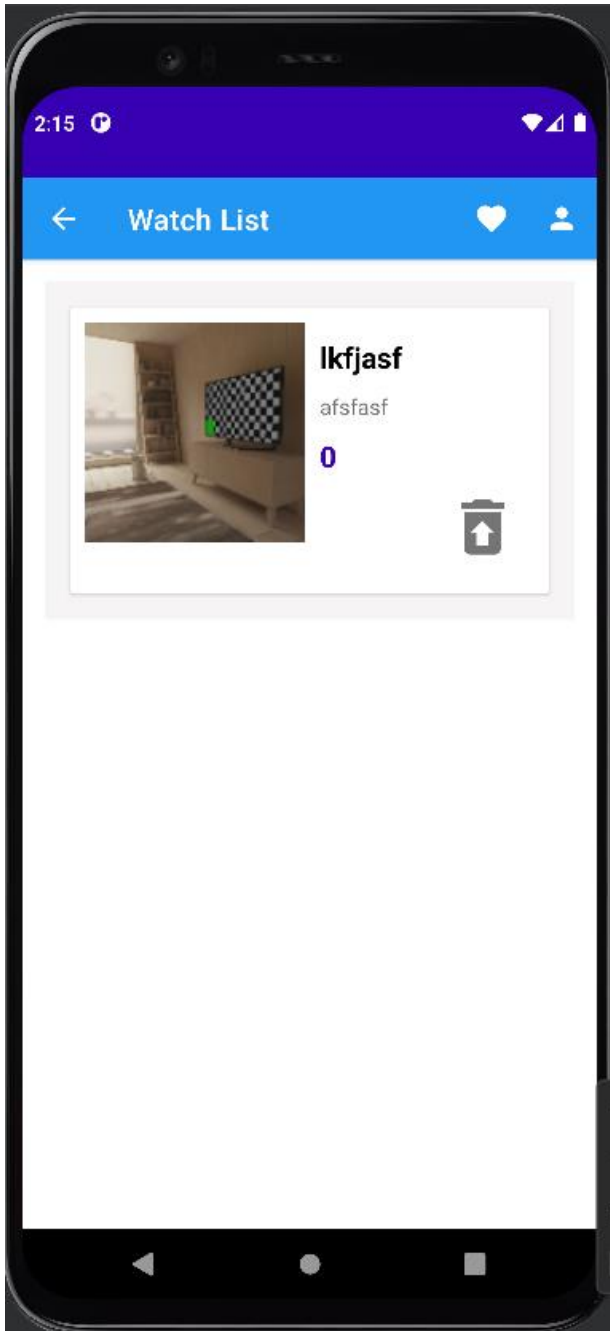


Figure 3.29 Watch List interface

Figure 3.29 shows the watch list interface, Users can view all the items that were added to the watch list. Users can remove the item from the watch list by clicking the remove button. Users can click the image and link to Figure 3.28 Item Detail Page.

3.6.5 Manage User

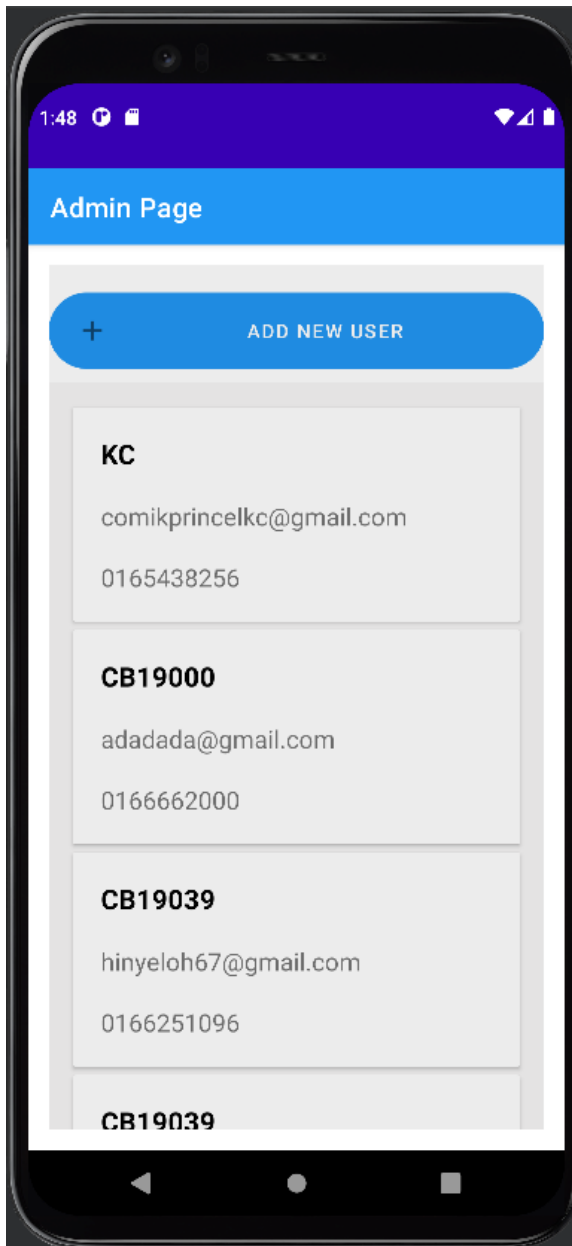


Figure 3.30 Admin Page Interface

Figure 3.30 shows the Admin Page interface, admin can view all the user information that were registered in this application. Admin can view, add, edit, and remove the user information. When the admin clicks the Add New User button, it will redirect to Figure 3.31 Add new user. If the admin clicks the edit button it will redirect to the Figure 3.32 Edit User and the admin click the delete button to remove the user from the database.

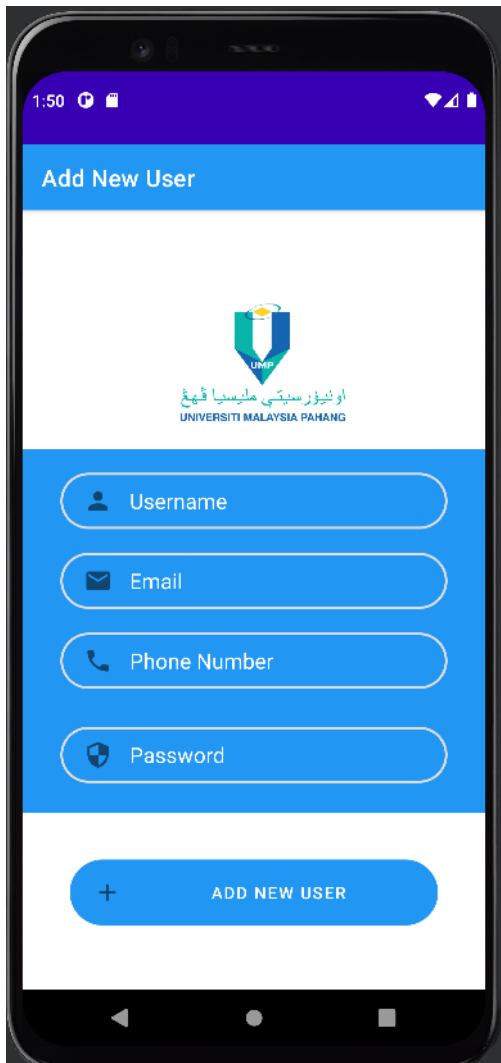


Figure 3.31 Add new user Interface

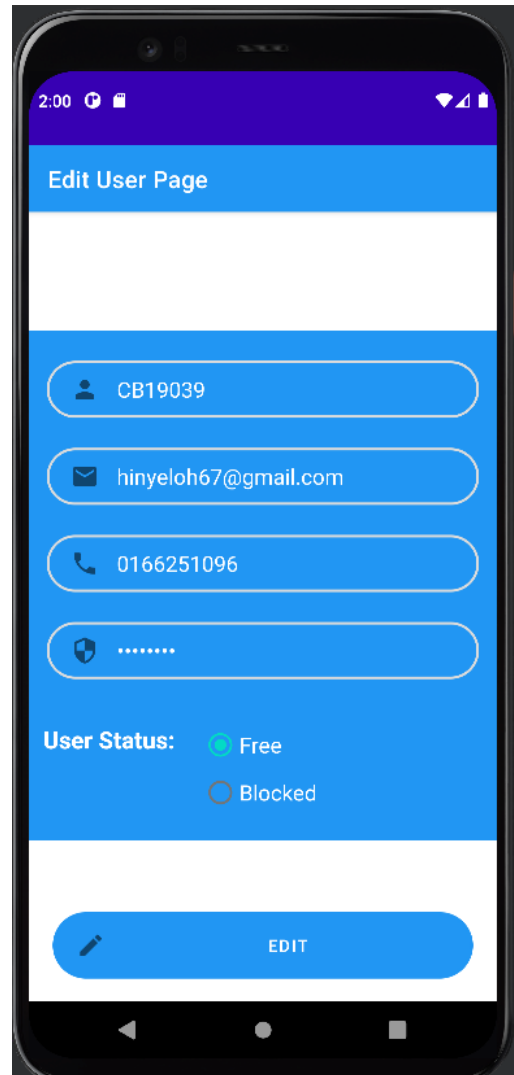


Figure 3.32 Edit user interface

Figure 3.31 shows the add new user interface that required admin to enter the user information to add a new user to system. After admin click the add new user button it will add the user information that had entered and redirect to the Figure 3.30 Admin Page. Figure 3.32 shows the edit user interface that able admin to select user status. After admin click the edit button it will update the user status and will redirect to the Figure 3.30 Admin Page.

3.6.6 Storyboard

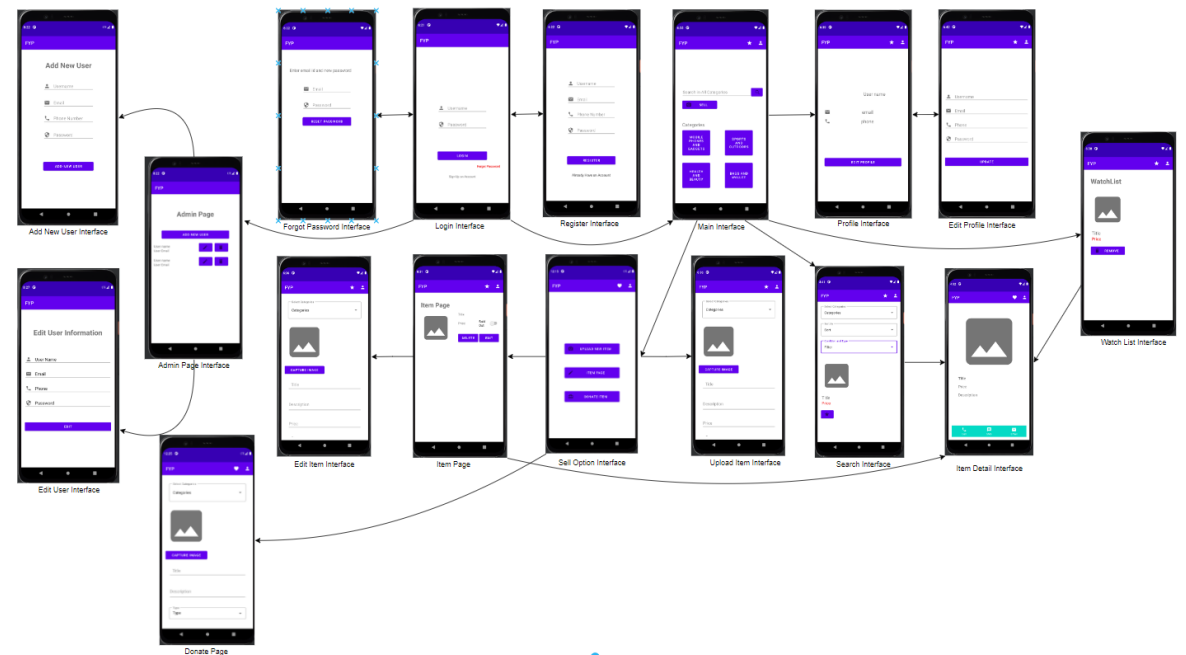


Figure 3.33 Storyboard

3.7 Testing Plan

Following the development of the system, testing should be carried out to uncover any errors or to check that the system fits the requirements. After the application has been launched, a User Acceptance Test (UAT) will be conducted to test the system. The purpose of conducting this UAT is to ensure the developed and deployed system meets the requirement as stated in the Software Requirements Specification (SRS). The testing is also to ensure the system is accepted by the end user.

Table 3.14 UAT form

No.	Module	Activities	Status		Comments
1.	Manage Registration	User login	Yes <input type="checkbox"/>	No <input type="checkbox"/>	

2.		User register	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
3.		User reset password	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
4.	Manage Profile	User views the profile information	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
5.		User edits the profile information.	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
6.	Manage Goods	User open Camera in application	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
7.		User selects category of item	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
8.		User selects type of item	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
9.		User selects condition of item	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
10.		User adds item information	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
11.		User edits item information	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
12.		User views item information	Yes <input type="checkbox"/>	No <input type="checkbox"/>	

13.		User edits the item status	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
14.		User removes item from database	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
15.		User donates the item	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
16.	Manage Order	User searches by the item title	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
17.		User views the searching result	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
18.		User sorts the item	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
19.		User filters the item	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
20.		User selects method for contacting seller	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
21.		User views the item added into the watchlist	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
22.		User removes the item from the watchlist	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
23.		User views the donated item	Yes <input type="checkbox"/>	No <input type="checkbox"/>	

24.	Manage User	Admin views the user information	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
25.		Admin add the user information	Yes <input type="checkbox"/>	No <input type="checkbox"/>	
26.		Admin edit the user information	Yes <input type="checkbox"/>	No <input type="checkbox"/>	

3.8 Potential Use of The Proposed Solution

Although nowadays online shopping or m-commerce is very common and popular there have some specific items or products like the used books in some specific courses. There has some application that was released for the used item like Mudah.my, Carousell, and Lelong.my but this application is specific for the UMP pekan student to trade their item or product included used item. This application can be used by UMP student to trade the item among students. Inside UMP there have a lot of people has many used books or some used items that are transferable and want to sell to juniors to reduce the waste of resources, and juniors also can save money based on just using the used items and no need to purchase new items. This application can help the UMP management to manage the idle resources like used bicycles which can rearrange the resource to someone that needed. The user can donate the used item to the junior or someone that needed.

CHAPTER 4

4.1 RESULT AND DISCUSSION

Result and discussion will show the results of the application that has been build which is the UMP Item Trade Application and discuss with the application. It will include the installation and implementation of IDE and database.

4.2 Introduction

This chapter involved the discussion of the development and testing of the functionalities of the proposed android application. Lastly, the User Acceptance Test (UAT) report has also been stated in this chapter.

4.3 Result

The results of the application will be show which include the following information that included the installation of IDE, implementation of database, and implementation of application.

4.3.1 Installation of IDE (Android Studio)

The IDE used to develop the project is Android Studio Bumblebee | 2021.1.1 Patch 2. The IDE is running in Windows 10 while developing and testing the code, through the built-in android emulator.

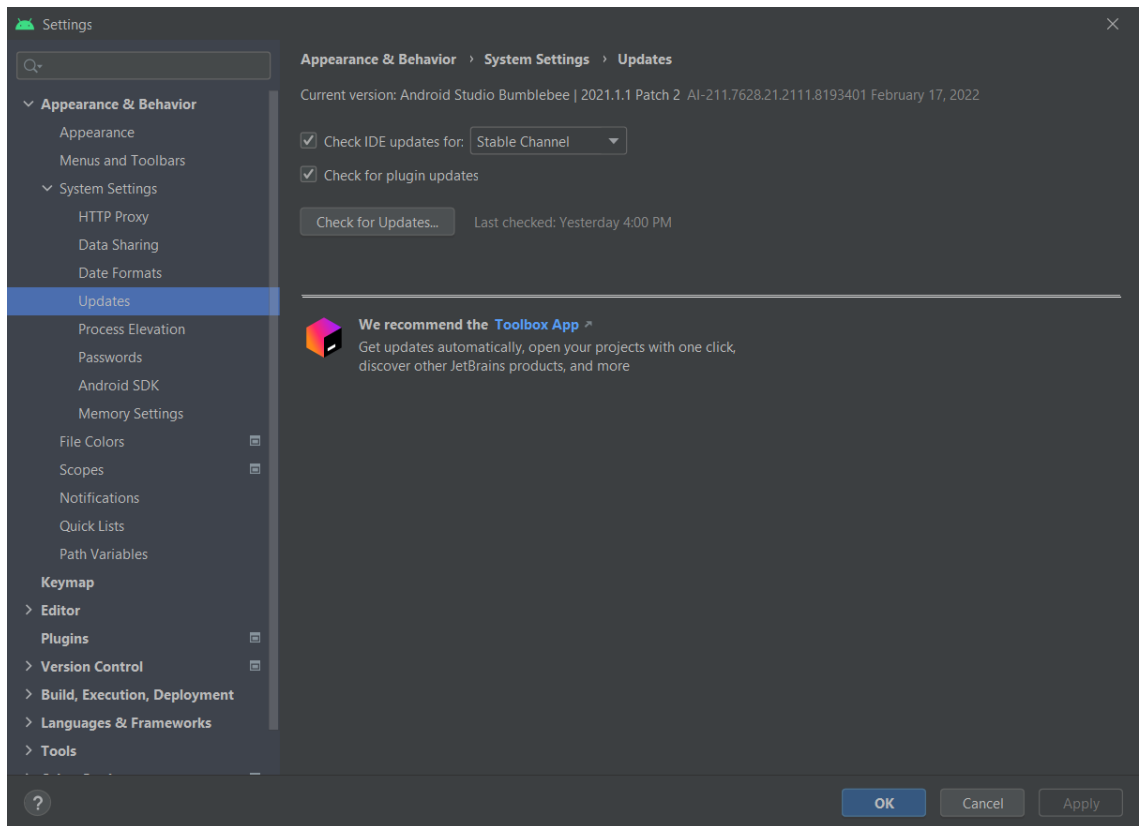


Figure 4.1 Android Studio IDE

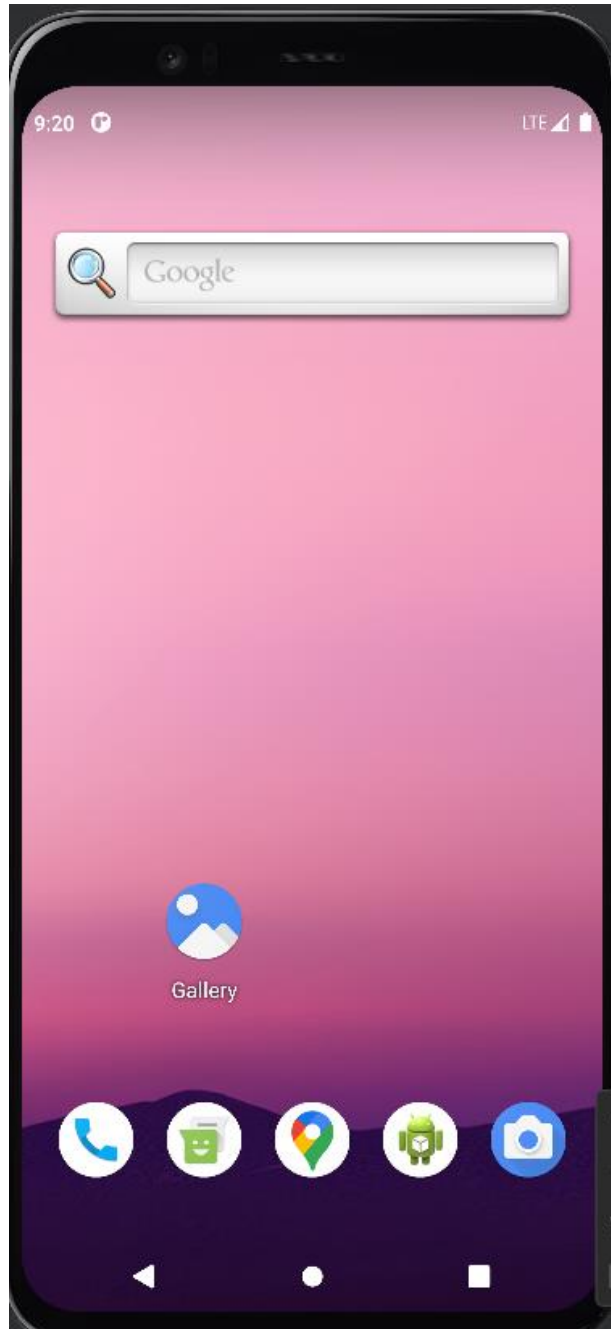


Figure 4.2 Android Studio

4.3.2 Implementation of database

The data storage of the application is implemented using an online open-source database named Firebase. Besides, in the project it was implement the Firebase authentication, Firebase Realtime database and Firebase Storage.

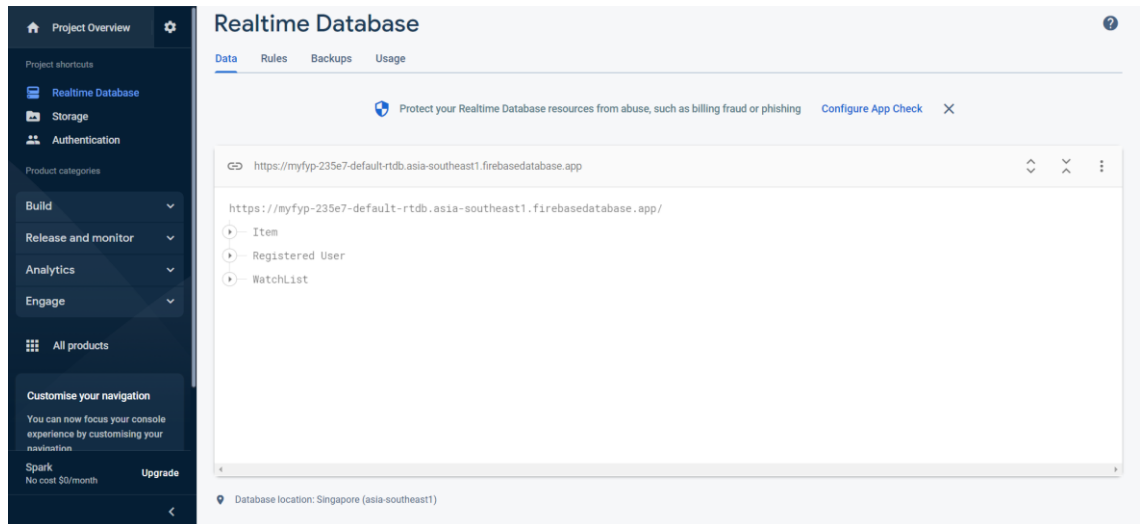


Figure 4.3 Firebase – Realtime Database

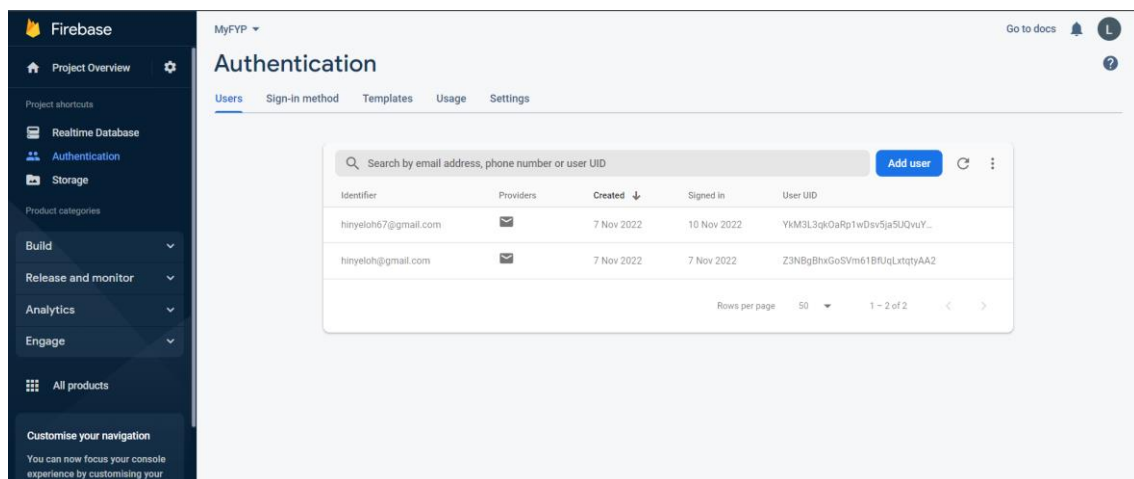


Figure 4.4 Firebase – Authentication Database

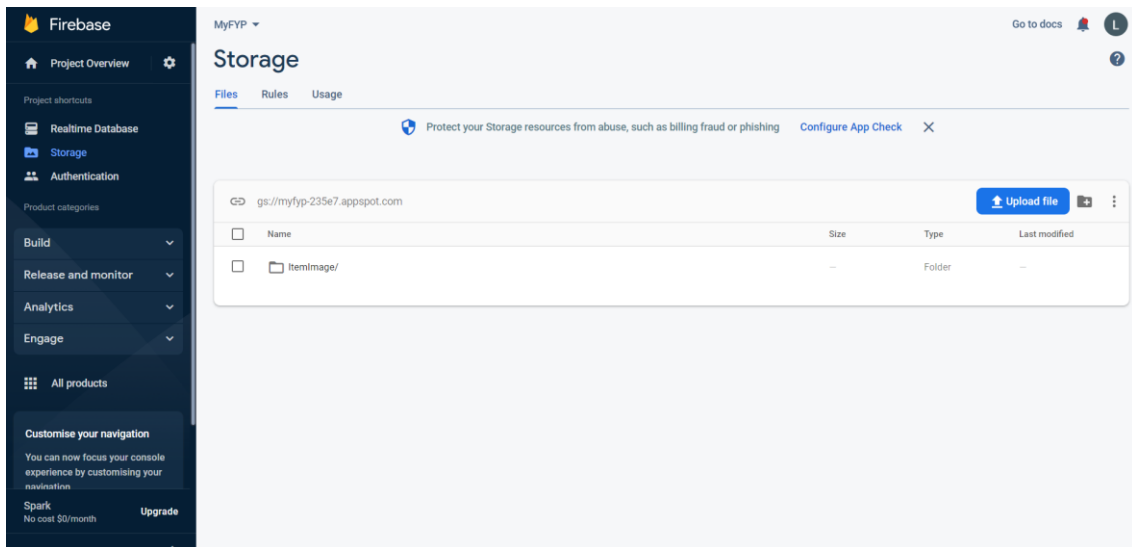


Figure 4.5 Firebase – Storage Database

4.3.3 Implementation of Application

The application is developed using Android Studio which support JAVA and NoSQL database implementation query. The implemented interface will be explained in this section. Below will show the application interface of the application. For the interface, it will separate to admin and user interface.

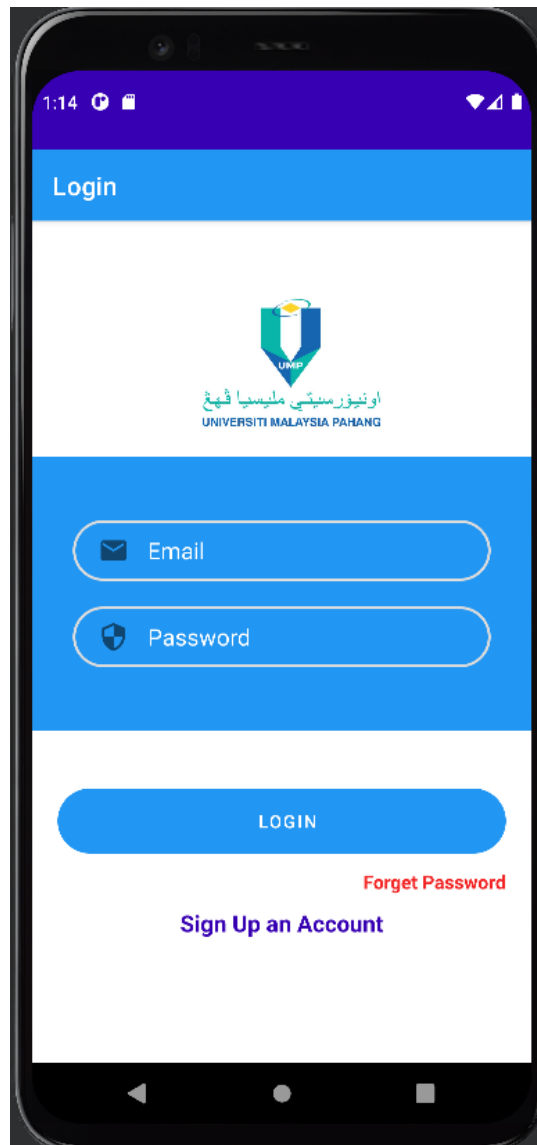


Figure 4.6 Login Interface

Figure 4.6 shows the Login Interface of the project which also the first interface that when open the application. User are required to enter the email and password to login to the system. After clicking the login buttons the system will verify the email and password that had registered in the application. User can click the forget password text to link to Forgot Password Interface to reset the password and click the Sign Up an Account text to redirect to Register Page.

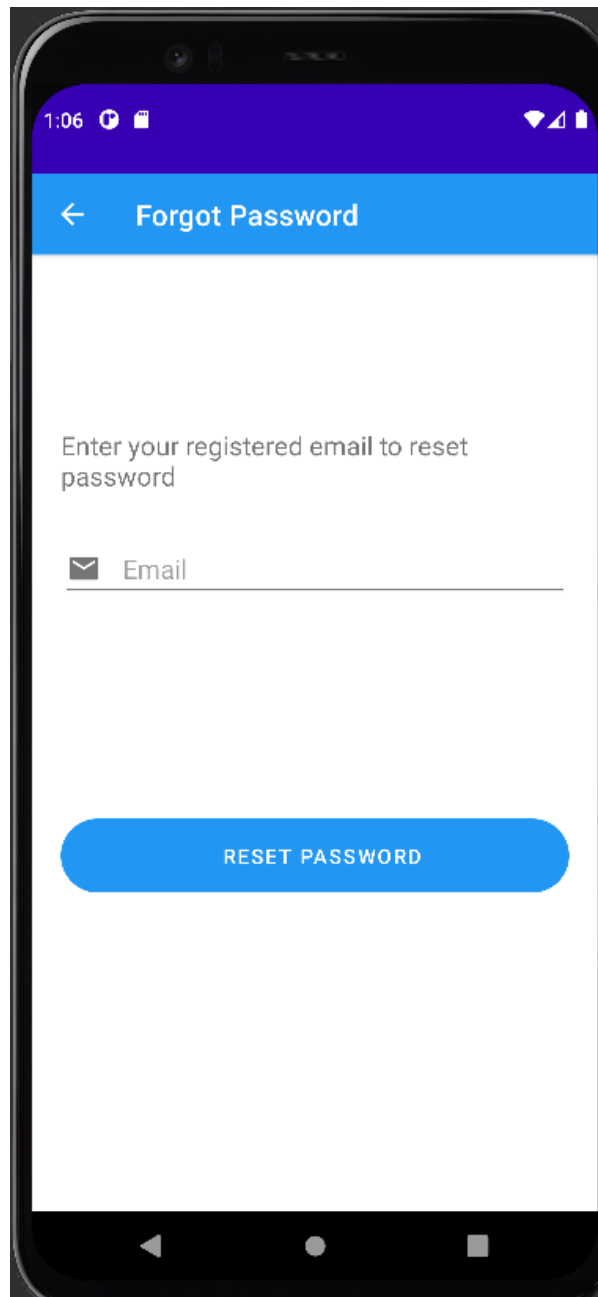


Figure 4.7 Forgot Password Interface

Figure 4.7 shows the Forgot Password Interface which can help user to reset their password. The user that wants to reset their password is required to enter their email and click the Reset Password button. After clicking the button the application will send a link to email for the resetting the password in the application.

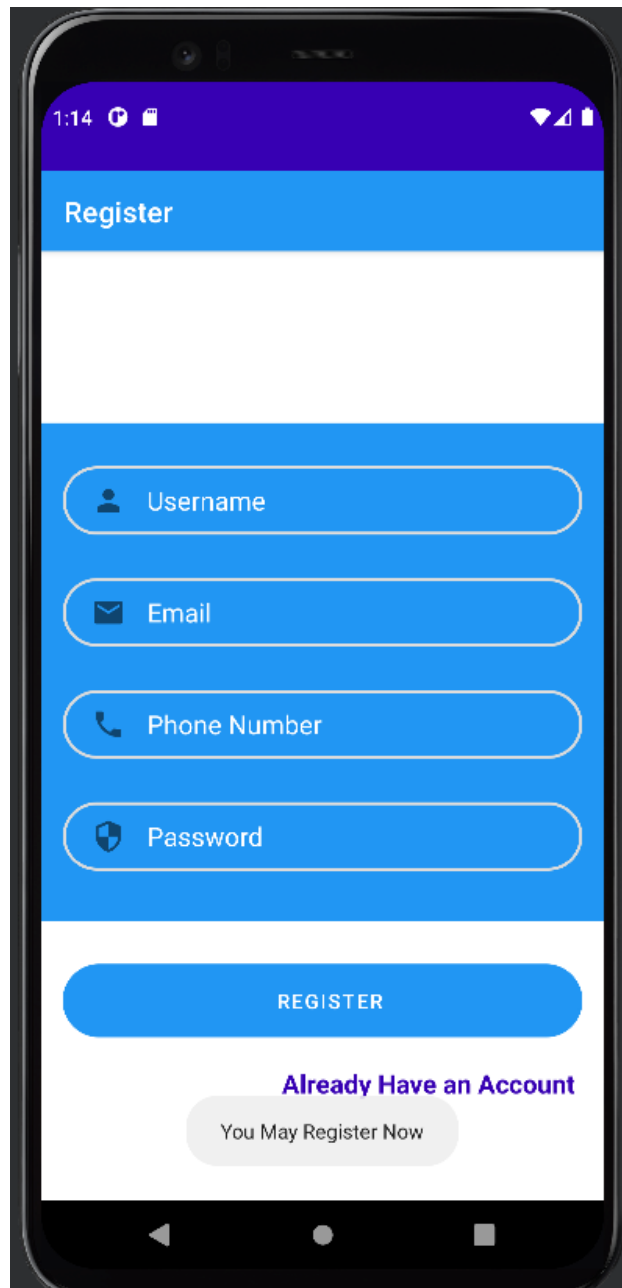


Figure 4.8 Register Interface

Figure 4.8 shows the Register Interface which allow user to register an account in the application. Users are required to enter username, email, phone number and password to register an account. After entering all the required information, clicks the register button to complete the registration flow and the system will verifying the information that entered. If all the information is valid the interface will redirect to Login Interface. User can click the Already Have an Account text to redirect to Login Interface.

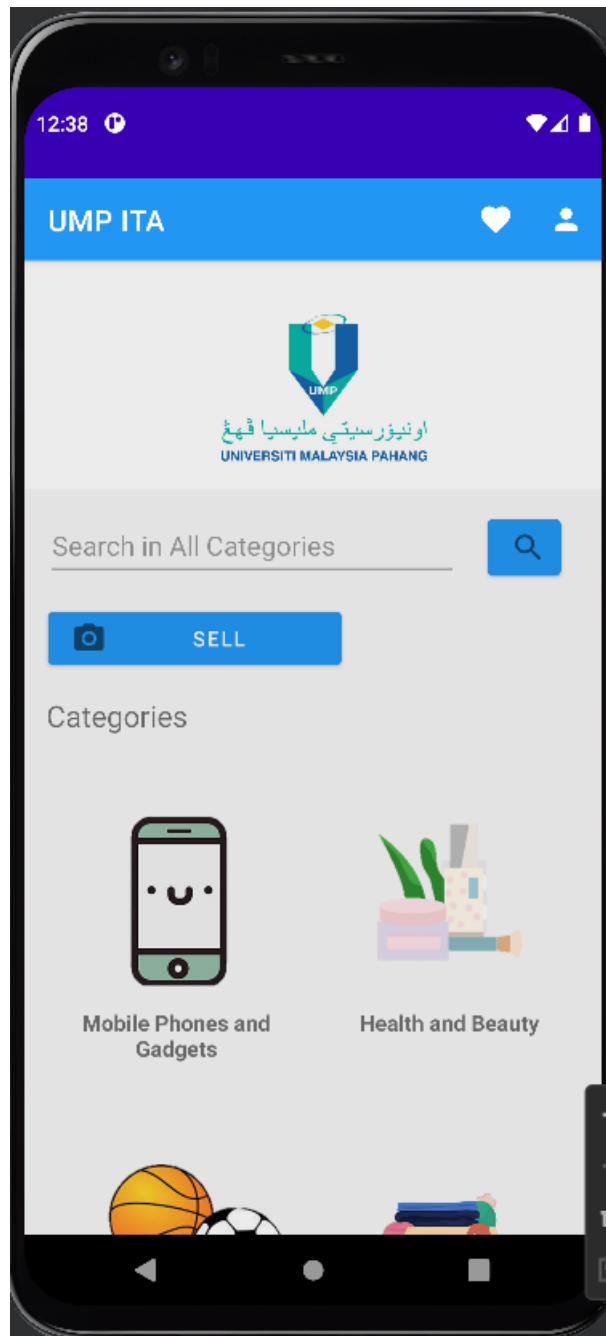


Figure 4.9 Main Interface

Figure 4.9 shows the main page of the application, user can redirect to seller page by clicking the sell button and redirect to search page by clicking the search button. After clicking the search button the system will read the text that the user want to search and if the search text is empty it will display all the item that had record in the database. User can directly search the item by click the categories button. User can sell their item by clicking the sell button to redirect to seller option page.

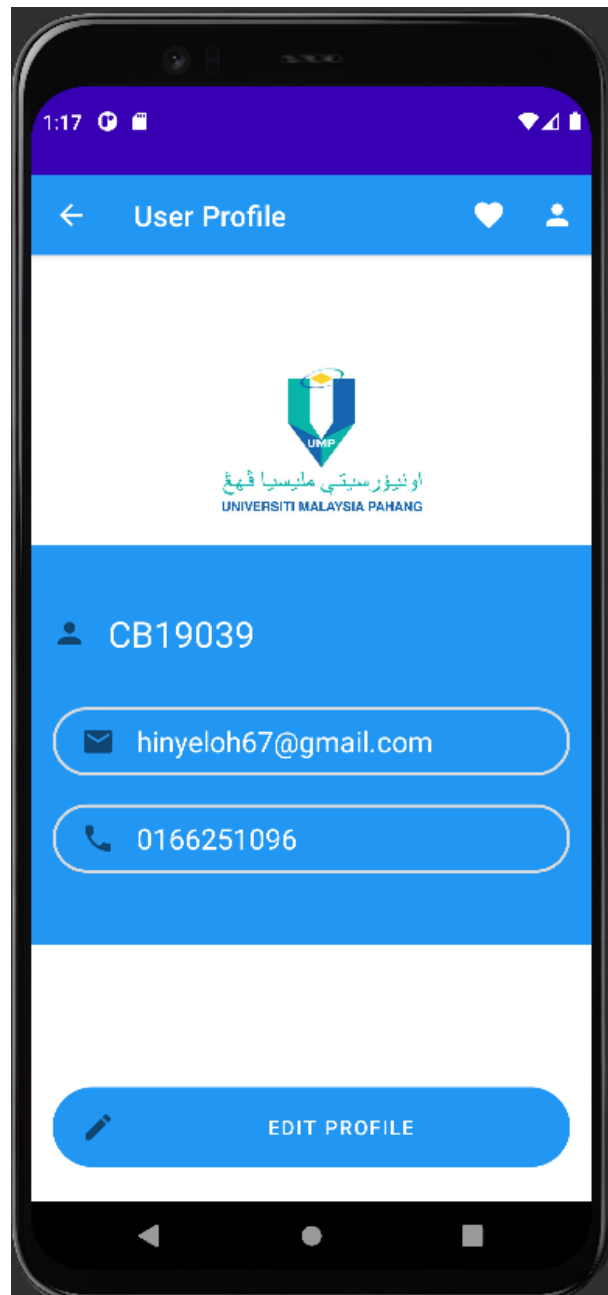


Figure 4.10 User Profile Interface

Figure 4.10 shows the User Profile Interface which will display profile information of the current user that had login to the application. The system will display the username, phone number and email of the user. The user can click the edit profile button to redirect to the edit profile interface.

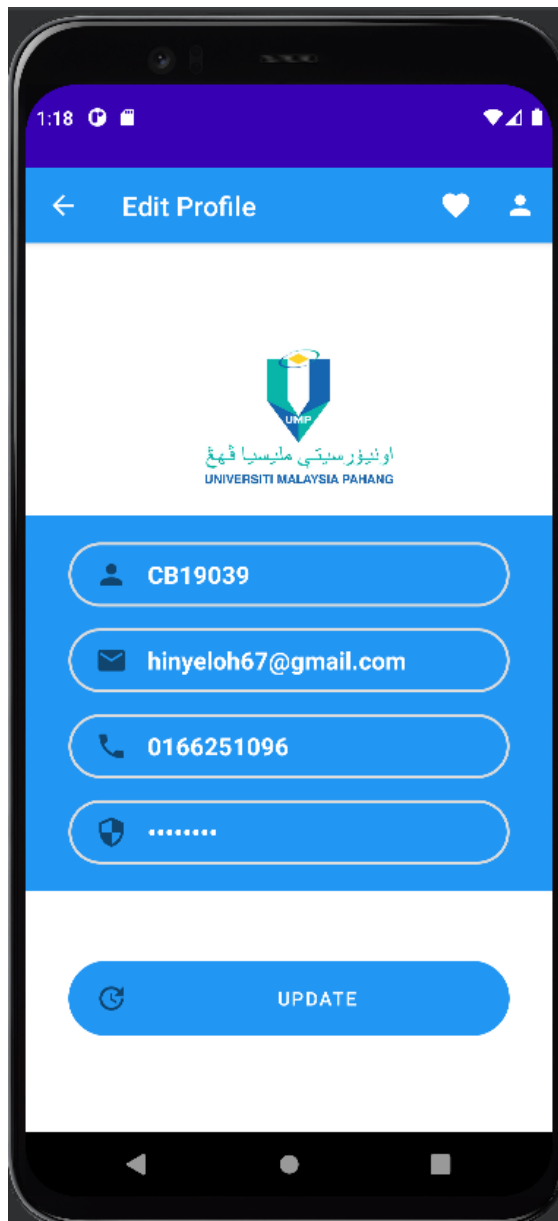


Figure 4.11 Edit Profile Interface

Figure 4.11 shows the user profile information for the current user and able user to update the information of the user. The system will fetch all the user information and set to the edit text view. Users can click the update button to update the user information.

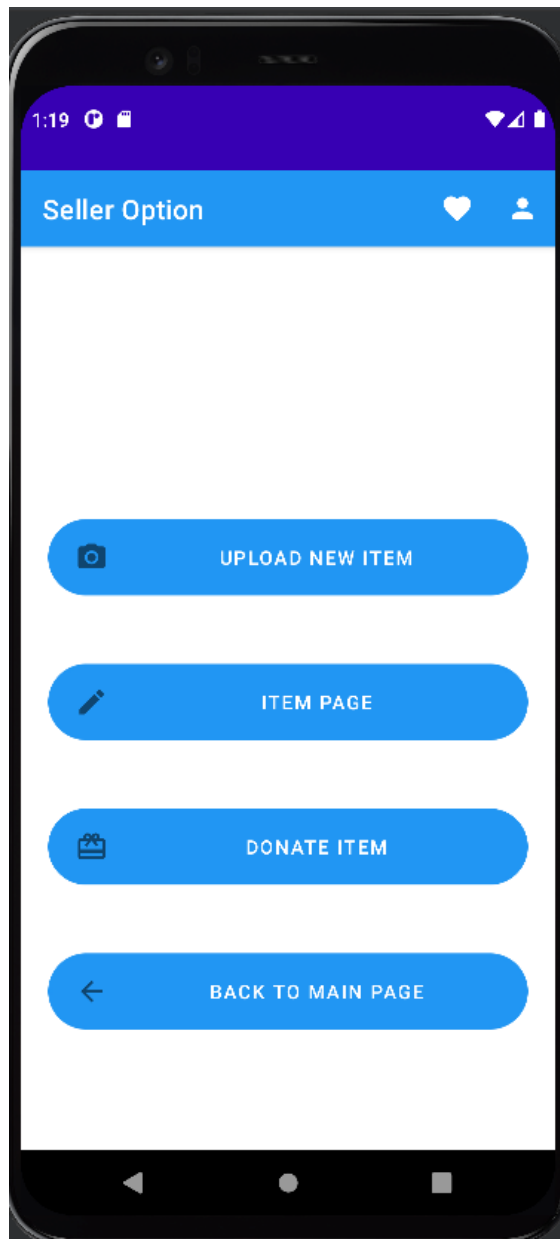


Figure 4.12 Seller Option Interface

Figure 4.12 shows the Seller option interface, in this interface there have four button which is upload new item, item page, donate item and back to main page. After clicking the upload new item button, it will redirect to sell page. After clicking item page button, it will redirect to item page. After clicking the donate item button, it will redirect to donate item page. After clicking the back to main page button, it will redirect to main page of the application.

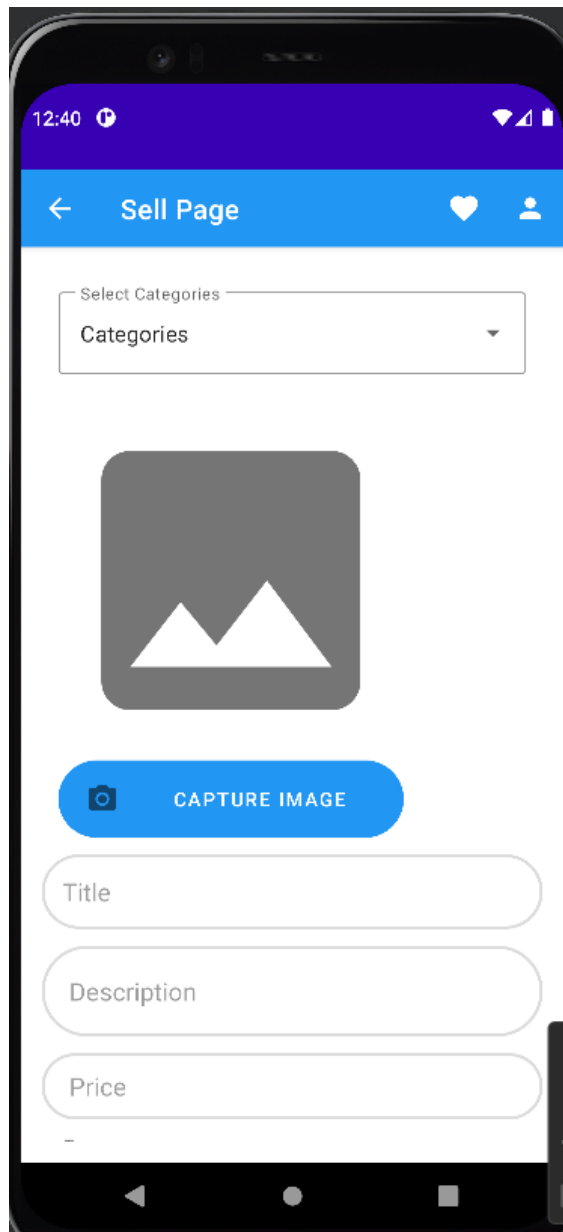


Figure 4.13 Sell Page Interface

Figure 4.13 shows the Sell page Interface which allow user uploads their item to the application. User is required to enter the title, description and price of the item and select the category, image, type and condition of the item. The user also allows to capture the image by clicking the capture image button which allow the device to open the camera. After entering all the required information, the user can click the upload button to upload the item to the application.

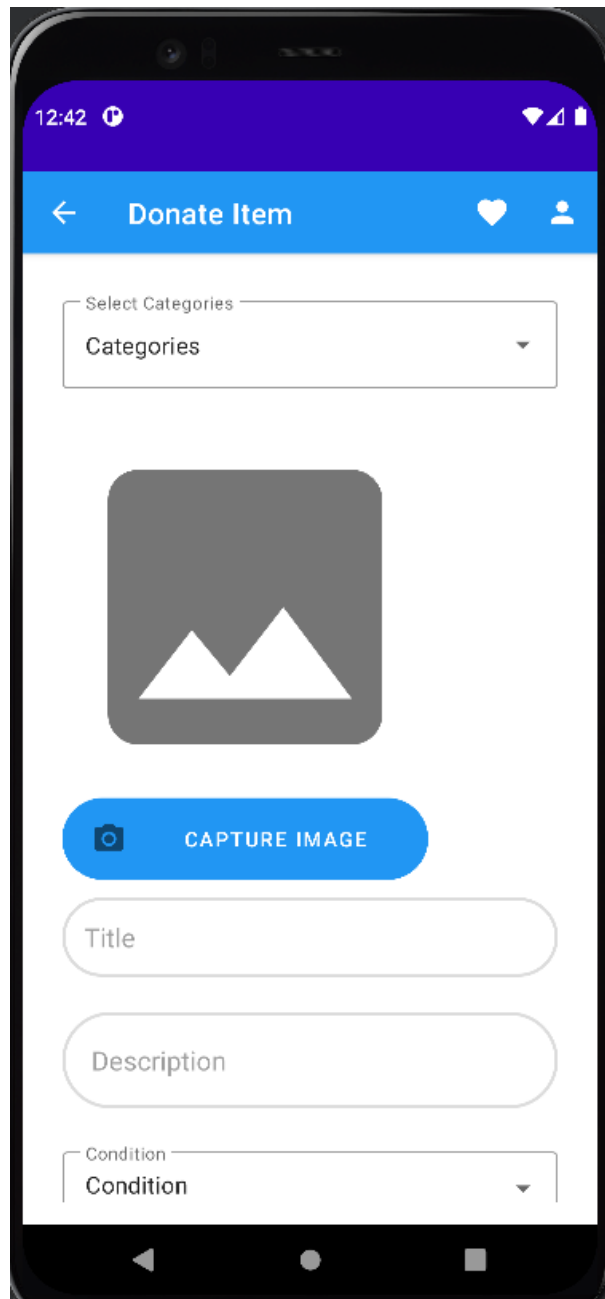


Figure 4.14 Donate Item Interface

Figure 4.14 shows the donate item interface which allow seller to donate their item. User required to enter the title and description and select the category, condition and item image before upload the item to the application.

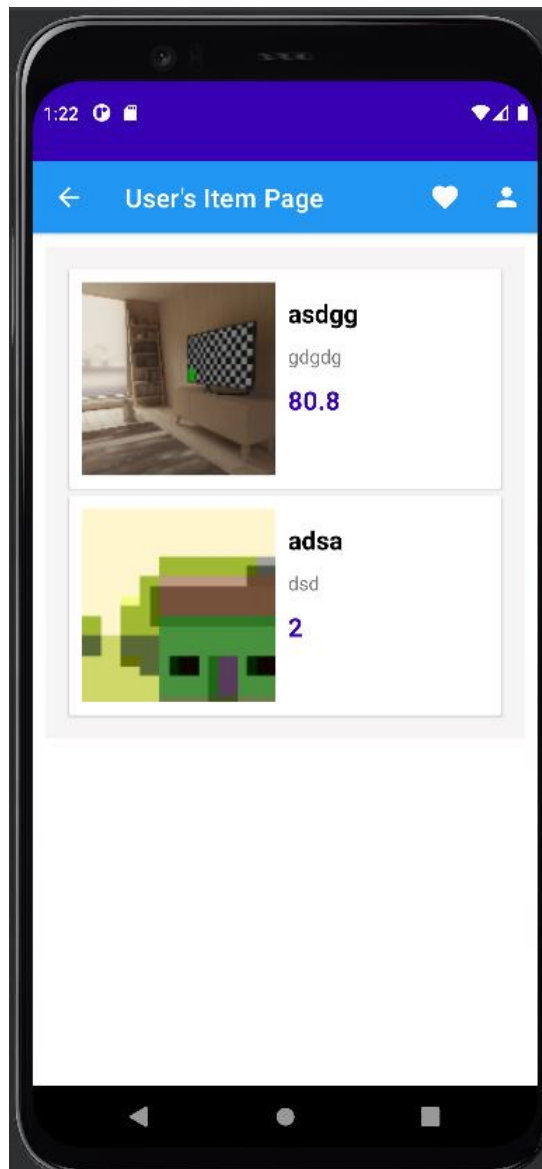


Figure 4.15 User's Item Interface

Figure 4.15 shows user's item interface which will show all the item that uploaded by the current user. The user can click the item card to redirect to the edit item page.

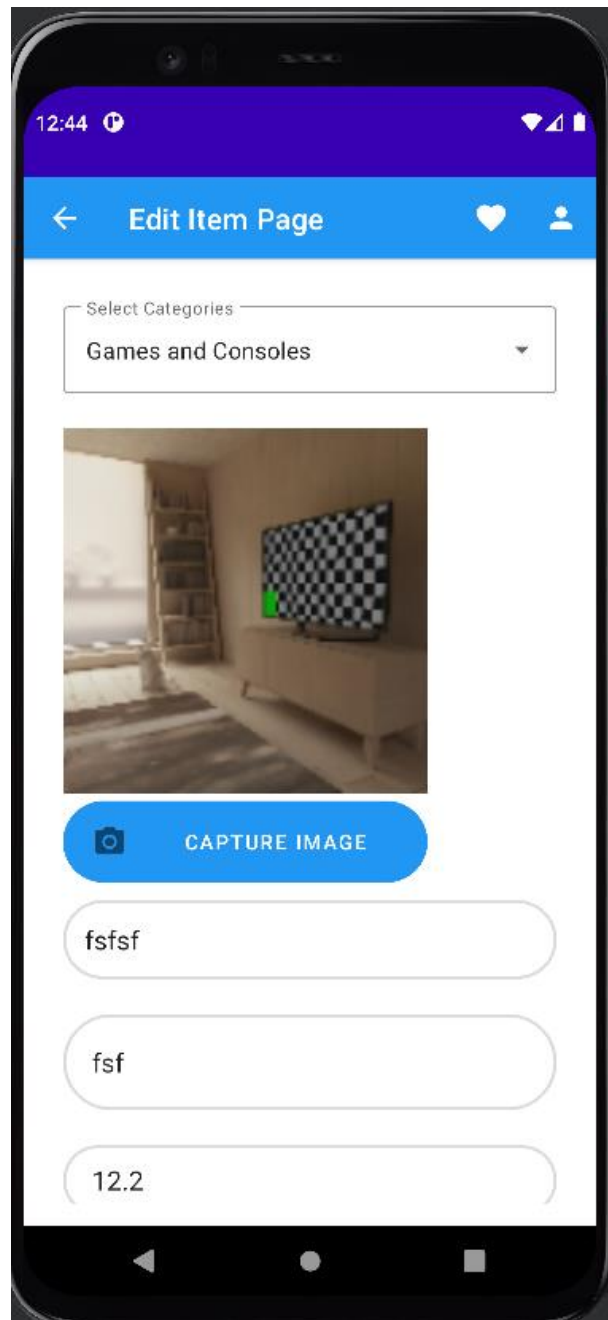


Figure 4.16 Edit Item Interface

Figure 4.16 shows the item interface, the edit item interface will fetch the information that had clicked in item interface. The system will display all the information of the item like category, image, title, description, price, condition, type and status of the item. User can enter the new item information and click the update information to update the item information. User also allow to click the remove item button to remove the item from the application.

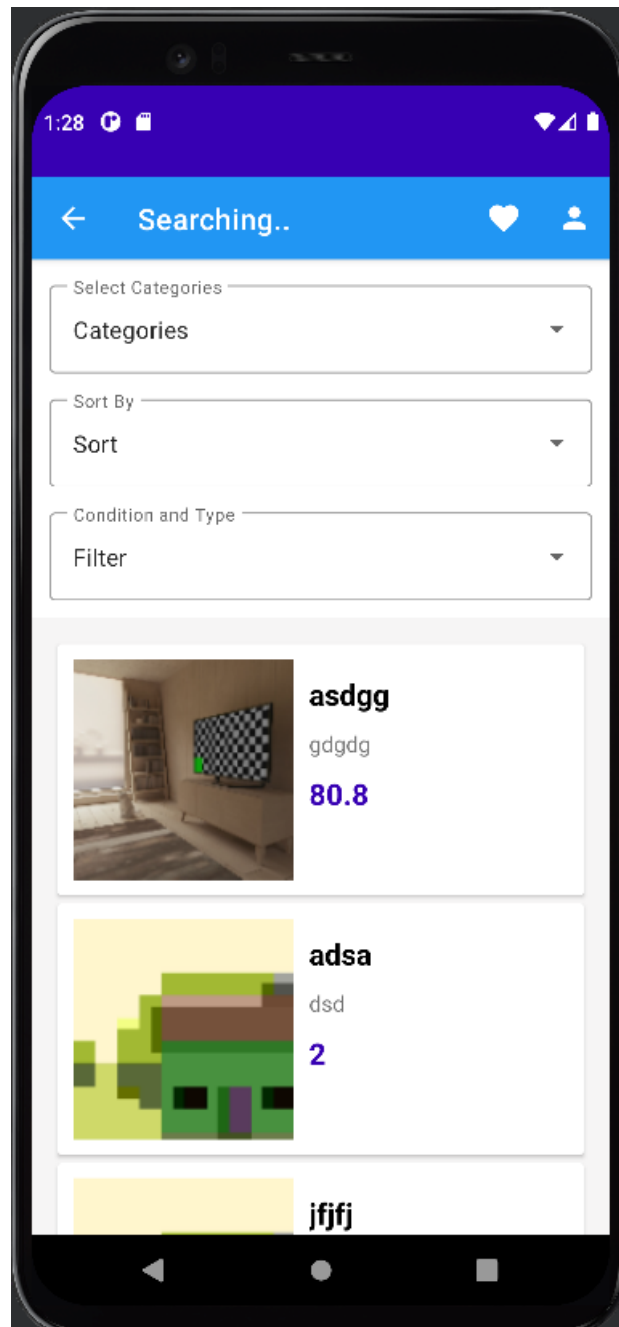


Figure 4.17 Searching Interface

Figure 4.17 shows the searching interface which will display the search interface. The user can filter the searching result by selecting the category, sort and filter. Users are allow to scroll down the searching result and after click the item it will redirect to the item detail interface.

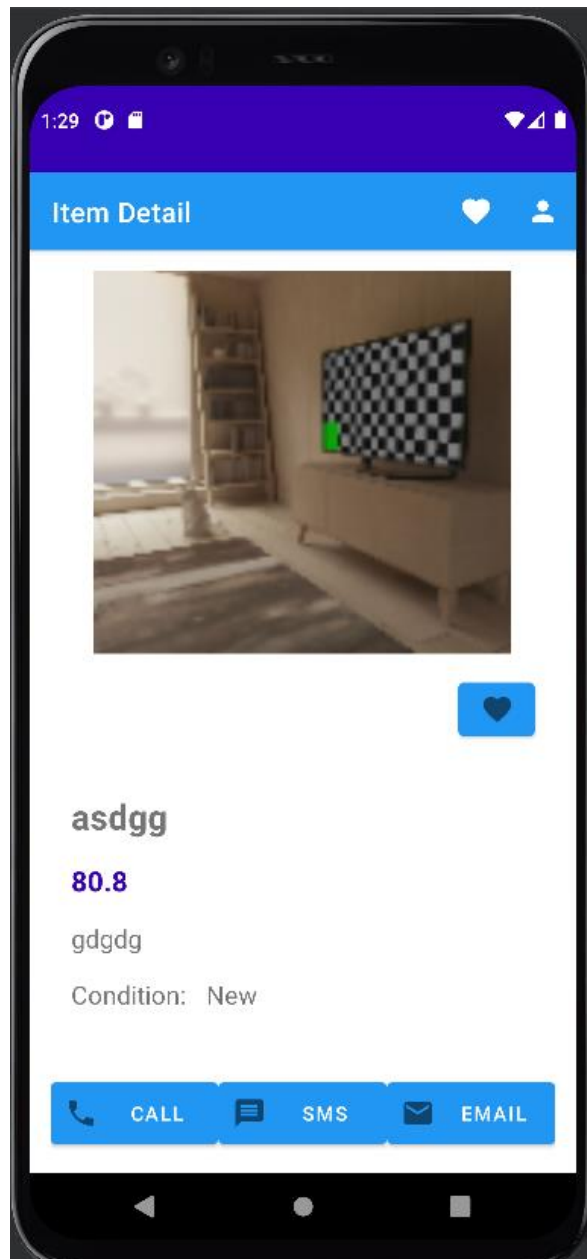


Figure 4.18 Item Detail Interface

Figure 4.18 shows the item detail interface which will display the item information of the item. User can click the favor button to add the item to the watch list. If the users are interest of the item, user can select the method to contact the seller by clicking the button at the bottom which is call, SMS and email.

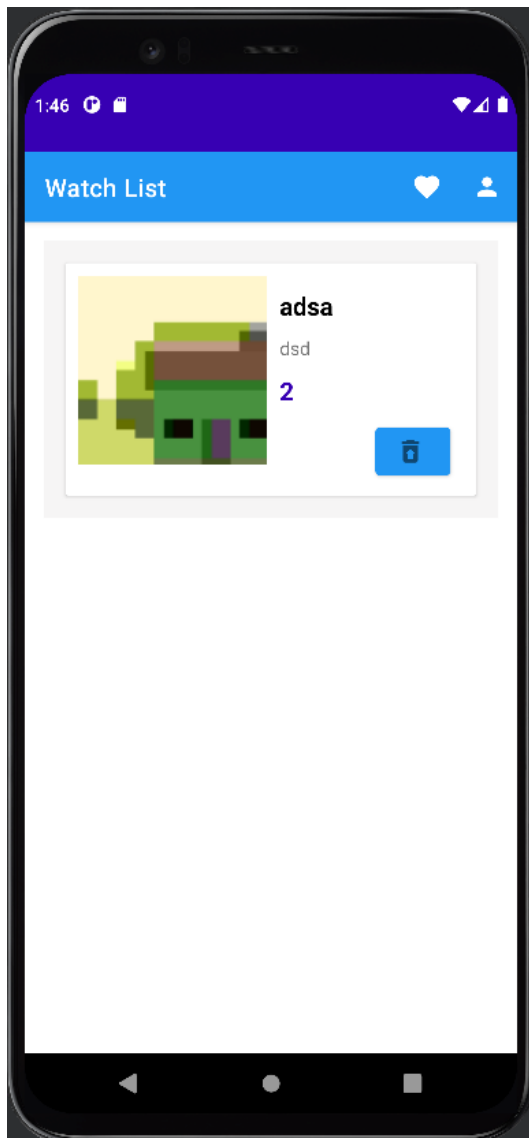


Figure 4.19 Watch List Interface

Figure 4.19 shows the watch list interface which will display the item that had added by the user. User can click the favor button on the action bar and it will redirect to this interface. After clicking the item it will redirect to the item detail page to display the item information.

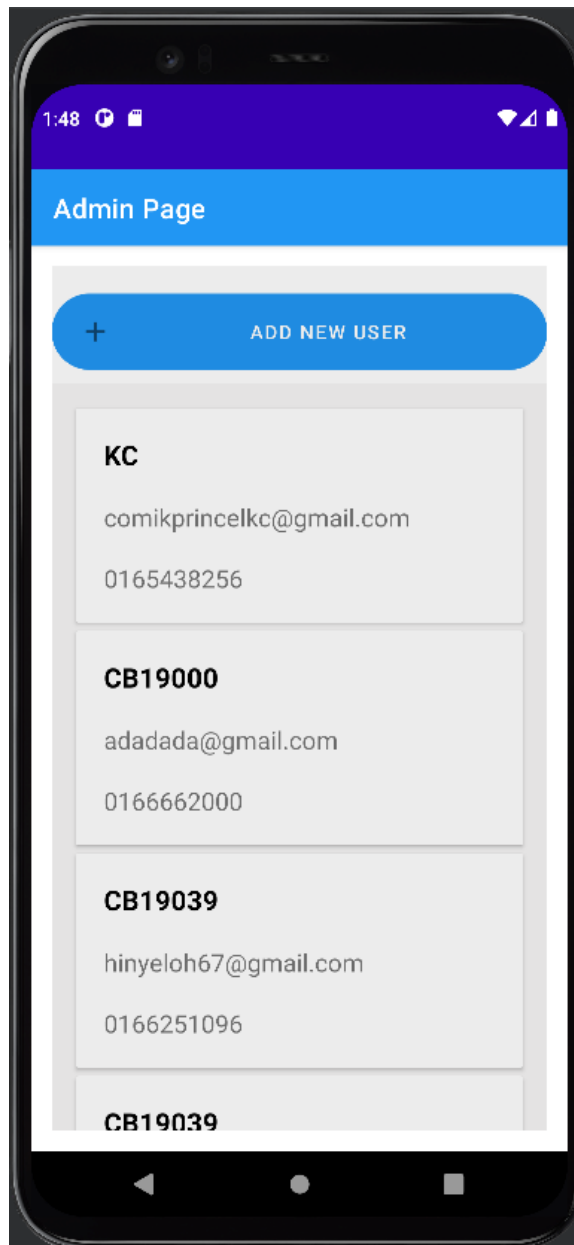


Figure 4.20 Admin Page Interface

Figure 4.20 shows admin page interface which will display all the user in the database. The admin can add a new user by clicking the add new user and it will redirect to add user interface. The admin can click the user and it will redirect to edit user interface.

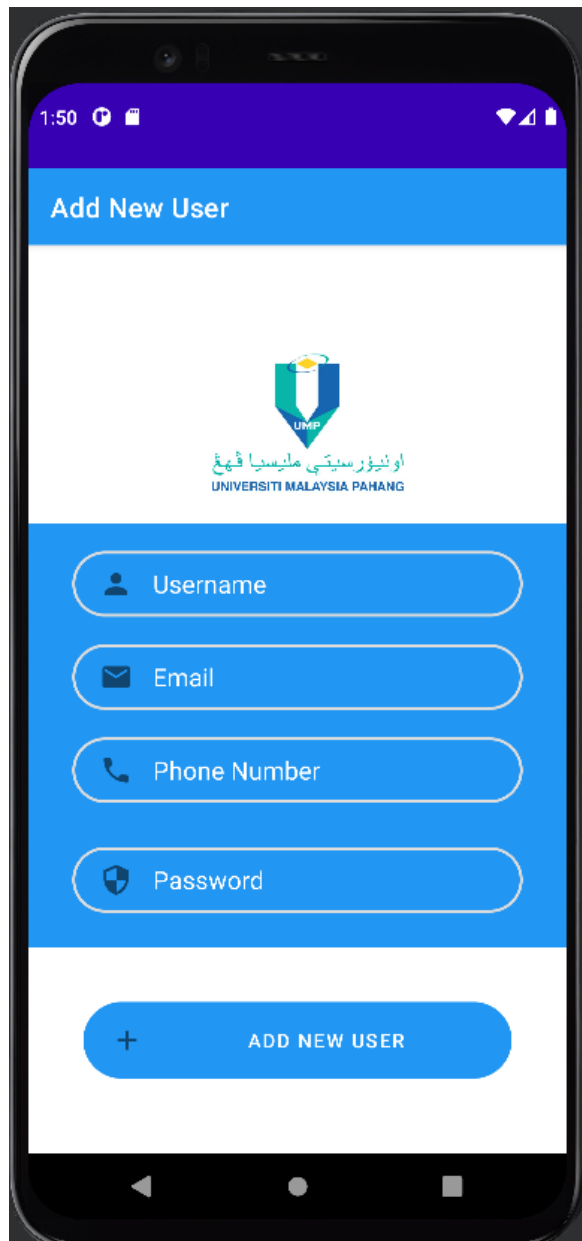


Figure 4.21 Add New User Interface

Figure 4.21 shows the add new user interface, the admin can add the new user by entering the username, email, phone number and password to create a new user. Admin will click the add new user button to create a new user and the system will verifying the data that entered by the admin.

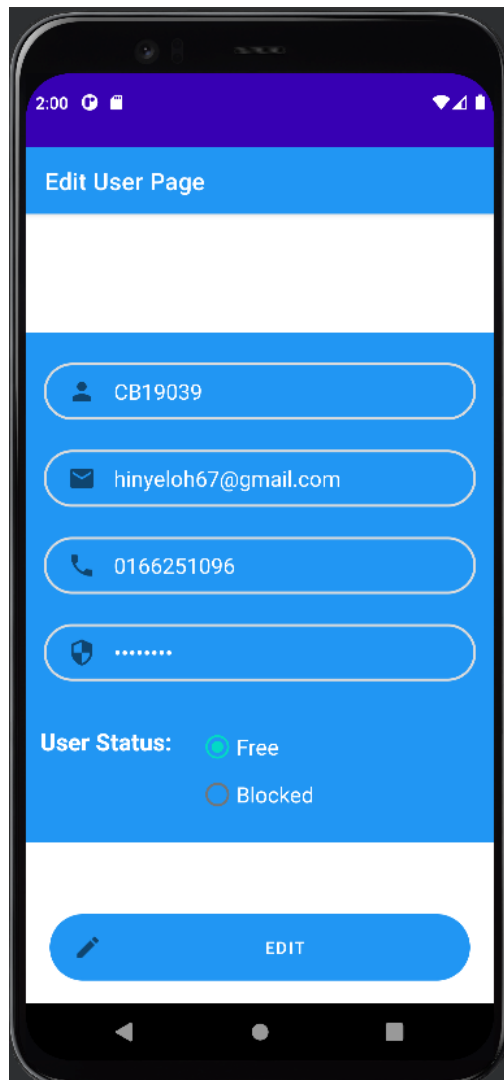


Figure 4.22 Edit User Page

Figure 4.22 shows the edit user page which allow admin to edit the user information which can block the user. The admin can click the edit button to update the new information of the user.

4.3.4 Implementation of Integrated Development Environment (IDE)

This section discusses about the implementation done for developing the system. The IDE being used for developing this system is Android Studio where it will implement for mobile application. Figure below will show three recycle view adapter which is using

for the watchlist, item and user display, calling function, SMS function and email function.

```
package com.example.fyp;

import ...

public class ItemAdapter extends RecyclerView.Adapter<ItemAdapter.MyViewHolder>{

    Context context;
    ArrayList<ItemHelperClass> list;
    private OnItemClickListener mOnItemClickListener;

    public ItemAdapter(Context context, ArrayList<ItemHelperClass> list, OnItemClickListener onItemClickListener) {
        this.context = context;
        this.list = list;
        this.mOnItemClickListener = onItemClickListener;
    }

    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(context).inflate(R.layout.item_row,parent, attachToRoot: false);

        return new MyViewHolder(v, mOnItemClickListener);
    }

    @Override
    public void onBindViewHolder(@NonNull ItemAdapter.MyViewHolder holder, int position) {
        ItemHelperClass itemHelperClass = list.get(position);
        holder.title.setText(itemHelperClass.getItem_title());
        holder.desc.setText(itemHelperClass.getItem_description());
        holder.price.setText(itemHelperClass.getItem_price());
```

```
        holder.price.setText(itemHelperClass.getItem_price());
        String img = itemHelperClass.getStorageReference();

        String cur = img;
        String[] sep = cur.split(regex: "ItemImage/");
        StorageReference sRef = FirebaseStorage.getInstance().getReference( location: "ItemImage").child(sep[1]);
        sRef.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
            @Override
            public void onSuccess(Uri uri) {
                Picasso.get().load(uri).resize( targetWidth: 125, targetHeight: 125).centerCrop().into(holder.img);
            }
        });
    }

    @Override
    public int getItemCount() { return list.size(); }

    public static class MyViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener{...}

    public interface OnItemClickListener{
        void onItemClick(int position);
    }
}
```

Figure 4.23 Item Adapter

Figure 4.23 shows the item adapter of the project. The item adapter will handle for the recycle view display of all the item that had been upload to the database. In this adapter, it can specific the click on the item and return the value of the item that the user clicked. The adapter will get the data from the database and set the data into the recycle view to be display in the interface.

```

package com.example.fyp;

import ...

public class WatchListAdapter extends RecyclerView.Adapter<WatchListAdapter.MyViewHolder> {

    Context context;
    ArrayList<WatchListClass> list;
    private OnItemClickListener mOnItemClickListener;

    public WatchListAdapter(Context context, ArrayList<WatchListClass> list, OnItemClickListener OnItemClickListener) {
        this.context = context;
        this.list = list;

        this.mOnItemClickListener = OnItemClickListener;
    }

    @NonNull
    @Override
    public WatchListAdapter.MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(context).inflate(R.layout.wishList_row,parent, attachToRoot: false);

        return new WatchListAdapter.MyViewHolder(v, mOnItemClickListener);
    }

    @Override
    public void onBindViewHolder(@NonNull WatchListAdapter.MyViewHolder holder, int position) {
        WatchListClass watchListClass = list.get(position);
        holder.btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

```

```

            DatabaseReference reference = FirebaseDatabase.getInstance().getReference( path: "WatchList");
            reference.child(watchListClass.getWatchList_ID()).removeValue();
            Toast.makeText(context, text:"Item are Removed from Your Watch List", Toast.LENGTH_LONG).show();
            Intent intent = new Intent(context, WatchListPage.class);
            context.startActivity(intent);
            notifyDataSetChanged();
        }
    });
    String item_ID = watchListClass.getItem_ID();
    DatabaseReference databaseReference = FirebaseDatabase.getInstance().getReference( path: "Item");
    databaseReference.child(item_ID).addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            ItemHelperClass itemHelperClass = snapshot.getValue(ItemHelperClass.class);
            String status = itemHelperClass.getItem_status();
            if(status.equals("Available")){
                holder.title.setText(itemHelperClass.getItem_title());
                holder.desc.setText(itemHelperClass.getItem_description());
                holder.price.setText(itemHelperClass.getItem_price());

                String img = itemHelperClass.getStorageReference();

                String cur = img;
                String[] sep = cur.split( regex: "ItemImage/");
                StorageReference sRef = FirebaseStorage.getInstance().getReference( location: "ItemImage").child(sep[1]);
                sRef.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                    @Override
                    public void onSuccess(Uri uri) {

```

```

        public void onSuccess(Uri uri) {
            Picasso.get().load(uri).resize(targetWidth: 125, targetHeight: 125).centerCrop().into(holder.img);
        }
    });
}

@Override
public void onCancelled(@NonNull DatabaseError error) {
}
});
}

@Override
public int getItemCount() { return list.size(); }
}

public static class MyViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener{
    TextView title, desc, price;
    ImageView btn;
    ImageView img;
    WatchListAdapter.OnItemsListener onItemsListener;
    public MyViewHolder(@NonNull View itemView, WatchListAdapter.OnItemsListener onItemsListener) {
        super(itemView);
        title = itemView.findViewById(R.id.item_title);
        desc = itemView.findViewById(R.id.item_description);
        price = itemView.findViewById(R.id.item_price);
        img = itemView.findViewById(R.id.item_image);
        btn = itemView.findViewById(R.id.remove_favour);
        this.onItemsListener = onItemsListener;

        itemView.setOnClickListener(this);
    }
}

```

```

        itemView.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) { onItemsListener.onItemsClick(getAdapterPosition()); }
}

public interface OnItemsListener{
    void onItemsClick(int position);
}
}

```

Figure 4.24 Watchlist Adapter

Figure 4.24 shows the Watchlist Adapter which allow user to will their own watchlist and display the specific information from the database to the recycle view. Besides, the adapter will handle the delete watchlist button which allow the user to remove the item from their watchlist. The OnItemListener function will get the correct data from the user click and redirect it to the Item Detail Page which allow user know more information of the item.

```

package com.example.fyp;

import ...

public class UserAdapter extends RecyclerView.Adapter<UserAdapter.MyViewHolder> {

    Context context;
    ArrayList<ReadWriteUserDetail> list;
    private OnUserListener mOnUserListener;

    public UserAdapter(Context context, ArrayList<ReadWriteUserDetail> list, OnUserListener onUserListener) {
        this.context = context;
        this.list = list;
        this.mOnUserListener = onUserListener;
    }

    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(context).inflate(R.layout.user_list,parent, attachToRoot false);
        return new MyViewHolder(v, mOnUserListener);
    }

    @Override
    public void onBindViewHolder(@NonNull UserAdapter.MyViewHolder holder, int position) {
        ReadWriteUserDetail readWriteUserDetail = list.get(position);
        holder.name.setText(readWriteUserDetail.getUsername());
        holder.email.setText(readWriteUserDetail.getUseremail());
        holder.phone.setText(readWriteUserDetail.getUserphone());
    }
}

```

```

@Override
public int getItemCount() { return list.size(); }

public static class MyViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener{
    TextView name, email, phone;
    OnUserListener onUserListener;
    public MyViewHolder(@NonNull View itemView, OnUserListener onUserListener) {
        super(itemView);
        name = itemView.findViewById(R.id.user_name);
        email = itemView.findViewById(R.id.user_email);
        phone = itemView.findViewById(R.id.user_phone);
        this.onUserListener = onUserListener;
        itemView.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) { onUserListener.onUserClick(getAdapterPosition()); }
}

public interface OnUserListener{
    void onUserClick(int position);
}
}

```

Figure 4.25 User Adapter

Figure 4.25 shows the User Adapter which will handle the data that getting from database and display the user information to the recycle view the user will be shown in the and able been clicked to redirect to the edit user page for the admin to edit the user

information and the admin are able to block the user if the admin receive some report from the user.

```
call = findViewById(R.id.call_btn);
call.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        reference = FirebaseDatabase.getInstance().getReference("Item");
        reference.child(item_id).addListenerForSingleValueEvent(new ValueEventListener() {
            @RequiresApi(api = Build.VERSION_CODES.N)
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                ItemHelperClass itemHelperClass = snapshot.getValue(ItemHelperClass.class);
                if(itemHelperClass != null){
                    String userID = itemHelperClass.getUser_ID();

                    if (ContextCompat.checkSelfPermission(context, Manifest.permission.CALL_PHONE) == PackageManager.PERMISSION_GRANTED){
                        userReference = FirebaseDatabase.getInstance().getReference("Registered User");
                        userReference.child(userID).addListenerForSingleValueEvent(new ValueEventListener() {
                            @Override
                            public void onDataChange(@NonNull DataSnapshot snapshot) {
                                ReadWriteUserDetail readWriteUserDetail = snapshot.getValue(ReadWriteUserDetail.class);
                                if (readWriteUserDetail != null){
                                    String phone_num = readWriteUserDetail.getUserphone();
                                    Intent i = new Intent(Intent.ACTION_CALL);
                                    i.setData(Uri.parse("tel:" + phone_num));
                                    startActivity(i);
                                }
                            }
                        });
                    }
                }
            }
        });
    }
});
```

```
    @Override
    public void onCancelled(@NonNull DatabaseError error) {
    }
});
} else {
    ActivityCompat.requestPermissions(activity, new String[] {Manifest.permission.CALL_PHONE}, requestCode);
}
}
}
@Override
public void onCancelled(@NonNull DatabaseError error) {
}
});
});
```

Figure 4.26 Calling function

Figure 4.26 shows the calling function which can get the item's owner contact number and call the item own for more item information.

```
    sms = findViewById(R.id.sms_btn);
    sms.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            reference = FirebaseDatabase.getInstance().getReference("Item");
            reference.child(item_id).addListenerForSingleValueEvent(new ValueEventListener() {
                @RequiresApi(api = Build.VERSION_CODES.N)
                @Override
                public void onDataChange(@NonNull DataSnapshot snapshot) {
                    ItemHelperClass itemHelperClass = snapshot.getValue(ItemHelperClass.class);
                    if(itemHelperClass != null){
                        String userID = itemHelperClass.getUser_ID();
                        String title = itemHelperClass.getItem_title();

                        if (ContextCompat.checkSelfPermission(context, ItemDetailPage.this, Manifest.permission.SEND_SMS) == PackageManager.PERMISSION_GRANTED){
                            userReference = FirebaseDatabase.getInstance().getReference("Registered User");
                            userReference.child(userID).addListenerForSingleValueEvent(new ValueEventListener() {
                                @Override
                                public void onDataChange(@NonNull DataSnapshot snapshot) {
                                    ReadWriteUserDetail readWriteUserDetail = snapshot.getValue(ReadWriteUserDetail.class);
                                    if (readWriteUserDetail != null){
                                        String phone_num = readWriteUserDetail.getUserphone();
                                        String message = "Is the "+title+" still available?";

                                        SmsManager smsManager = SmsManager.getDefault();
```

```
                                        smsManager.sendTextMessage(phone_num, scAddress: null, message, sendIntent: null, deliveryIntent: null);
                                        Toast.makeText(context, ItemDetailPage.this, text: "SMS Send Successfully", Toast.LENGTH_SHORT);
                                    }else{
                                        Toast.makeText(context, ItemDetailPage.this, text: "SMS Send Failure", Toast.LENGTH_SHORT);
                                    }
                                }
                            });
                        }
                    }
                }
            });
        }else{
            ActivityCompat.requestPermissions(activity, ItemDetailPage.this, new String[] {Manifest.permission.SEND_SMS}, requestCode: 181);
        }
    }
}
@Override
public void onCancelled(@NonNull DatabaseError error) {
}
});
});
```

Figure 4.27 SMS function

Figure 4.27 shows the SMS function which can get the item's owner contact number and sending a SMS to ask for the item availability.

```

email = findViewById(R.id.email_btn);
email.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        reference = FirebaseDatabase.getInstance().getReference("Item");
        reference.child(item_id).addListenerForSingleValueEvent(new ValueEventListener() {
            @RequiresApi(api = Build.VERSION_CODES.N)
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                ItemHelperClass itemHelperClass = snapshot.getValue(ItemHelperClass.class);
                if(itemHelperClass != null){
                    String userID = itemHelperClass.getUser_ID();
                    String title = itemHelperClass.getItem_title();

                    userReference = FirebaseDatabase.getInstance().getReference("Registered User");
                    userReference.child(userID).addListenerForSingleValueEvent(new ValueEventListener() {
                        @Override
                        public void onDataChange(@NonNull DataSnapshot snapshot) {
                            ReadWriteUserDetail readWriteUserDetail = snapshot.getValue(ReadWriteUserDetail.class);
                            if (readWriteUserDetail != null){

                                String userEmail = readWriteUserDetail.getUseremail();
                                String subject = "Check Item Available Status";
                                String body = "Is the "+title+" still available?";

                                Intent i = new Intent(Intent.ACTION_SEND);
                                i.setType("plain/text");
                                i.putExtra(Intent.EXTRA_EMAIL, new String[]{UserEmail});
                                i.putExtra(Intent.EXTRA_SUBJECT, subject);
                                i.putExtra(Intent.EXTRA_TEXT, body);

                                i.setType("message/rfc822");

                                if (i.resolveActivity(getPackageManager()) != null){
                                    startActivity(Intent.createChooser(i, "Choose an email:"));
                                }else {
                                    Toast.makeText(context, "Not Support", Toast.LENGTH_LONG).show();
                                }
                            }
                        }
                    });
                }
            }
        });
    }
});

@Override
public void onCancelled(@NonNull DatabaseError error) {
}

@Override
public void onCancelled(@NonNull DatabaseError error) {
}
});
});

```

Figure 4.28 email function

Figure 4.28 shows the email function which can get the item's owner email and sending an email to ask for the item availability.

4.4 Discussion

This part discusses about the outcome produced throughout the development process of the project. Testing and inspection process were carried out along the process to ensure the outcome is not against expectation. A user acceptance test (UAT) report is done to test the functionality of the developed application. The functionality test is done to ensure that the system meets the user requirement.

Each test case is designed and implemented based on the interfaces in the project and the functions, text fields, button, data displayed and so on had been tested as much as possible for each interface. The test cases are shown in the table below.

Table 4.1 Test Case Login Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
L001	Check empty input for email	Click <<LOGIN>> button without email input	Display focus error on the email field and display message "Email is required"	Display focus error on the email field and display message "Email is required"	Pass	-
L002	Check empty input for password	Click <<LOGIN>> button without password input	Display focus error on the password field and display message "Password is required"	Display focus error on the password field and display message "Password is required"	Pass	-
L003	Check for not matched email and password as the registered account	Email and password input incorrectly	Display focus error on the email field and display message "Invalid credentials."	Display focus error on the email field and display message "Invalid credentials."	Pass	-

			Please re-enter”	Please re-enter”		
L004	Check the validation of the email	Input email address without “@” and “.”	Display focus error on the email field and display message “Valid email is required”	Display focus error on the email field and display message “Valid email is required”	Pass	-
L005	Forget Password	Click the <<Forget Password>> text	Redirect to reset password interface	User redirect to reset password interface and the interface display correctly	Pass	-
L006	Sign up an account	Click the <<Sign up an account>> text	Redirect to the register interface	User redirect to register interface and the interface display correctly	Pass	-
L007	Login as user	Input the valid and registered email and password and click the <<LOGIN>> button	Login successfully , display message “User is logged in now” and	Login successfully, display message “User is logged in now” and	Pass	-

			redirect to the main page.	redirect to the main page.		
L008	Login as admin	Input the admin email and admin password	Login successfully, display message “Admin is logged in now” and redirect to the admin page.	Login successfully, display message “Admin is logged in now” and redirect to the admin page.	Pass	-

Table 4.2 Test Case Register Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
R001	Check empty input for username	Click <<REGISTER>> button without username input	Display focus error on the username field and display message “Username is required”	Display focus error on the username field and display message “Username is required”	Pass	-
R002	Check empty input for email	Click <<REGISTER>> button	Display focus error on the email field and	Display focus error on the email field and display	Pass	-

		without email input	display message “Email is required”	message “Email is required”		
R003	Check empty input for phone number	Click <<REGISTER>> button without phone number input	Display focus error on the phone number field and display message “Phone number is required”	Display focus error on the phone number field and display message “Phone number is required”	Pass	-
R004	Check empty input for password	Click <<REGISTER>> button without password input	Display focus error on the password field and display message “Password is required”	Display focus error on the password field and display message “Password is required”	Pass	-
R005	Check validation of username	Enter username more than 20 characters	Display focus error on the username field and display message “Username is too long”	Display focus error on the username field and display message “Username is too long”	Pass	-

R006	Check validation of phone number	Enter less than 10 digits of phone number	Display focus error on the phone number field and display message “Phone Number should be more or equal to 10 no.”	Display focus error on the phone number field and display message “Phone Number should be more or equal to 10 no.”	Pass	-
R007	Check the validation of phone number	Input phone number not started from “01”	Display focus error on the phone number field and display message “Phone number is not valid”	Display focus error on the phone number field and display message “Phone number is not valid”	Pass	-
R008	Check validation of phone number	Enter more than 12 digits of phone number	Display focus error on the phone number field and display message	Display focus error on the phone number field and display message “Phone Number	Pass	-

			“Phone Number should be less or equal to 11 no.”	should be less or equal to 11 no.”		
R009	Check the validation of the email	Input email address without “@” and “.”	Display focus error on the email field and display message “Valid email is required”	Display focus error on the email field and display message “Valid email is required”	Pass	-
R010	Check validation of password	Input password with less than 5 characters	Display focus error on the password field and display message “Password should more than 5 characters”	Display focus error on the password field and display message “Password should more than 5 characters”	Pass	-
R011	Already Have an Account	Click the << Already Have an Account >> text	Redirect to login interface	User redirect to login interface and the interface display correctly	Pass	-

R012	Register an account	Input email, username, phone number and password with the correct format and the email has not been used before	Display message “User Register Successfull y” and redirect to the login page	Display message “User Register Successfully” and redirect to the login page	Pass	-
R013	Check the email already existed	Input the email that had registered before	Display focus error on the email field and display message “User is already registered with this email”	Display focus error on the email field and display message “User is already registered with this email”	Pass	-

Table 4.3 Test Case Forget Password Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
F001	Check empty input for email	Click <<RESET PASSWORD >> button without email input	Display focus error on the email field and display message	Display focus error on the email field and display message “Email is required”	Pass	-

			“Email is required”			
F002	Check the validation of the email	Input email address without “@” and “.”	Display focus error on the email field and display message “Valid email is required”	Display focus error on the email field and display message “Valid email is required”	Pass	-
F003	Reset Password	Input valid and registered email	Display message “Please check your email for password reset link” and redirect to the login page	Display message “Please check your email for password reset link” and redirect to the login page	Pass	-
F004	Check for not registered email	Input valid email but not registered email	Display message “Something Wrong”	Display message “Something Wrong”	Pass	-

Table 4.4 Test Case Main Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
---------	-------	-----------	-----------------	---------------	------------	---------

M001	View Watchlist	Click the favour icon button on the action bar	Redirect to the watch list page	Redirect to the watch list page	Pass	-
M002	View Profile	Click the profile icon button on the action bar	Redirect to the profile page	Redirect to the profile page	Pass	-
M003	View Sell Option page	Click the <<SELL>> button	Redirect to the Sell Option page	Redirect to the Sell Option page	Pass	-
M004	Check empty of search box	Click the search icon button without search box input	Redirect to the search page and show all item	Redirect to the search page and show all item	Pass	-
M005	Check search box with data	Input some character and click the search icon button.	Redirect to the search page and show the item title that contains the input of the search box	Redirect to the search page and show the item title that contains the input of the search box	Pass	-
M006	Check the Mobile Phone and Gadgets button	Click the image of the “Mobile Phone and Gadgets”	Redirect to the search page	Redirect to the search page	Pass	-

M007	Check the Health and Beauty button	Click the image of the “Health and Beauty”	Redirect to the search page	Redirect to the search page	Pass	-
M008	Check the Sports and Outdoor button	Click the image of the “Sports and Outdoor”	Redirect to the search page	Redirect to the search page	Pass	-
M009	Check the Donation button	Click the image of the “Donation”	Redirect to the search page	Redirect to the search page	Pass	-

Table 4.5 Test Case Sell Option Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
SO00 1	View Watchlist	Click the favour icon button on the action bar	Redirect to the watch list page	Redirect to the watch list page	Pass	-
SO00 2	View Profile	Click the profile icon button on the action bar	Redirect to the profile page	Redirect to the profile page	Pass	-
SO00 3	Upload new Item	Click the <<UPLOAD NEW ITEM>> button	Redirect to Sell page	Redirect to Sell page	Pass	-
SO00 4	Check Item that had	Click the <<Item	Redirect to Item page	Redirect to Item page	Pass	-

	uploaded by user	Page>> button				
SO005	Donate Item	Click the <<DONATE ITEM>> button	Redirect to donate item page	Redirect to donate item page	Pass	-
SO006	Back to main page	Click the <<BACK TO MAIN PAGE>> button	Redirect to main page	Redirect to main page	Pass	-

Table 4.6 Test Case Upload Item Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
UI001	View Watchlist	Click the favour icon button on the action bar	Redirect to the watch list page	Redirect to the watch list page	Pass	-
UI002	View Profile	Click the profile icon button on the action bar	Redirect to the profile page	Redirect to the profile page	Pass	-
UI003	Check empty input for title	Click <<UPLOAD>> button without title input	Display focus error on the title field and display message “Title is required”	Display focus error on the email field and display message “Title is required”	Pass	-

UI004	Check empty input for description	Click <<UPLOAD>> button without description input	Display focus error on the description field and display message “Item description is required”	Display focus error on the email field and display message “Item description is required”	Pass	-
UI005	Check empty input for price	Click <<UPLOAD>> button without price input	Display focus error on the price field and display message “Item price is required”	Display focus error on the email field and display message “Item price is required”	Pass	-
UI006	Upload Item	Enter all the valid input and click the <<UPLOAD>> button	Display message “Item Added Successfully” and redirect to sell option page	Display message “Item Added Successfully” and redirect to sell option page	Pass	-
UI007	Check empty input for category	Click <<UPLOAD>> button without category input	Display focus error on the category field and display	Display focus error on the category field and display message	Pass	-

			message “category is required”	“category is required”		
UI008	Check empty input for type	Click <<UPLOAD>> button without type input	Display focus error on the type field and display message “type is required”	Display focus error on the type field and display message “type is required”	Pass	-
UI009	Check empty input for condition	Click <<UPLOAD>> button without condition input	Display focus error on the condition field and display message “condition is required”	Display focus error on the condition field and display message “condition is required”	Pass	-

Table 4.7 Test Case Item Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
I001	View Watchlist	Click the favour icon button on the action bar	Redirect to the watch list page	Redirect to the watch list page	Pass	-
I002	View Profile	Click the profile icon	Redirect to the profile page	Redirect to the profile page	Pass	-

		button on the action bar				
I003	Check item	Click the selected item	Redirect to the edit item page with correct item information	Redirect to the edit item page with correct item information	Pass	-

Table 4.8 Test Case Edit Item Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
EI001	View Watchlist	Click the favour icon button on the action bar	Redirect to the watch list page	Redirect to the watch list page	Pass	-
EI002	View Profile	Click the profile icon button on the action bar	Redirect to the profile page	Redirect to the profile page	Pass	-
EI003	Check empty input for title	Click <<UPDATE>> button without title input	Display focus error on the title field and display message "Title is required"	Display focus error on the email field and display message "Title is required"	Pass	-
EI004	Check empty input for description	Click <<UPDATE>> button without	Display focus error on the description	Display focus error on the email field and display	Pass	-

		description input	field and display message “Item description is required”	message “Item description is required”		
EI005	Check empty input for price	Click << UPDATE>> button without price input	Display focus error on the price field and display message “Item price is required”	Display focus error on the email field and display message “Item price is required”	Pass	-
EI006	Update item	Input valid data and click the <<UPDATE> > button	Display message “Edit Successfull y” and redirect to Sell option page	Display message “Edit Successfully” and redirect to Sell option page	Pass	-
EI007	Delete item	Click the <<DELETE> > button	Display message “Item Deleted Successfull y” and redirect to sell option page	Display message “Item Deleted Successfully” and redirect to sell option page	Pass	-

Table 4.9 Test Case Donate Item Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
DI001	View Watchlist	Click the favour icon button on the action bar	Redirect to the watch list page	Redirect to the watch list page	Pass	-
DI002	View Profile	Click the profile icon button on the action bar	Redirect to the profile page	Redirect to the profile page	Pass	-
DI003	Check empty input for title	Click <<DONATE >> button without title input	Display focus error on the title field and display message "Title is required"	Display focus error on the email field and display message "Title is required"		
DI004	Check empty input for description	Click << DONATE>> button without description input	Display focus error on the description field and display message "Item description is required"	Display focus error on the email field and display message "Item description is required"		
DI005	Check empty input	Click <<DONATE >> button	Display focus error on the	Display focus error on the condition	Pass	-

	for condition	without condition input	condition field and display message “condition is required”	field and display message “condition is required”		
DI006	Check empty input for category	Click <<DONATE >> button without category input	Display focus error on the category field and display message “category is required”	Display focus error on the category field and display message “category is required”	Pass	-
DI007	Donate Item	Enter all the valid information and click the <<DONATE >> button	Display message “Item Added Successfully” and redirect to sell option page	Display message “Item Added Successfully” and redirect to sell option page	Pass	-

Table 4.10 Test Case Watchlist Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
W001	View Watchlist	Click the favour icon	Redirect to the watch list page	Redirect to the watch list page	Pass	-

		button on the action bar				
W002	View Profile	Click the profile icon button on the action bar	Redirect to the profile page	Redirect to the profile page	Pass	-
W003	Check watch list item	Click on the selected watch list item	Redirect to the item detail page with the correct item information	Redirect to the item detail page with the correct item information	Pass	-
W004	Delete Watch List	Click the trash bin icon	Display message “Item are Removed from your watch list” and refresh the page	Display message “Item are Removed from your watch list” and refresh the page	Pass	-

Table 4.11 Test Case Profile Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
P001	View Watchlist	Click the favour icon button on the action bar	Redirect to the watch list page	Redirect to the watch list page	Pass	-
P002	View Profile	Click the profile icon	Redirect to the profile page	Redirect to the profile page	Pass	-

		button on the action bar				
P003	Edit Profile	Click the <<EDIT PROFILE>> button	Redirect to edit profile page	Redirect to edit profile page	Pass	-

Table 4.12 Test Case Edit Profile Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
EP00 1	View Watchlist	Click the favour icon button on the action bar	Redirect to the watch list page	Redirect to the watch list page	Pass	-
EP00 2	View Profile	Click the profile icon button on the action bar	Redirect to the profile page	Redirect to the profile page	Pass	-
EP00 3	Check empty input for username	Click <<UPDATE> > button without username input	Display focus error on the username field and display message "Username is required"	Display focus error on the username field and display message "Username is required"	Pass	-
EP00 4	Check character of username	Click <<UPDATE> > button with more than 20	Display focus error on the username	Display focus error on the username field and	Pass	-

	more than 20	characters input	field and display message “Username is too long”	display message “Username is too long”		
EP005	Check validation of phone number	Enter less than 10 digits of phone number	Display focus error on the phone number field and display message “Phone Number should be more or equal to 10 no.”	Display focus error on the phone number field and display message “Phone Number should be more or equal to 10 no.”	Pass	-
EP006	Check validation of phone number	Enter more than 12 digits of phone number	Display focus error on the phone number field and display message “Phone Number should be less or equal to 11 no.”	Display focus error on the phone number field and display message “Phone Number should be less or equal to 11 no.”	Pass	-

EP00 7	Check the validation of phone number	Input phone number not started from "01"	Display focus error on the phone number field and display message "Phone number is not valid"	Display focus error on the phone number field and display message "Phone number is not valid"	Pass	-
EP00 8	Check empty input for phone number	Click <<UPDATE>> button without phone number input	Display focus error on the phone number field and display message "Phone number is required"	Display focus error on the phone number field and display message "Phone number is required"	Pass	-
EP00 9	Update Profile	Input the valid information and click the <<UPDATE>> button	Display message "Profile Update Successful" and redirect to profile page	Display message "Profile Update Successfully" and redirect to profile page	Pass	-

Table 4.13 Test Case Search Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
S001	View Watchlist	Click the favour icon button on the action bar	Redirect to the watch list page	Redirect to the watch list page	Pass	-
S002	View Profile	Click the profile icon button on the action bar	Redirect to the profile page	Redirect to the profile page	Pass	-
S003	Filter by Category	Click and select the drop-down list from the select categories drop-down list.	The item will be filter as the selected categories	The item will be filter as the selected categories	Pass	-
S004	Filter by Condition	Click and select the drop-down list from the Condition and Type drop-down list.	The item will be filter as the selected Condition	The item will be filter as the selected Condition	Pass	-
S005	Filter by Type	Click and select the drop-down list from the Condition and Type drop-down list.	The item will be filter as the selected type	The item will be filter as the selected type	Pass	-

S006	Sort by price	Click and select the drop-down list from the Sort By drop-down list.	The item will be sort as the selected type of sort in ascending or descending	The item will be sort as the selected type of sort in ascending or descending	Pass	-
S007	Select Item	Click the select item	Redirect to the item detail page with the correct data of the item	Redirect to the item detail page with the correct data of the item	Pass	-

Table 4.14 Test Case Item Detail Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
ID001	View Watchlist	Click the favour icon button on the action bar	Redirect to the watch list page	Redirect to the watch list page	Pass	-
ID002	View Profile	Click the profile icon button on the action bar	Redirect to the profile page	Redirect to the profile page	Pass	-
ID003	Add to Watch list	Click the favor icon button	Display message "Item Successfull	Display message "Item Successfully	Pass	-

			y add to watch list”	add to watch list”		
ID004	Call seller	Click the <<CALL>> button	Redirect to the call method of device	Redirect to the call method of device	Pass	-
ID005	Message seller using SMS	Click the <<SMS>> button	SMS send successfully	SMS send successfully	Pass	-
ID006	Email seller	Click the <<email>> button	Redirect to the other application that able to send email	Redirect to the other application that able to send email	Pass	-

Table 4.15 Test Case Admin Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
A001	Add new User	Click the <<Add New User>>	Redirect to add new user page	Redirect to add new user page	Pass	-
A002	Select user	Select the specific user	Redirect to edit user page with the correct data	Redirect to edit user page with the correct data	Pass	-

Table 4.16 Test Case Edit User Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
EU001	Block or unblock user	Select the radio button of the status	Admin can select the radio button	Admin can select the radio button	Pass	-
EU002	Update user	Click the <<EDIT>> button	Display message “Edit Successfull y” and redirect to admin page	Display message “Edit Successfully” and redirect to admin page	Pass	-

Table 4.17 Test Case Add New User Page

Test ID	Event	Test Data	Expected Result	Actual Result	Pass /Fail	Comment
ANU001	Check empty input for username	Click <<ADD NEW USER>> button without username input	Display focus error on the username field and display message “Username is required”	Display focus error on the username field and display message “Username is required”	Pass	-
ANU002	Check empty input for email	Click <<ADD NEW USER >> button without email input	Display focus error on the email field and display message	Display focus error on the email field and display message	Pass	-

			“Email is required”	“Email is required”		
ANU 003	Check empty input for phone number	Click << ADD NEW USER >> button without phone number input	Display focus error on the phone number field and display message “Phone number is required”	Display focus error on the phone number field and display message “Phone number is required”	Pass	-
ANU 004	Check empty input for password	Click << ADD NEW USER >> button without password input	Display focus error on the password field and display message “Password is required”	Display focus error on the password field and display message “Password is required”	Pass	-
ANU 005	Check validation of phone number	Enter less than 10 digits of phone number	Display focus error on the phone number field and display message “Phone Number should be	Display focus error on the phone number field and display message “Phone Number should be more or equal to 10 no.”	Pass	-

			more or equal to 10 no.”			
ANU 006	Check the validation of phone number	Input phone number not started from “01”	Display focus error on the phone number field and display message “Phone number is not valid”	Display focus error on the phone number field and display message “Phone number is not valid”	Pass	-
ANU 007	Check validation of phone number	Enter more than 12 digits number of phone number	Display focus error on the phone number field and display message “Phone Number should be less or equal to 11 no.”	Display focus error on the phone number field and display message “Phone Number should be less or equal to 11 no.”	Pass	-
ANU 008	Check the validation of the email	Input email address without “@” and “.”	Display focus error on the email field and display message	Display focus error on the email field and display message	Pass	-

			“Valid email is required”	“Valid email is required”		
ANU 009	Check validation of password	Input password with less than 5 characters	Display focus error on the password field and display message “Password should more than 5 characters”	Display focus error on the password field and display message “Password should more than 5 characters”	Pass	-
ANU 010	Check validation of username	Enter username more than 20 characters	Display focus error on the username field and display message “Username is too long”	Display focus error on the username field and display message “Username is too long”	Pass	-
ANU 011	Check the email already existed	Input the email that had registered before	Display focus error on the email field and display message “User is already registered	Display focus error on the email field and display message “User is already registered with this email”	Pass	-

			with this email”			
ANU 012	Add new user	Input all valid information and click <<ADD NEW USER>> button	Display message “User Added Successfully” and redirect to admin page	Display message “User Added Successfully” and redirect to admin page	Pass	-

4.5 Chapter Summary

To conclude, the project has been implemented and tested with User Acceptance Test with several functionality of the application. As result to the test, the application can perform correctly with its proposed functionality.

CHAPTER 5

CONCLUSION

5.1 Objective Revisited

There are three objectives proposed for this project. Each of the objectives will be revisited in this section.

The first objective of this project is to study the existing mobile application related to mobile commerce. Project report and thesis shows that the objective has successfully done throughout the process of developing the application.

The second objective of this project is to design and develop an application as a platform for UMP students to purchase and sell items using Android Studio. This project has successfully enabled user to upload their item to the application and able be view by other user according to their specific needs.

The third objective of this project is to evaluate the functionality of the developed mobile application that are able students to trade items. This means to evaluate the functionality to upload new item, donate item, add item to watchlist and other functions. This project has achieved this objective since the application is consist of all the functions expected by the user.

5.2 Limitation

There are several limitations on the resulted product or application of this project. The limitations are listed as below.

1. The resulted application can only be installed and used in android devices since the project is implementing using Android Studio. The limitation of the Android Studio is using JAVA or Kotlin which is the official language of Android Studio. However, unlike the web application, the web application can run on any platform.

2. To use the application, the user must login with the registered account before using the function of the application. All the users are required to have an account before logged in to the application. However, there have some application that allow user to view some function of the application before logged in to the application.
3. The application does not provide any payment method for users. A lot of e-commerce application are allows to make payment in the application for making order but in this project is not allowed.

5.3 Future Work

Based on the listed limitations, several improvement and implementation can be made for future work. First, for the application that can only be installed and used in android devices and it can be improved by broaden the supporting operating system used by the application. This could be done by implementing an application that can support both Android and iOS operating system which is broadly used by the people worldwide. For both Android and iOS operating system, it can be solved by using multiple platform supportive framework like Flutter to implement the application. Second, for the application it is only allow the user login with the registered account before using the function of the application and it can be improved by change the initial setting of the application which the user can view the item before logged in but it will request user to login with the registered account when using some specific functions like upload item, view the watchlist and other functions. Third, for the application it do not provide any payment method for users and it can be improved by adding a mobile payment gateway into the application which allow the user to do the payments directly from the application. For the application, it can add the top payment gateways like PayPal, TnG and any bank transfer gateway to make the application complete.

REFERENCES

- i) Mroczkowska, A. (2022, January 14). What is a mobile app?: App Development Basics for Businesses. Droids On Roids, from <https://www.thedroidsonroids.com/blog/what-is-a-mobile-app-app-development-basics-for-businesses>
- ii) Contributor, T. T. (2019, April 3). What is M-Commerce (Mobile Commerce)? - definition from whatis.com. SearchMobileComputing, 2022, from <https://www.techtarget.com/searchmobilecomputing/definition/m-commerce>
- iii) Kumari, R. (2020, October 2). M-Commerce: Applications, Advantages and Disadvantages | Analytics Steps, from <https://www.analyticssteps.com/blogs/m-commerce-applications-advantages-and-disadvantages>
- iv) Ying, L. (2021, June 20). 10 MOBILE USAGE STATISTICS EVERY MARKETER SHOULD KNOW IN 2021, from <https://www.oberlo.com/blog/mobile-usage-statistics>
- v) Liem, Vo. (2018, Aug 28). The MVP architecture pattern in Flutter with simple demo, from <https://medium.com/codechai/the-mvp-architecture-pattern-in-flutter-with-simple-demo-65ab3282c54b>
- vi) Raluca, Budiu. (2015, April 19). Mobile User Experience: Limitations and Strengths, from <https://www.nngroup.com/articles/mobile-ux/>
- vii) Daleska Pedriques. (2022, April 06). What is a Context Diagram (and How Can You Create One)? From <https://venngage.com/blog/context-diagram/>

- viii) Toby Cox. (2022, January 14). What Is Rapid Application Development (RAD)? from <https://blog.capterra.com/what-is-rapid-application-development/>
- ix) Koelsch, G. (2016). Requirements writing for system engineering. Apress. from https://books.google.com.my/books?id=2OtNDQAAQBAJ&pg=PA84&lpg=PA84&dq=Business+Rules+Transaction+corrections,+adjustments+and+cancellations+Administrative+functions+Authentication+Authorization+levels+Audit+Tracking+External+Interfaces+Certification+Requirements+Reporting+Requirements+Historical+Data+Legal+or+Regulatory+Requirements&source=bl&ots=cgIbpvGznS&sig=ACfU3U0jJWExE4D5ND8VxrNSrCMbzfTVTw&hl=en&sa=X&redir_esc=y#v=onepage&q&f=false
- x) Thomas Hamilton. (2022, April 30). What is User Acceptance Testing (UAT)? with Examples, from <https://www.guru99.com/user-acceptance-testing.html>
- xi) Zanifina Svirca. (2020, July 18). Model View Presenter(MVP), from <https://medium.datadriveninvestor.com/model-view-presenter-mvp-5c3439227f83>

APPENDIX A

SRS

2022

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

[UMP ITEMS TRADING APPLICATION]



DOCUMENT APPROVAL

	Name	Date
Authenticated by: <hr/> Name		
Approved by:		

<hr/> Client		
-----------------	--	--

Software :

Archiving Place :

TABLE OF CONTENT

CONTENT	PAGE
DOCUMENT APPROVAL	II
TABLE OF CONTENT	IV
LIST OF FIGURES	VI
LIST OF TABLES	VII
LIST OF APPENDICES	VIII
1	CHAPTER 1 1
1.1 PROJECT DESCRIPTION	1
1.2 SYSTEM IDENTIFICATION	2
1.3 CONTEXT DIAGRAM	3
1.4 DATA FLOW DIAGRAM	5
2	CHAPTER 2 1
2.1 USE CASE DIAGRAM AND DESCRIPTION	1

2.1.1	Manage Registration	1
2.1.2	Manage Profile	5
2.1.3	Manage Goods	8
2.1.4	Manage Order	13
2.1.5	Manage User	17
2.2	SEQUENCE DIAGRAM	20
3	CHAPTER 3	28
3.1	INTERFACE DESIGN	28
3.2	HARDWARE AND SOFTWARE SPECIFICATION	45

LIST OF FIGURES

Figure 1.1 Context Diagram of ITA	3
Figure 1.2 Data Flow Diagram Level 1 Register, Login, and Edit Profile	5
Figure 1.3 Data Flow Diagram Level 1 Search items, Filter items, and Contact Seller	6
Figure 1.4 Data Flow Diagram Level 1 Upload Item, Edit Item and Donate Item by seller.....	7
Figure 2.1 Use Case Diagram of Registration	1
Figure 2.2 Use Case Diagram of Manage Profile	5
Figure 2.3 Use Case Diagram of Manage Trades	8
Figure 2.4 Use Case Diagram of Manage Order	13
Figure 2.5 Use Case Diagram of Manage User	17
Figure 2.6 Sequence Diagram of Manage Registration	20
Figure 2.7 Sequence Diagram of Manage Profile	21
Figure 2.8 Sequence Diagram of Manage Goods with Alternative 1	22
Figure 2.9 Sequence Diagram of Manage Goods with Alternatives 2, 3 & 4	23
Figure 2.10 Sequence Diagram of Manage Goods with Alternatives 5	24
Figure 2.11 Sequence Diagram of Manage Order	25
Figure 2.12 Sequence Diagram of Manage User	26

LIST OF TABLES

Table 2.1 Manage Registration	2
Table 2.2 Manage Profile	5
Table 2.3 Manage Goods	9
Table 2.4 Manage Order	14
Table 2.5 Manage User	17
Table 3.1 Software and Hardware Specification	45

LIST OF APPENDICES

Appendix - A 1: Login Page	28
Appendix - A 2: Forget Password Page	29
Appendix - A 3: Register Page	30
Appendix - A 4: Main Page	31
Appendix - A 5: Sell Option Page	32
Appendix - A 6: Upload New Item Page	33
Appendix - A 7: Item Page	34
Appendix - A 8: Edit Item Page	35
Appendix - A 9: Search Page	36
Appendix - A 10: Item Detail Page	37
Appendix - A 11: Watch List Page	38
Appendix - A 12: Profile Page	39
Appendix - A 13: Edit Profile Page	40
Appendix - A 14: Admin Page	41
Appendix - A 15: Add New User	42
Appendix - A 16: Edit User	43
Appendix - A 17: Donate Page	44

1 CHAPTER 1

1.1 PROJECT DESCRIPTION

UMP Items Trading Application is an online mobile application that will be used by students of University Malaysia Pahang where it is specializing in purchases and selling items among UMP students. This application will be used by the sellers to sell their items on the platform and buyers able to buy the items uploaded by the sellers. This application can keep huge data organized and secured.

This application will be an online commerce system. The application will be designed to allow all the UMP students to use the system to manage and review the relevant information. The functionality requirement of the ITA is to manage registration, manage profiles, manage trades, and manage orders.

The first module is to manage registration. This module will be used by users to register an account for the application. After the users get the account from the application, users can log in to the system and get the application's main function like searching for items and selling items in the application.

The second module is to manage profiles. This module will be used by users to edit their profiles and update their information in the application database. Users can change their name, and phone number in their profile.

The third module is to manage goods. This module will be used by sellers to manage the items that want to sell. Sellers can upload items, edit items and delete items of their items. Sellers uploading items are required to fulfill all the required information for the items to sell.

The fourth module is to manage orders. This module will be used by buyers to manage the item that they want to buy. Buyers can search for items and add the items to the watchlist. Buyers can make orders with the seller by communicating with the seller by email, SMS, call, or chat boxes to make orders directly with the seller.

The last module is manage users. This module will be used by admin to manage the user which include add and block the user in the system. Admin can add the user by enter the username, email, phone number and password to create a new user to the system. Admin also have permission to block the user which the user that had been blocked will not allow to log in to the application.

1.2 SYSTEM IDENTIFICATION

System Title: UMP Items Trading Application

System Abbreviation: ITA

System Identification Number: ITA v1 2022

1.3 CONTEXT DIAGRAM

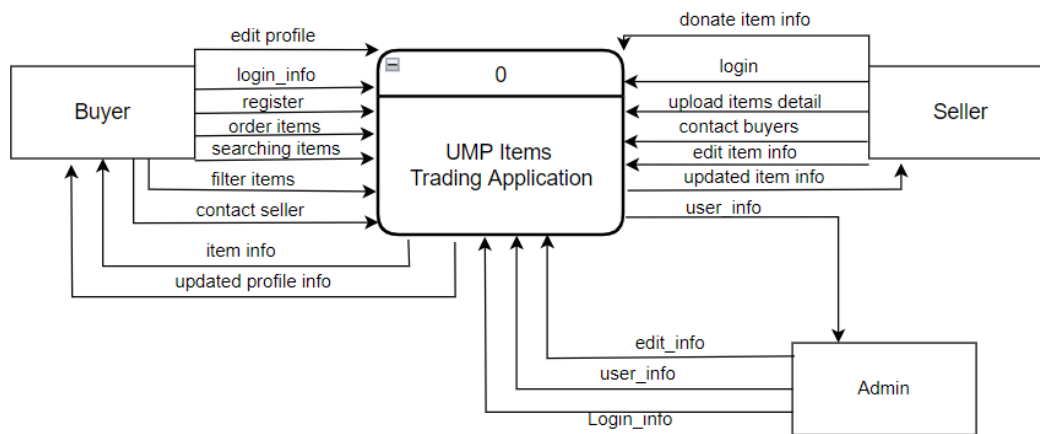


Figure 1.1 Context Diagram of ITA

Based on figure 1.1 shows the Context Diagram (Data Flow Diagram Level 0) of the application. A system context diagram in engineering is a diagram that defines the boundary between a system or part of a system and its environment, showing the entities it interacts with. The diagram is a high-level view of the system. It is similar to a block diagram.

Solution explanation:

- Buyer will register a new account for the application
- Buyer will log in to the application using the username and password that has been registered
- Buyer will order items with the seller by the application
- Buyers will enter new profile information to update their profile information
- Buyer will enter the item name that they want to search
- Buyer will select the filter function to filter the searching items

- g) Buyer will select a method to contact the seller
- h) Buyer will receive the items information after searching and the filtering function is applied
- i) Buyer will receive the updated information of the profile
- j) Seller will enter item detail information for selling the items
- k) Seller will enter the item information that the seller want to donate.
- l) Seller will reply to question from the buyer
- m) Seller will enter new information about the items to update the latest item information
- n) Seller will receive the updated items information after the update by the seller
- o) Admin will receive the user information from the database.
- p) Admin will login to the application by using username and password that specific for admin.
- q) Admin will enter the new user information to add user into database.
- r) Admin will enter the new information of the user to update the user information.

1.4 DATA FLOW DIAGRAM

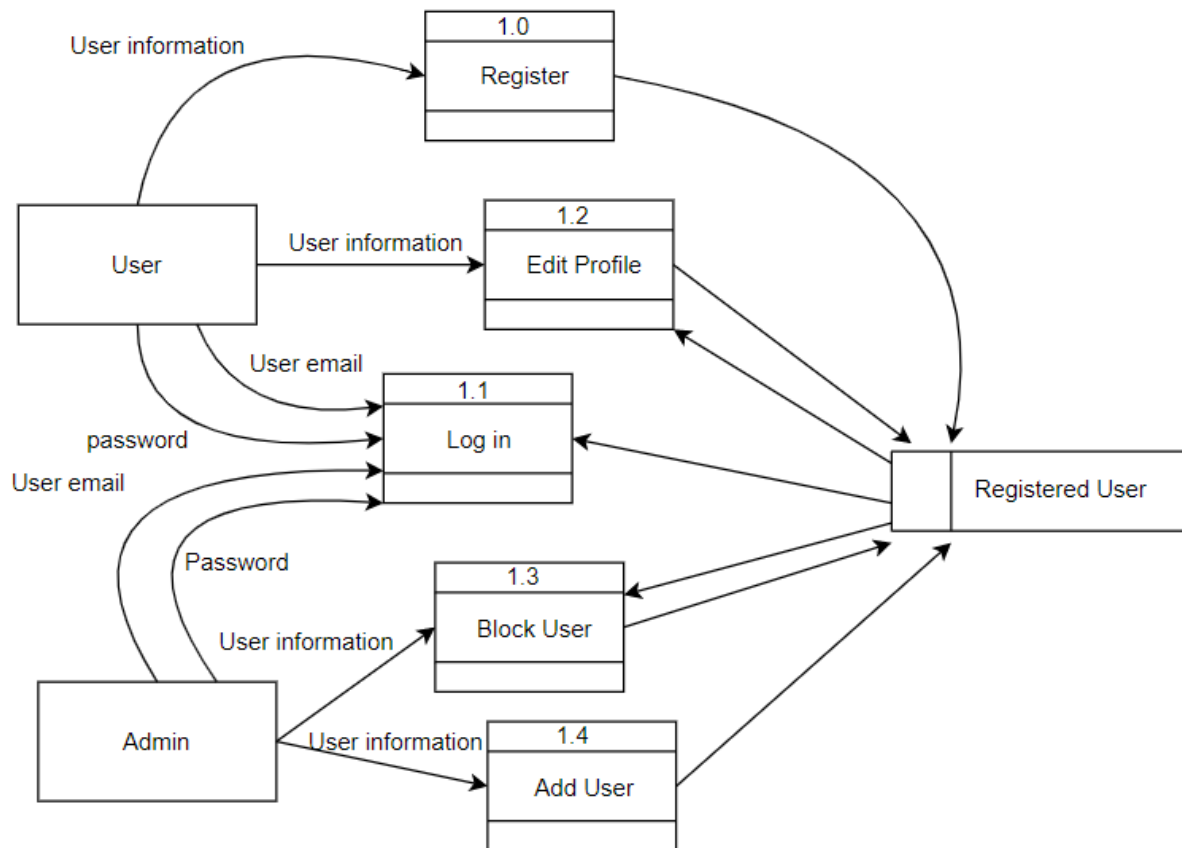


Figure 1.2 Data Flow Diagram Level 1 Register, Login, and Edit Profile

Figure 1.2 shows the data flow diagram level 1 of the user's function register, log-in, edit profile, add user, and block user. Users need to enter the information that are required and the information will save into the user database and users will enter a user email and password to log in to the application with the registered email and password. Users can enter new user information to update their profile. Admin needs to enter the specific user email and password to login to the system. Admin needs to enter the user information to add user to the system. Admin is able to block the user from the system.

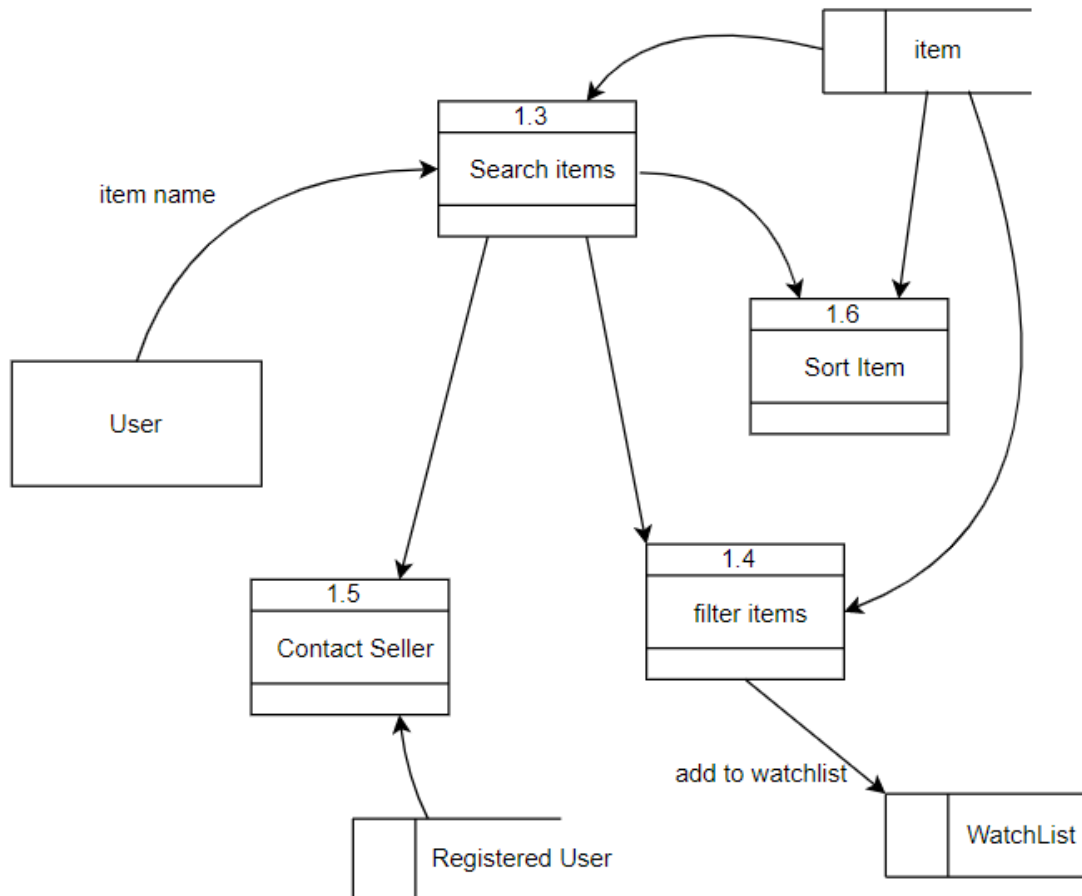


Figure 1.3 Data Flow Diagram Level 1 Search items, Filter items, and Contact Seller

Figure 1.3 shows the data flow diagram level 1 of the function search items, filter item, and Contact Seller. Users need to enter the item title to search for the specific items and users can filter and sorting items. User can add their favorite items to the watchlist and the user also can directly contact to seller.

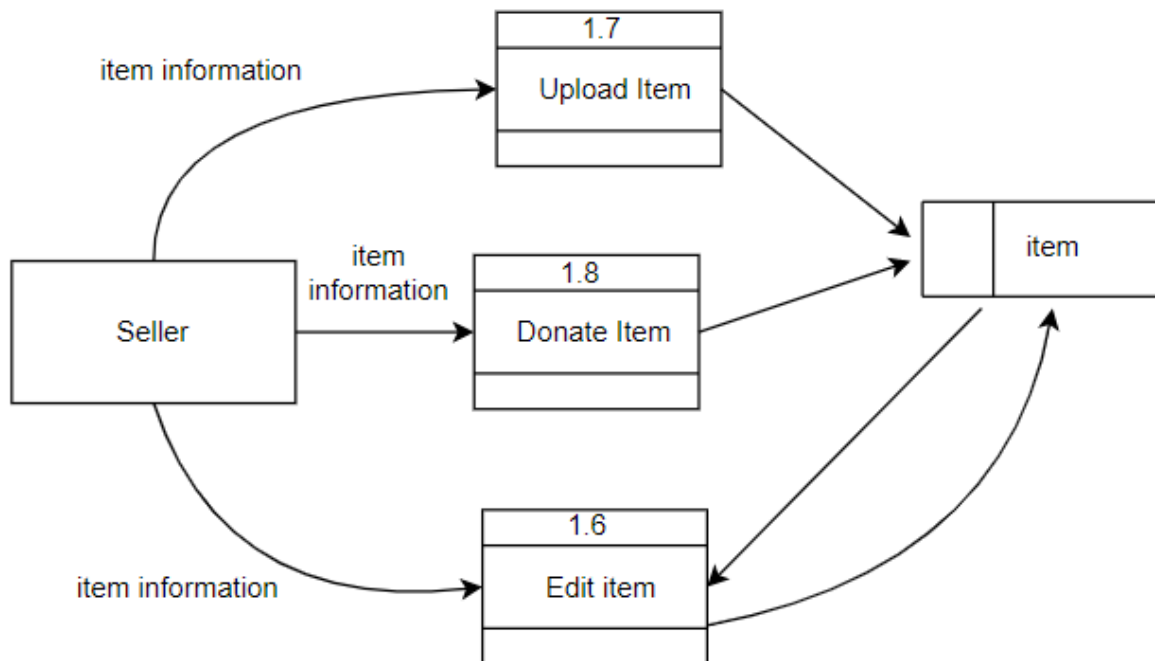


Figure 1.4 Data Flow Diagram Level 1 Upload Item, Edit Item and Donate Item by seller

Figure 1.4 shows the data flow diagram level 1 of the function upload item, edit item, and donate item. Sellers are required to enter the item information detail to upload item, edit item, and donate item information.

2.1 USE CASE DIAGRAM AND DESCRIPTION

2.1.1 Manage Registration

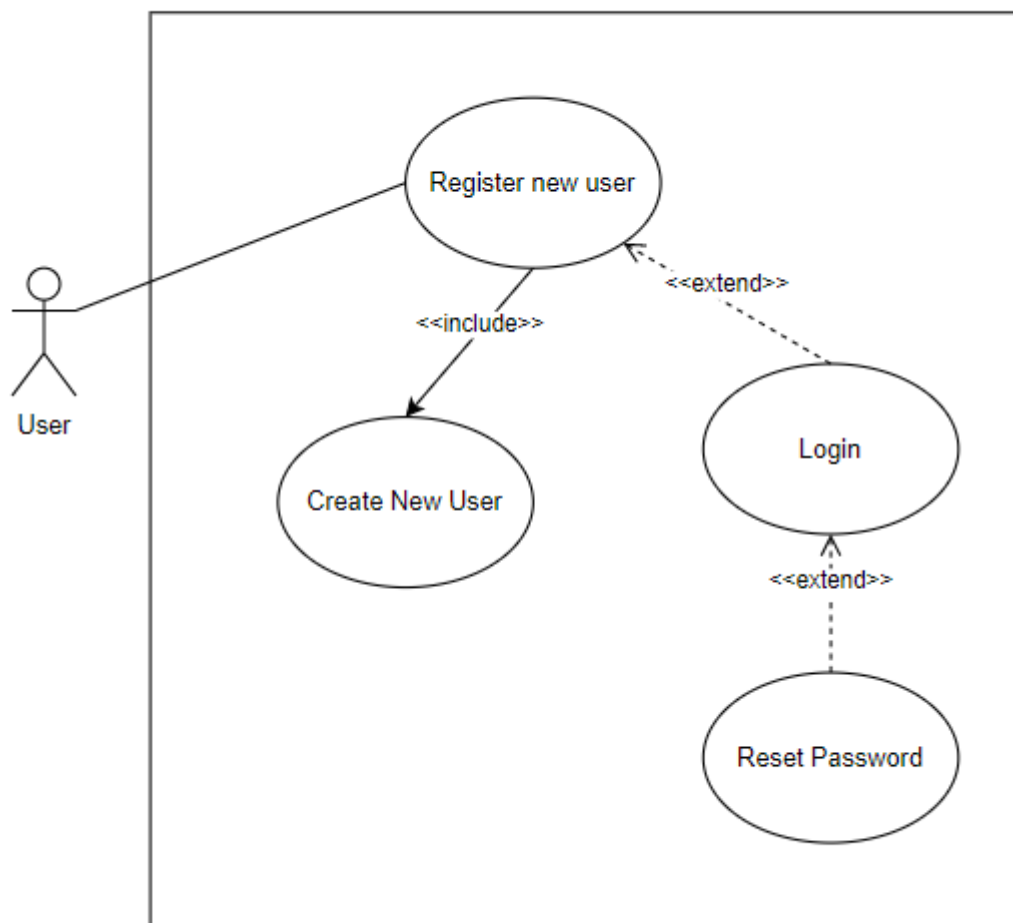


Figure 2.1 Use Case Diagram of Registration

Table 2.1 Manage Registration

Use Case ID	ITA_UCI_1000
Description	This use case describes users registering a new user in the application and users can log in to the application by the registered account.
Actor	User
Pre-Conditions	User must have a valid student id/staff id, password, email, and phone number to register
Basic Flow	<ol style="list-style-type: none"> 1) The use case starts when the user wishes to create an account by clicking the <<Register>> button 2) The system verifies that the user is entering the correct data and the data save in user information [E1 - Incorrect data] 3) The system provides a portal to access the user through a secure channel using login credentials. [A1 – Log in] 4) Users can log in to the application [A2 – Reset Password]

	<p>5) Users can reset their password through the link on the login page</p> <p>6) The use case end.</p>
Alternative Flow	<p>[A1 – Log in]</p> <p>1) The user input the username and password</p> <p>2) The user clicks the <<Log in>> button after inputting data</p> <p>[E2 - Incorrect username or password]</p> <p>3) Continue step 5 in the basic flow</p> <p>[A2 – Reset Password]</p> <p>1) The user inserts the valid email to reset the password</p> <p>2) Continue step 4 in the basic flow</p>
Exception Flow	<p>[E1 - Incorrect data]</p> <p>1. The system displays an error message showing that the registration process is failed.</p> <p>2. The user continues to step 2 in the basic flow.</p> <p>[E2 - Incorrect username or password]</p> <p>1. The system displays an error message, showing that the login process is failed.</p>

	2. The user continues to step 2 in the basic flow.
Post-Condition	<ul style="list-style-type: none">• The system provides a portal to access the user through a secure channel using login credentials.
Rules	-
Constraints	<ul style="list-style-type: none">• User id and password just type in the correct blank space.

2.1.2 Manage Profile

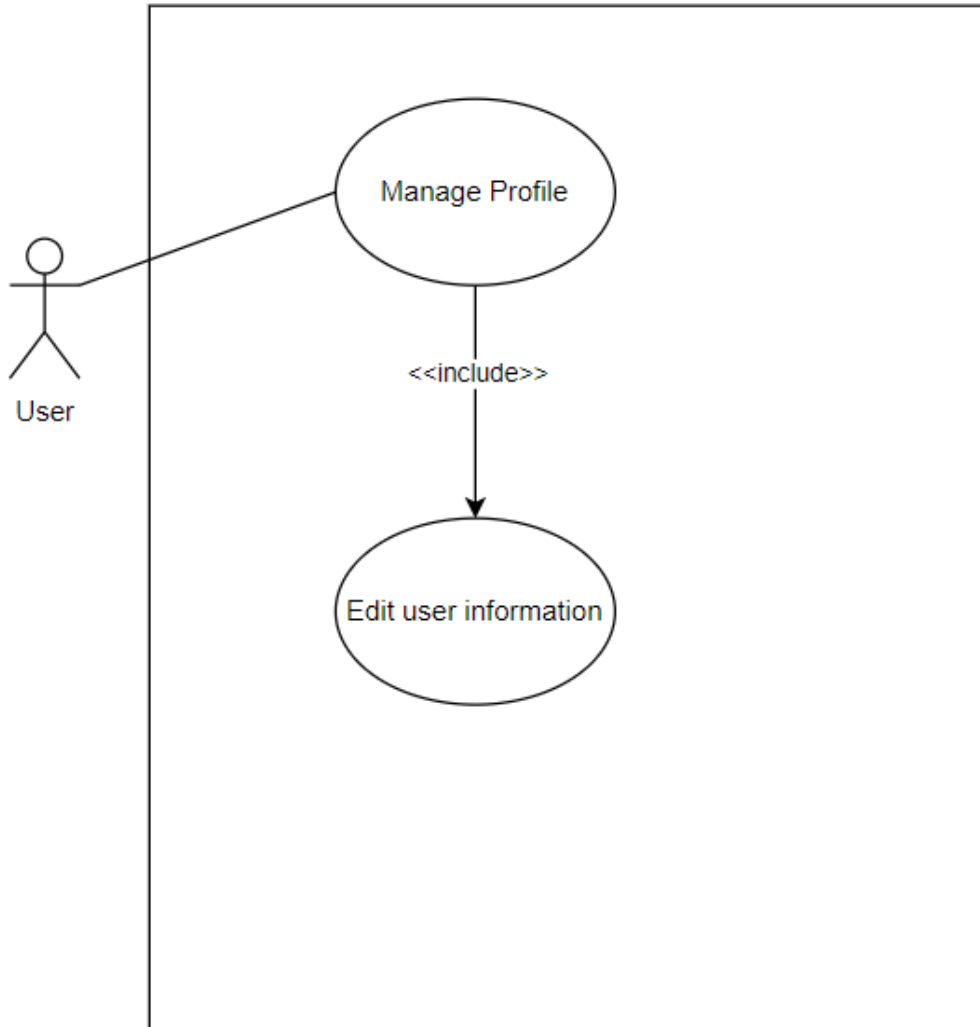


Figure 2.2 Use Case Diagram of Manage Profile

Table 2.2 Manage Profile

Use Case ID	ITA_UCI_2000
Description	This use case describes users manage their profiles to view and edit the profile information.
Actor	User
Pre-Conditions	<ul style="list-style-type: none">• User must have an account• User needs to log in to the account before viewing and editing the profile
Basic Flow	<ol style="list-style-type: none">1) The use case starts after users login into their account2) User needs to click the person icon to view the profile information3) Users can view or enter new information for updating the profile <p>[E1 - Incorrect data]</p> <ol style="list-style-type: none">4) The use case ends
Alternative Flow	None
Exception Flow	<p>[E1 - Incorrect data]</p> <ol style="list-style-type: none">1. The system displays an error message showing that the updating profile process is failed.

	2. The user continues to step 2 in the basic flow.
Post-Condition	<ul style="list-style-type: none">• System will update the new information in the application
Rules	<ul style="list-style-type: none">• Users can only edit their profile of themselves
Constraints	<ul style="list-style-type: none">• User unable to edit the user email in the profile

2.1.3 Manage Goods

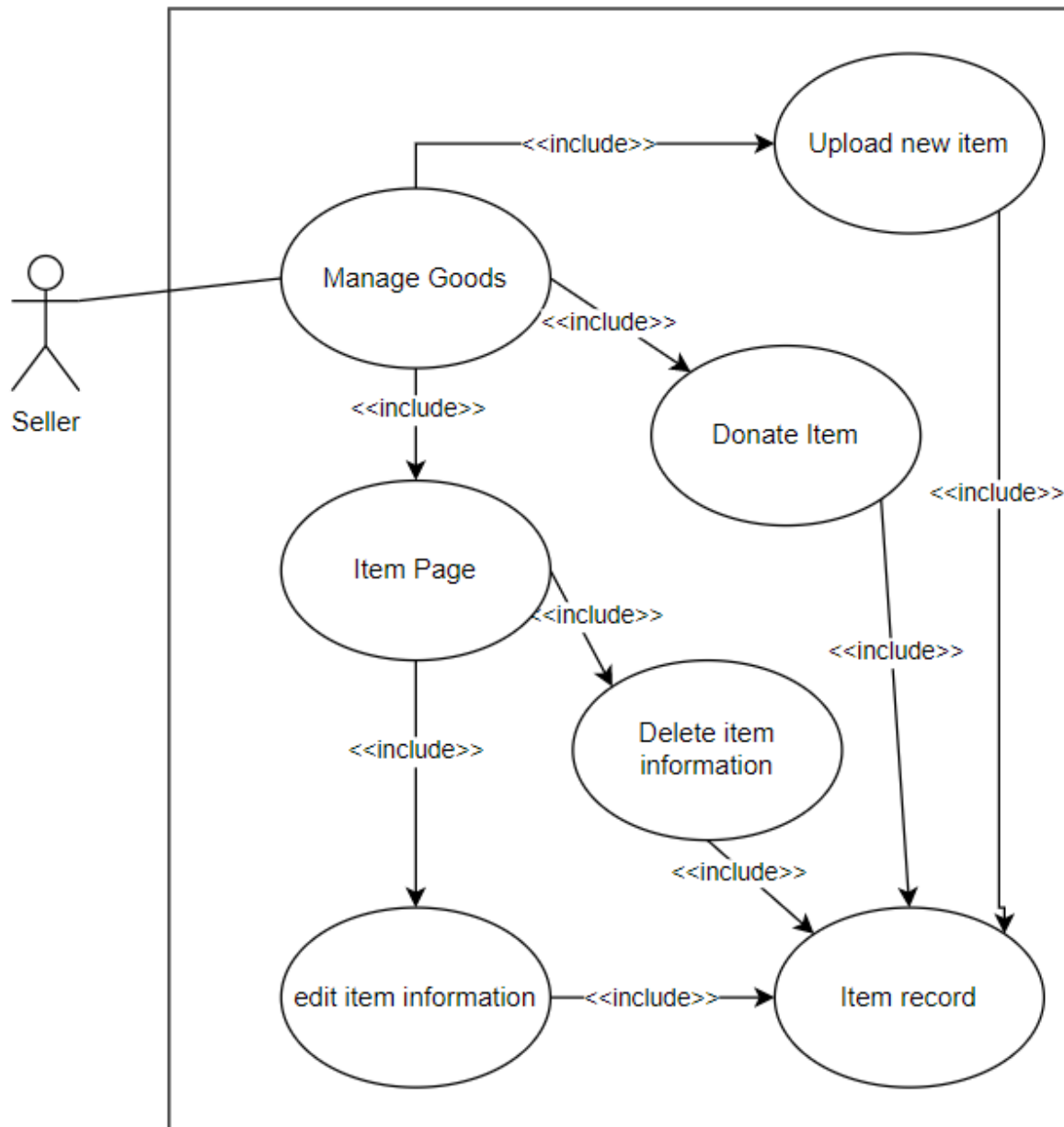


Figure 2.3 Use Case Diagram of Manage Trades

Table 2.3 Manage Goods

Use Case ID	ITA_UCI_3000
Description	This use case describes seller manage their trade items by uploading and editing item information.
Actor	Seller
Pre-Conditions	<ul style="list-style-type: none"> • Seller must have an account • Seller needs to log in to the account before uploading, deleting and editing item information
Basic Flow	<ol style="list-style-type: none"> 1) The use case starts after clicking the <<sell>> button on the main page 2) The seller can choose: <ul style="list-style-type: none"> - Click on the <<Upload New Item>> button to upload the item to the application database <li style="text-align: center;">[A1: Upload new item] - Click on the <<Item Page>> button to link to the item page in the application <li style="text-align: center;">[A2: Item Page] - Click <<Edit>> button to edit an item in the application database <li style="text-align: center;">[A3: Edit item]

	<ul style="list-style-type: none"> - Click on the <<Delete>> button to delete an item in the application database <p style="text-align: center;">[A4: Delete item]</p> <ul style="list-style-type: none"> - Click on the <<Donate Item>> button to add a donation item. <p style="text-align: center;">[A5: Donate item]</p> <p>3) The use case ends</p>
Alternative Flow	<p>[A1: Upload new item]</p> <ol style="list-style-type: none"> 1) The use case starts when the user clicks on the <<Upload New Item>> button 2) Seller needs to select a photo or snap a photo as an item picture. 3) Seller requires to insert all required information of the item. 4) Click <<Upload>> to upload the item to the application 5) Continue step 3 in the base flow <p>[A2: Item Page]</p> <ol style="list-style-type: none"> 1) The use case starts when seller click <<Item Page>> button 2) Seller can click: <ul style="list-style-type: none"> - <<Edit>> Button <p style="text-align: center;">[A3: Edit item]</p> <ul style="list-style-type: none"> - <<Delete Button>>

[A4: Delete item]

- Seller can modify status of the item

3) Continue Step 1 in Alternative 2

[A3: Edit item]

- 1) The use case starts when the user clicks on the <<Edit Item>> button
- 2) Seller can enter or edit information of the items
- 3) Click <<Update>> to update the information on the existing item.
- 4) Continue step 3 in the base flow

[A4: Delete item]

- 1) The use case starts when the user clicks on the <<Delete>> button
- 2) Click <<Delete>> to delete the existing item.
- 3) Continue step 3 in the base flow

[A5: Donate item]

- 1) The use case starts when the user clicks on the <<Donate Item>> button
- 2) Seller needs to select a photo or snap a photo as an item picture
- 3) Seller requires to insert all required information of the item
- 4) Click <<Upload>> to upload the item to the application
- 5) Continue step 3 in the base flow

Exception Flow	None
Post-Condition	<ul style="list-style-type: none">• New item was uploaded to application database• New item information was updated by the seller
Rules	<ul style="list-style-type: none">• Seller only can edit the items that uploaded by themselves
Constraints	-

2.1.4 Manage Order

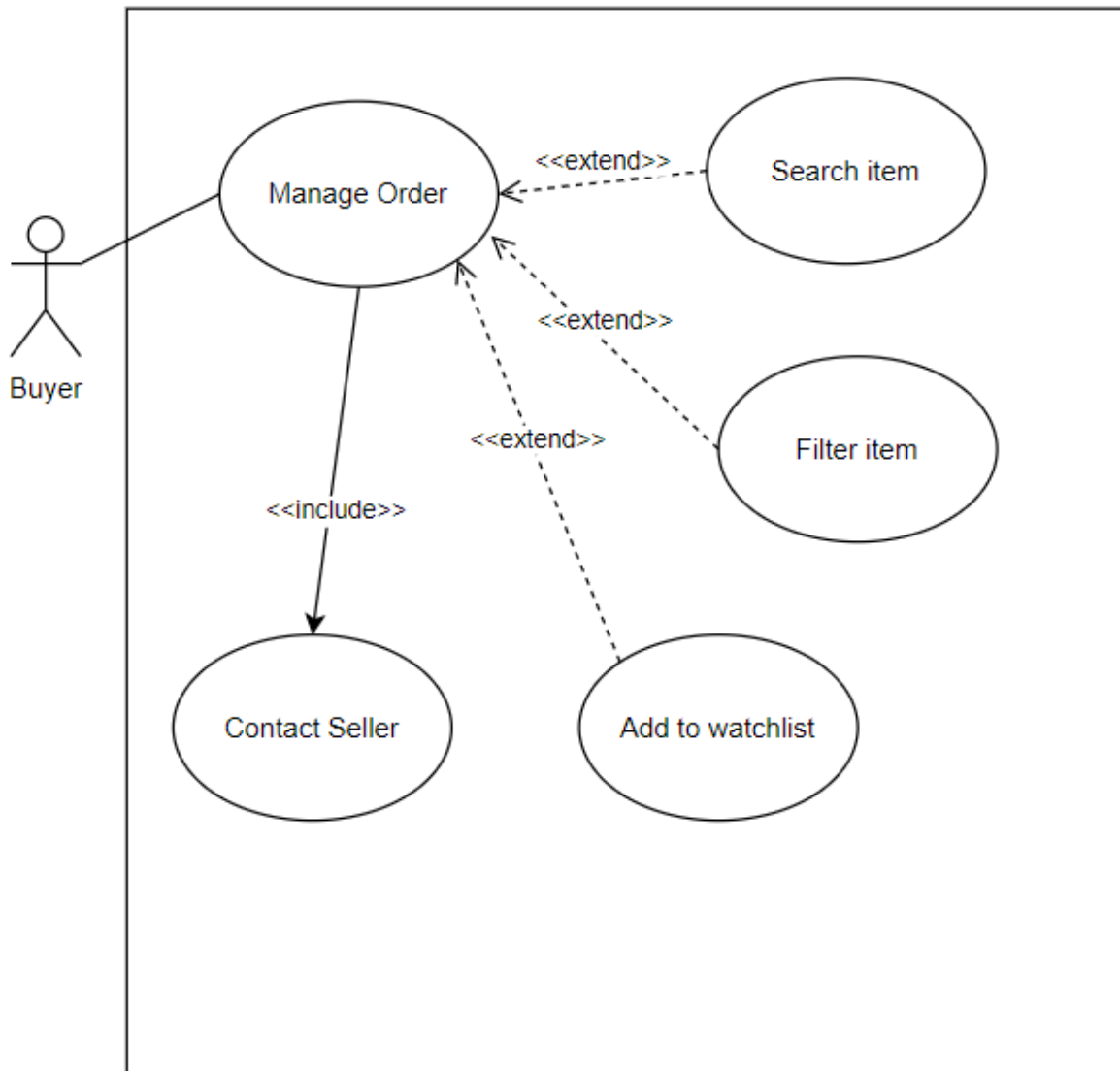


Figure 2.4 Use Case Diagram of Manage Order

Table 2.4 Manage Order

Use Case ID	ITA_UCI_4000
Description	This use case describes buyers manage orders in the application
Actor	Buyer
Pre-Conditions	<ul style="list-style-type: none"> • Buyer must have an account • Buyer needs to log in to the account before manage orders by the buyers.
Basic Flow	<ol style="list-style-type: none"> 1. The user case starts after the user log in to the application 2. The buyer can choose: <ul style="list-style-type: none"> - Search items by entering item names into the search box for searching items. [A1: Search item] - Filter items by selecting the type of filter prepared by the application [A2: Filter item] - Add to the watchlist by clicking the icon of the watchlist [A3: Add item to watchlist] - Contact Seller by selecting the contact method prepared by the application [A4: Contact Seller]

	3. Use case ends
Alternative Flow	<p>[A1: Search item]</p> <ol style="list-style-type: none">1) The use case starts when the user clicks the search on the search box2) Enter the item title in the search box3) The application will display the information of the searching item4) Continues step 2 in the basic flow <p>[A2: Filter item]</p> <ol style="list-style-type: none">1) The use case starts when the user selects a filtering function like filter by categories, condition, and so on.2) The application will display the information of the item that is filtered3) Continues step 2 in the basic flow <p>[A3: Add item to watchlist]</p> <ol style="list-style-type: none">1. The use case starts when the user clicks the watchlist icon2. The items will add to the watchlist3. Continues step 2 in the basic flow <p>[A4: Contact Seller]</p>

	<ol style="list-style-type: none">1) The use case starts when the user wants to purchase an item from the seller2) The buyer can select the method to contact the seller using the Call, SMS, and email provided by the application.3) Continues step 2 in the basic flow
Exception Flow	None
Post-Condition	<ul style="list-style-type: none">• Buyer able contact to seller• Buyer able to add the item to the watchlist• Buyer able to search and filter items.
Rules	-
Constraints	-

2.1.5 Manage User

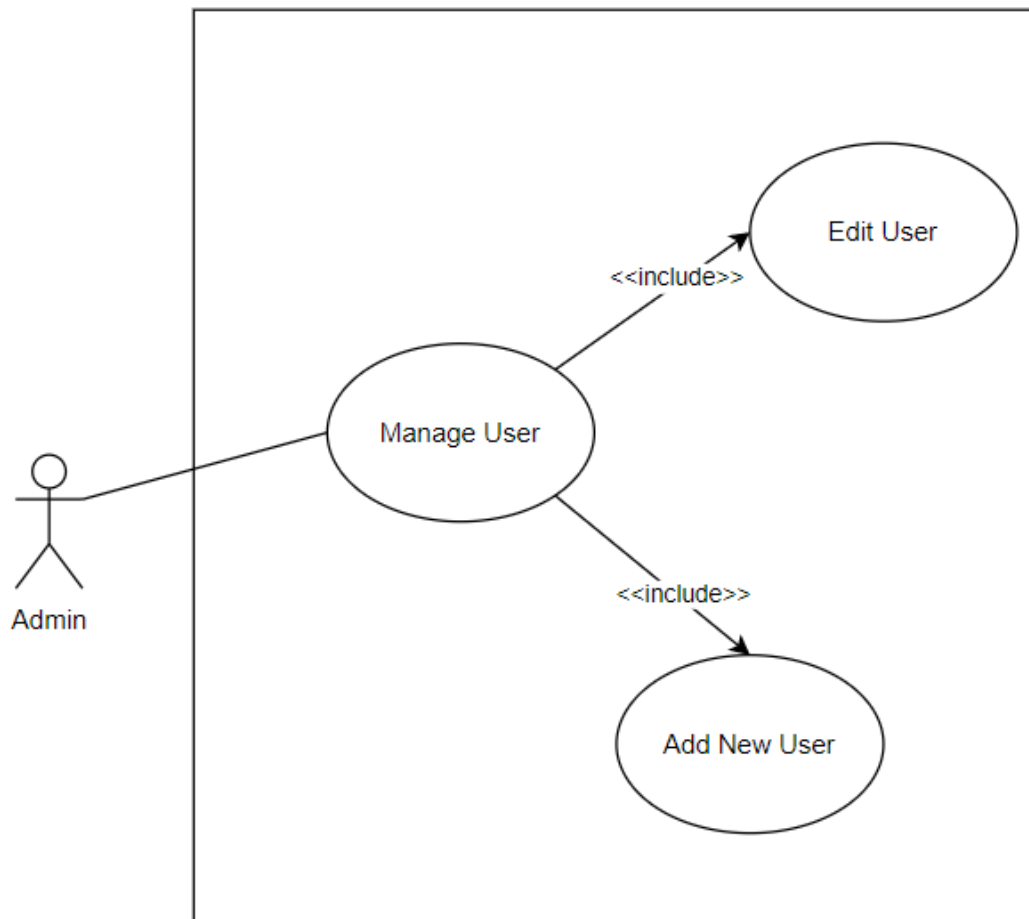


Figure 2.5 Use Case Diagram of Manage User

Table 2.5 Manage User

Use Case ID	ITA_UCI_5000
Description	This use case describes admin to manage the user.
Actor	Buyer
Pre-Conditions	<ul style="list-style-type: none"> Admin had login to the system
Basic Flow	<ol style="list-style-type: none"> The use case starts when the admin login to the system. The admin able to: <ul style="list-style-type: none"> Add new user [A1: Add new user] Edit user [A2: Edit user] The use case end
Alternative Flow	<p>[A1: Add new user]</p> <ol style="list-style-type: none"> Admin click the <<Add new user>> button Entering the user information Click the <<Add>> button Continues step 2 in basic flow <p>[A2: Edit user]</p>

	<ol style="list-style-type: none">1) Admin click the <<Edit>> button2) Select user status which is block or free3) Click the <<Edit>> button4) Continues step 2 in basic flow
Exception Flow	None
Post-Condition	<ul style="list-style-type: none">• Admin able to add new user• Admin able to edit user information
Rules	-
Constraints	-

2.2 SEQUENCE DIAGRAM

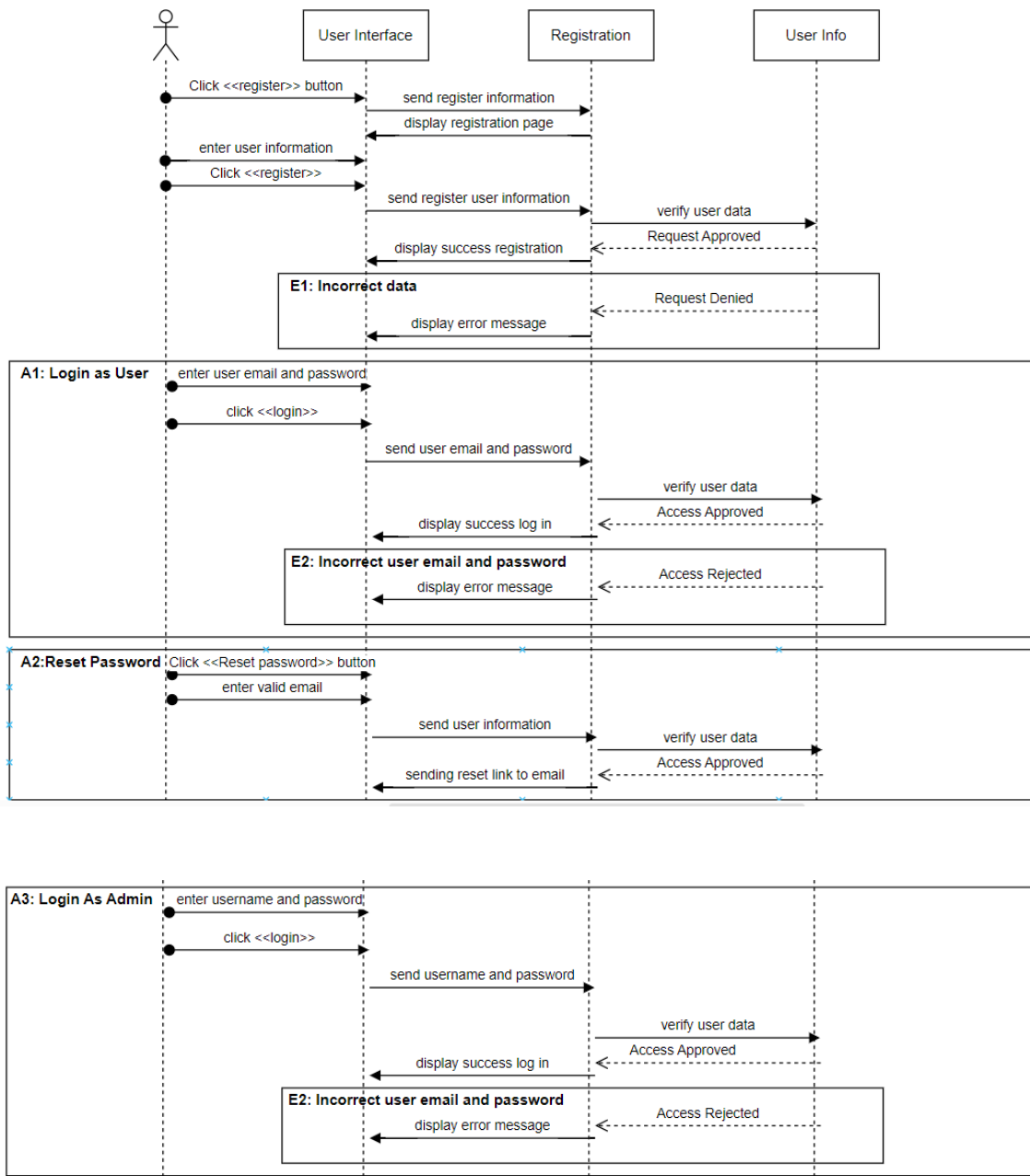


Figure 2.6 Sequence Diagram of Manage Registration

Figure 2.6 shows Sequence Diagram of manage registration in the application. Users need to click the Register button to link to the registration page. Users need to enter all the user information that is required and click the register button to send the user's information to the database. The system will verify user data if all the information is fulfilled system will display a successful message and if there is some error information like repeated id it will display an error message for the registration. Manage registration function also included for the login function which allow user and admin login.

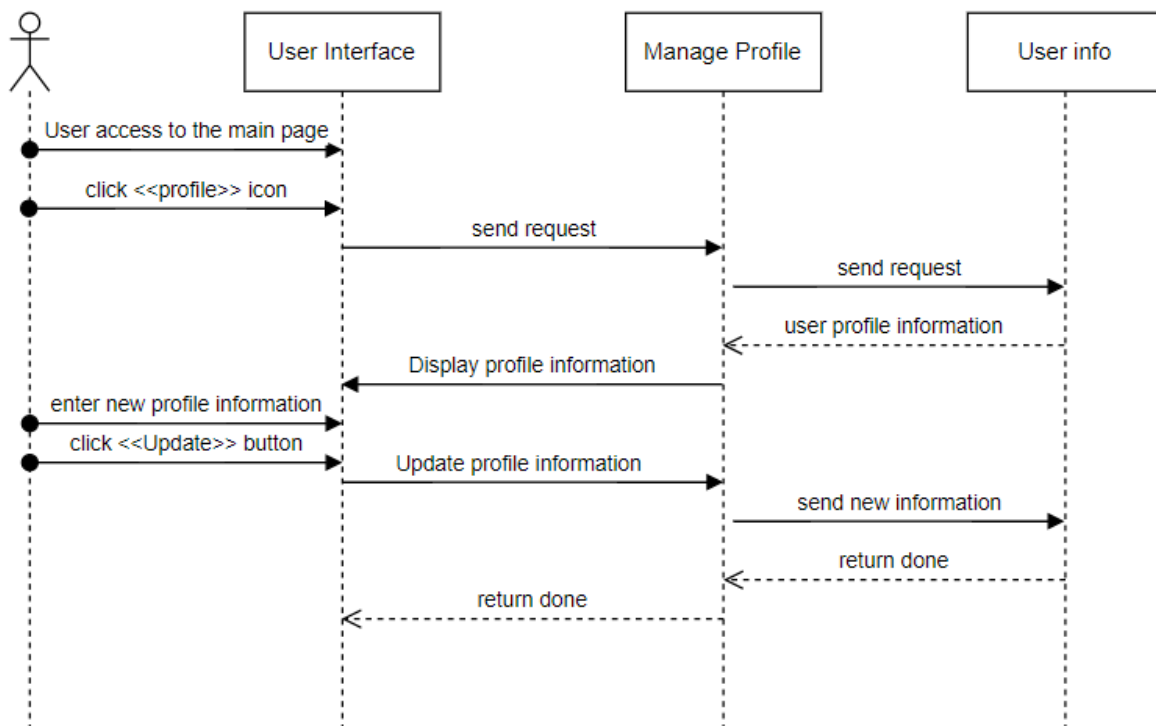


Figure 2.7 Sequence Diagram of Manage Profile

Figure 2.7 shows Sequence Diagram of Manage Profile in the application. Users are access to the main page by using the login function. Users need to click the profile icon to get the user information data and display it on the profile page. Users can enter new user information like email, phone number, and so on. Users can click the update button to update all the new information that is entered. The system will return a successful message when the new information is uploaded.

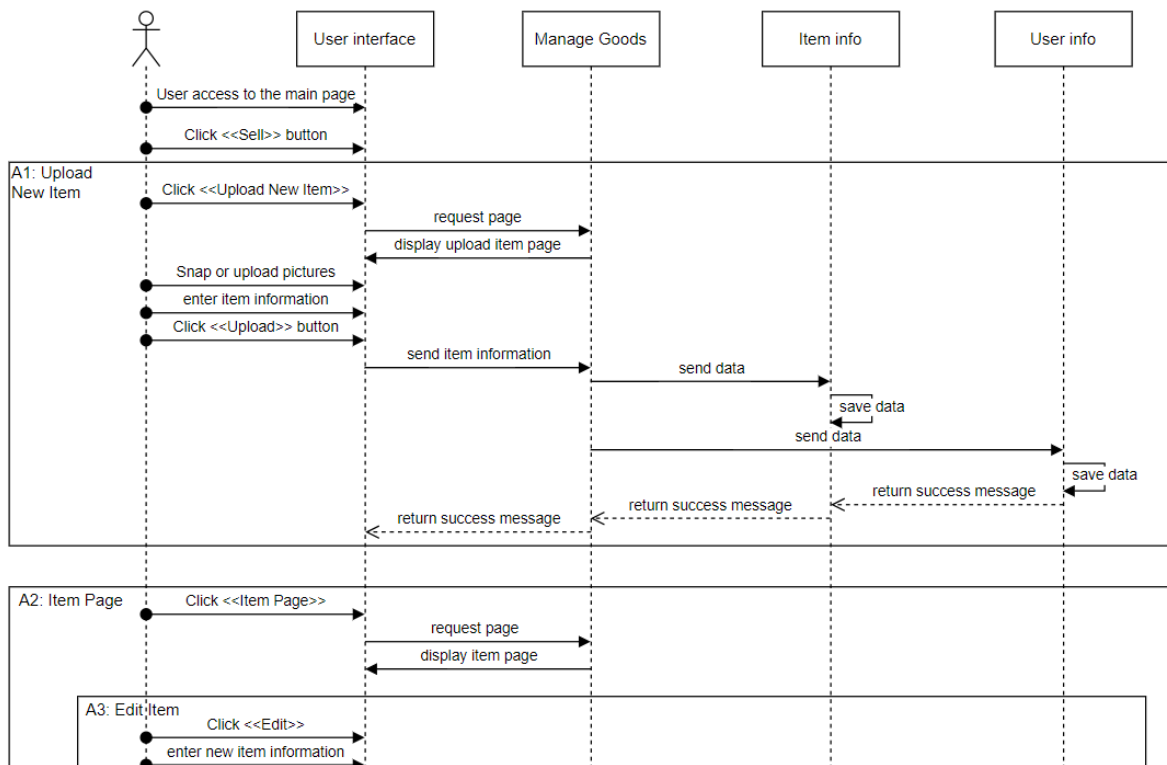


Figure 2.8 Sequence Diagram of Manage Goods with Alternative 1

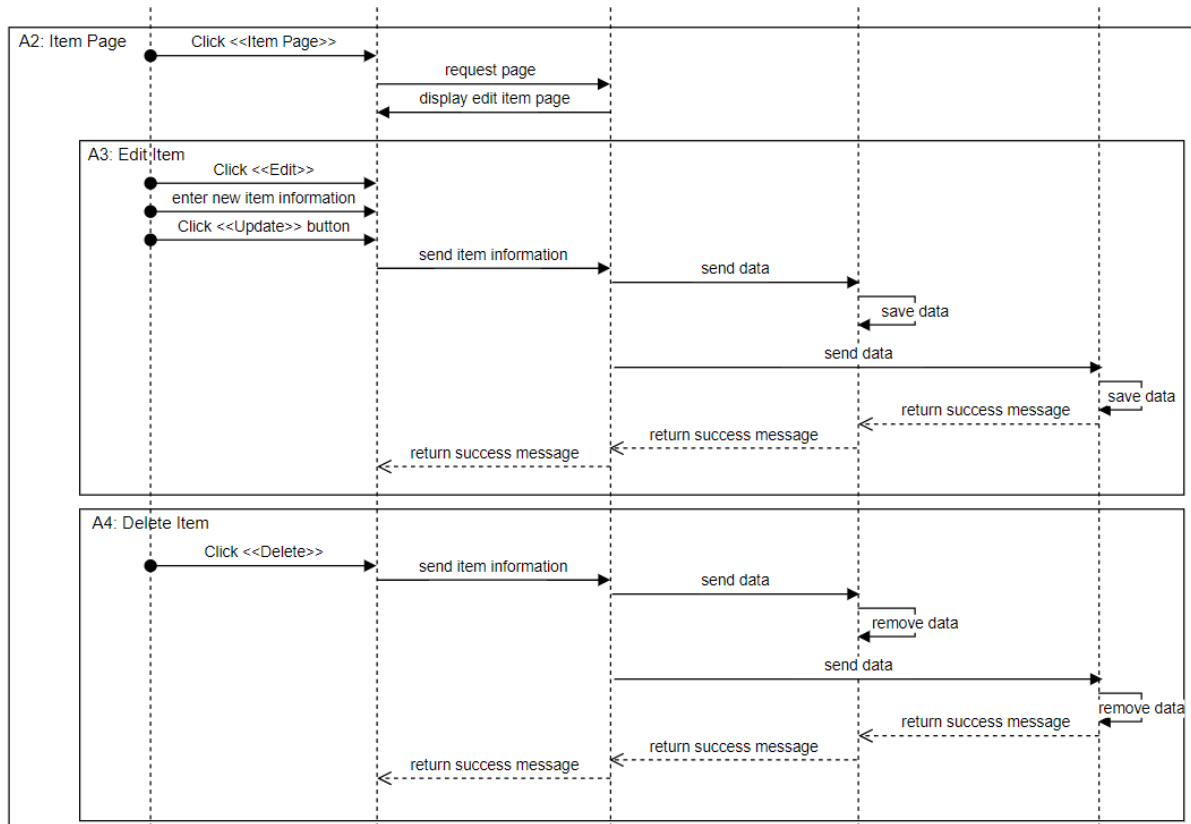


Figure 2.9 Sequence Diagram of Manage Goods with Alternatives 2, 3 & 4

Figure 2.8 and Figure 2.9 shows the Sequence of Manage Trade and its Alternative in the application. Users are access to the main page by using the login function. Users need to click the sell button to link to the manage trade page. Users can select the upload new item button to insert a new item for sale. Users are required to snap pictures or select pictures from the gallery and users are required to enter the information about the new item. After entering all the required data, users can click the upload button to send the item information to the item database and user database. The system will send a successful message when the item information is inserted into the database.

Users can select the edit item button to edit items' information. Users are required to select the item that needs to be edited and it will display the edit item page to let users edit the items' information. Users can click the update button to send the item information to the database. The system will send a successful message when the item information is updated into the database.

Users can select the delete item button to delete items' information. Users are required to select the item that needs to be deleted. Users can click the delete button to delete the item from the database. The system will send a successful message when the item information is deleted from the database.

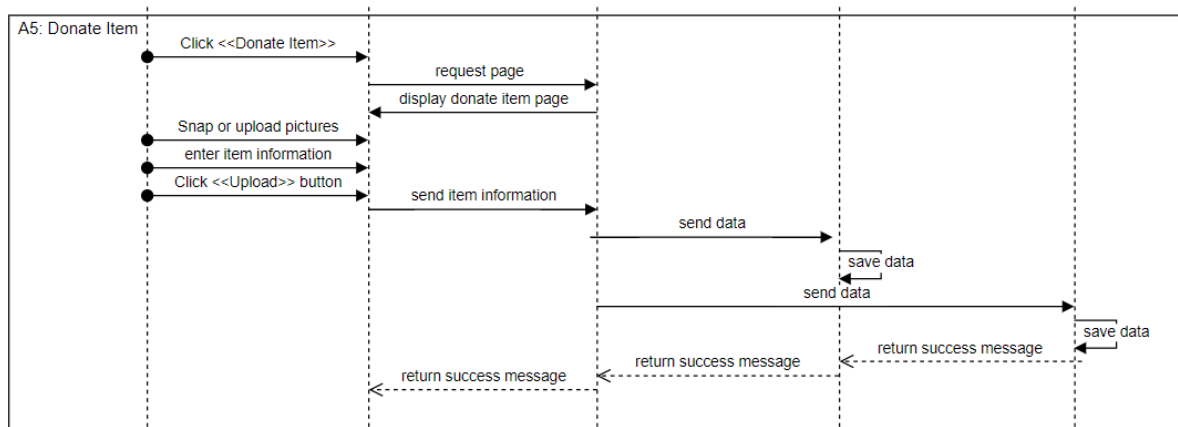


Figure 2.10 Sequence Diagram of Manage Goods with Alternatives 5

Figure 2.10 shows the Sequence Diagram of Manage Goods with Alternatives 5. Users can select the Donate Item button to donate an item. Users are required to enter the item information for the donated item. Users can click the upload button to send the item information to the database. The system will send a successful message when the item information is stored in the database.

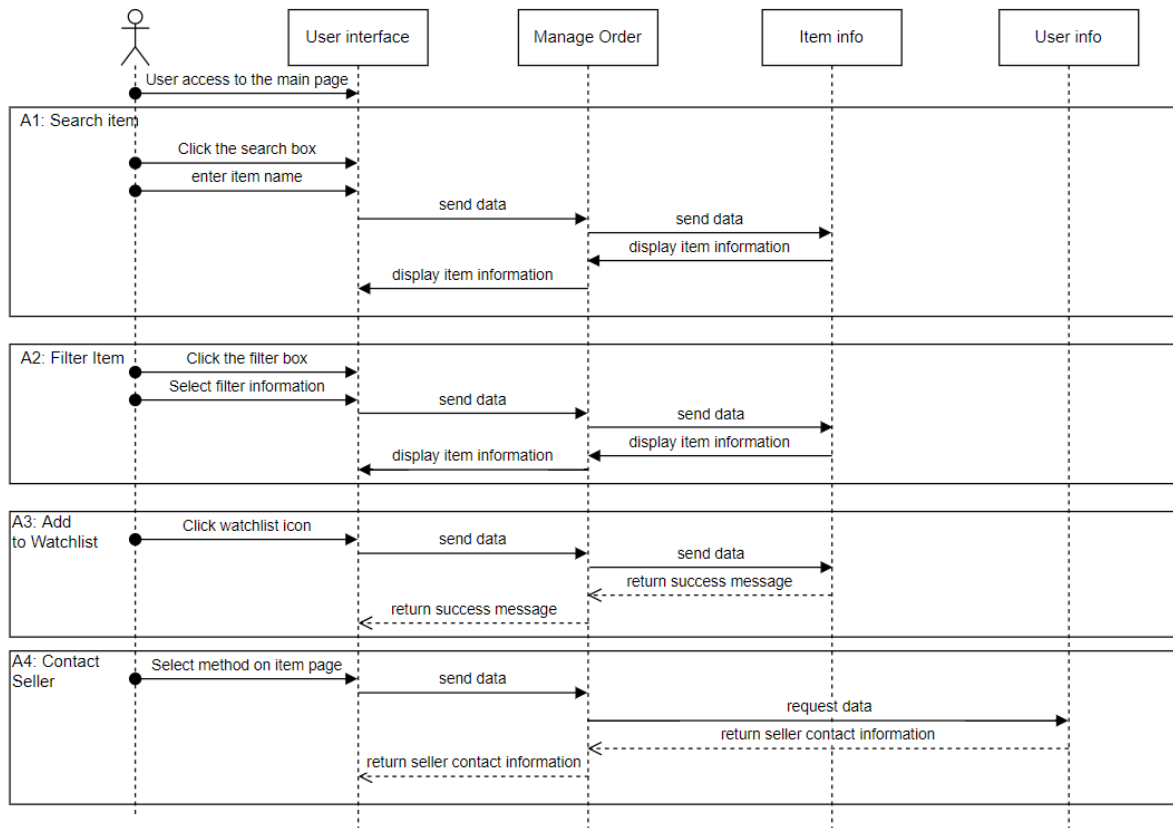


Figure 2.11 Sequence Diagram of Manage Order

Figure 2.11 shows the Sequence of Manage Order and its Alternative in the application. Users are access to the main page by using the login function. Users can search for an item by clicking the search box on the main page and entering the item name into the search box. The system will display the item information. Users can use the filter item function by clicking the filter box and selecting the filter information like categories, prices, and so on. The system will display the filtered item information. Users can click the watchlist icon on the item page to add the item to the watchlist and return a success message. Users can select the contact method on

the item page like SMS, call, and so on to directly contact the seller. The system will request the contact information and return the seller’s contact information to let users contact the seller.

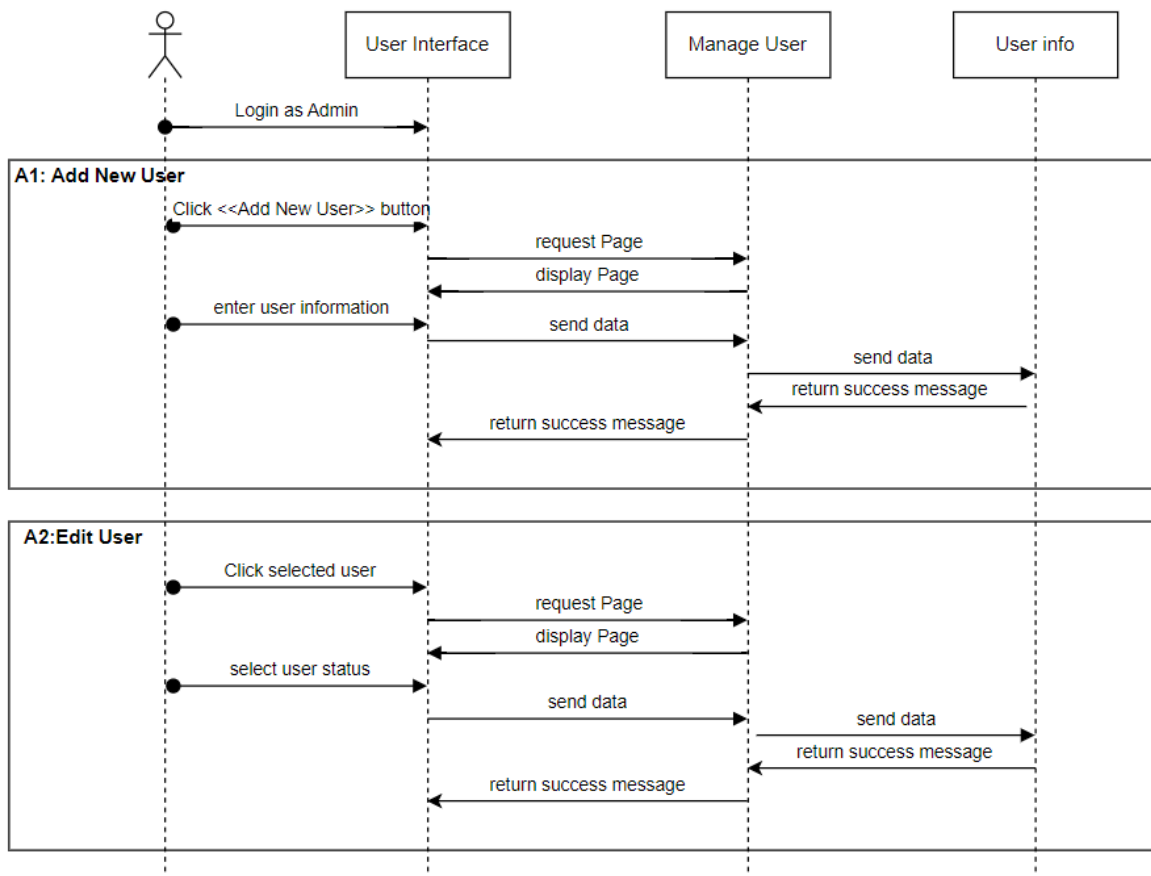
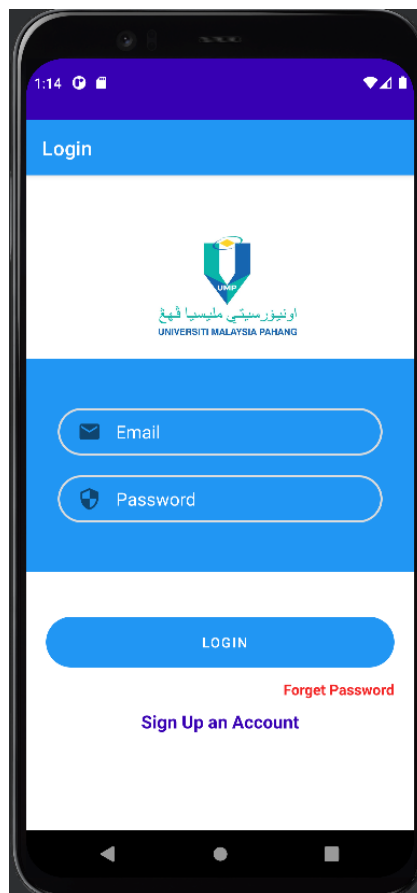


Figure 2.12 Sequence Diagram of Manage User

Figure 2.12 shows the Sequence of Manage User and its Alternative in the application. Admin can access the admin page by using the login function. The admin can view the user information in the admin page. Admin can add a new user by clicking the add new user button and entering the user information to add the user to the system. Admin can edit user status by select the status and click the edit button.

3 CHAPTER 3

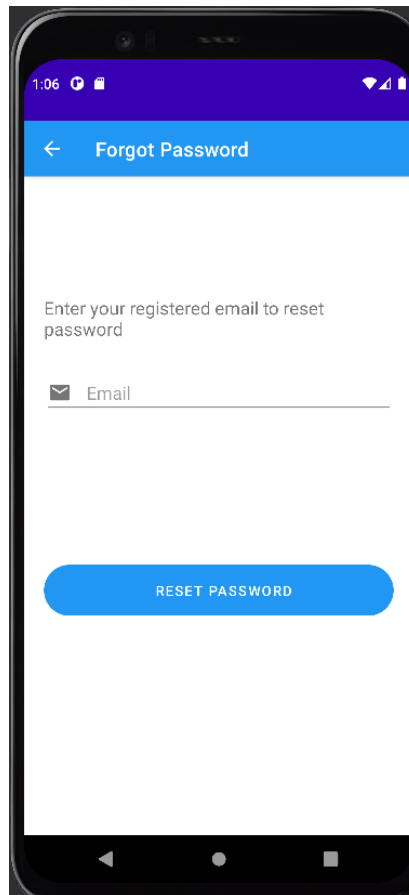
3.1 INTERFACE DESIGN



Appendix - A 1: Login Page

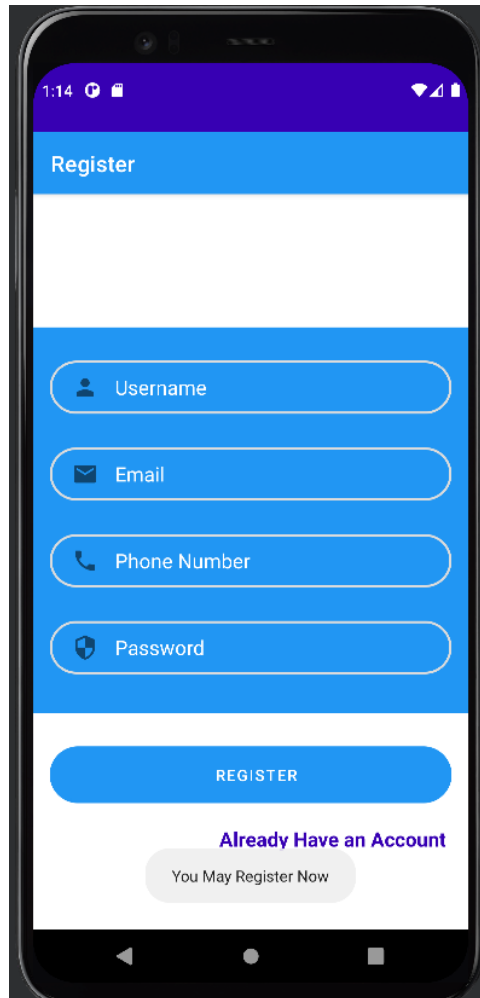
This is the first page that will appear after user click application icon in user device. User are required to enter their username and password to log in to the application. The <<Login>> button is used to log in to the application after users enter the user email

and password. When click Forget Password text it will link to Appendix A-2: Forget Password Page and if click the Sign Up an Account text it will link to Appendix A-3: Register Page.



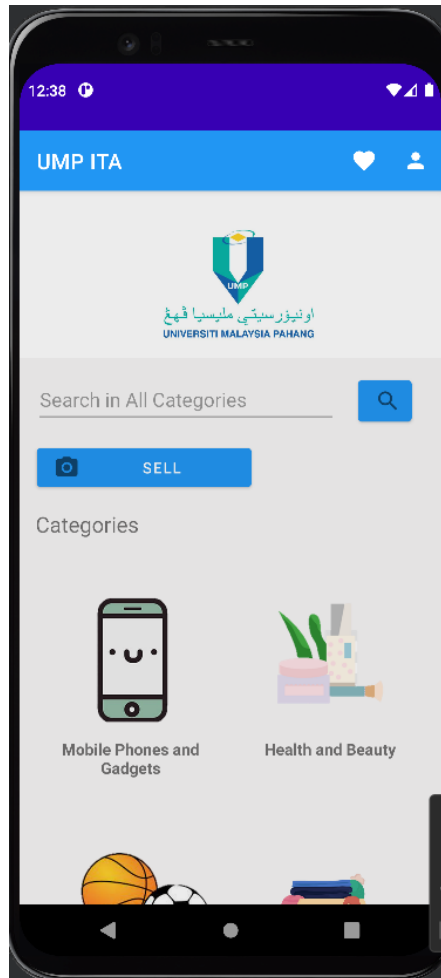
Appendix - A 2: Forget Password Page

When the users click the Forget Password text in Appendix A-1: Login Page, it will directly link to this page. Users are required to enter the email that had registered and click the Reset Password button to reset the password of the user account. After clicking the button the application will send a link to email for the resetting the password in the application.



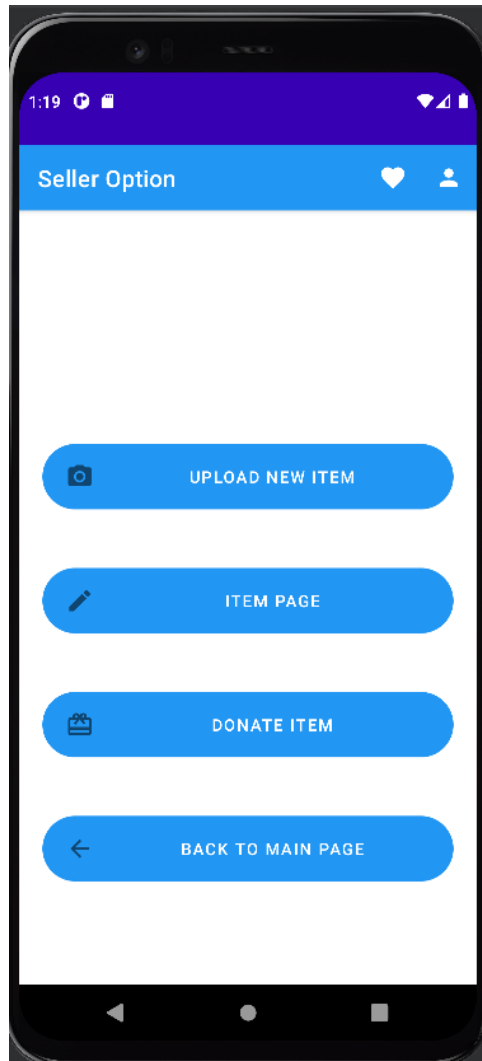
Appendix - A 3: Register Page

When the users click the Sign Up an Account text in Appendix A-1: Login Page, it will directly link to this page. The user is required to enter the username, email, phone number, and password to register an account. After entering all the required information, the user can click the Register button to register an account. If the user clicks the Already Have an Account text it will link to Appendix A-1: Login Page.



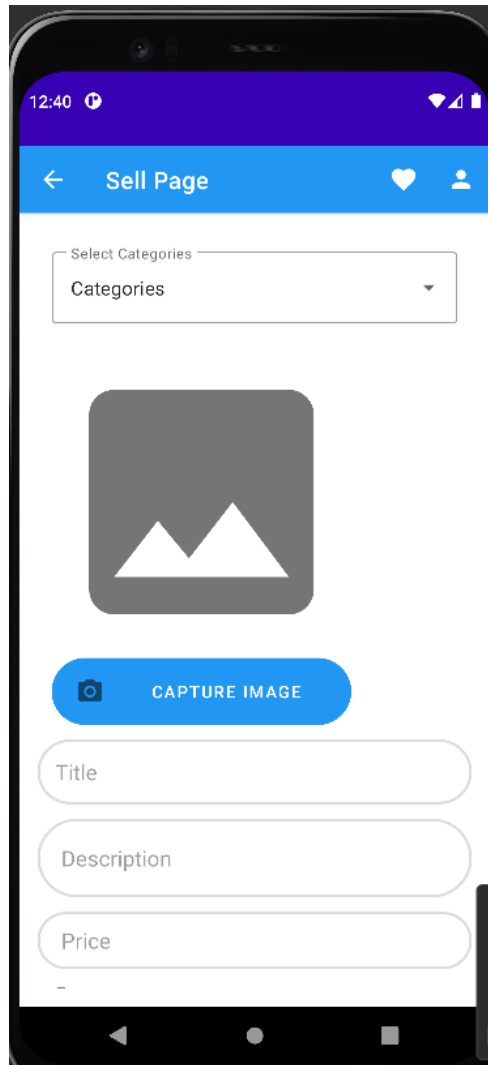
Appendix - A 4: Main Page

When the users entering the username and password and click the Login button in Appendix A-1: Login Page, it will verify the user information into the database if all the data are verified it will link to this page. There has a search box to let users enter the item name and search for it and it will link to Appendix A-9: Search Page. Same as the button below the Categories text which will link to Appendix A-9: Search Page by using filter by the category the selected. If the users click the Sell button, it will link to Appendix A-5: Sell Option Page which able users to upload a new item and view the item that users had uploaded.



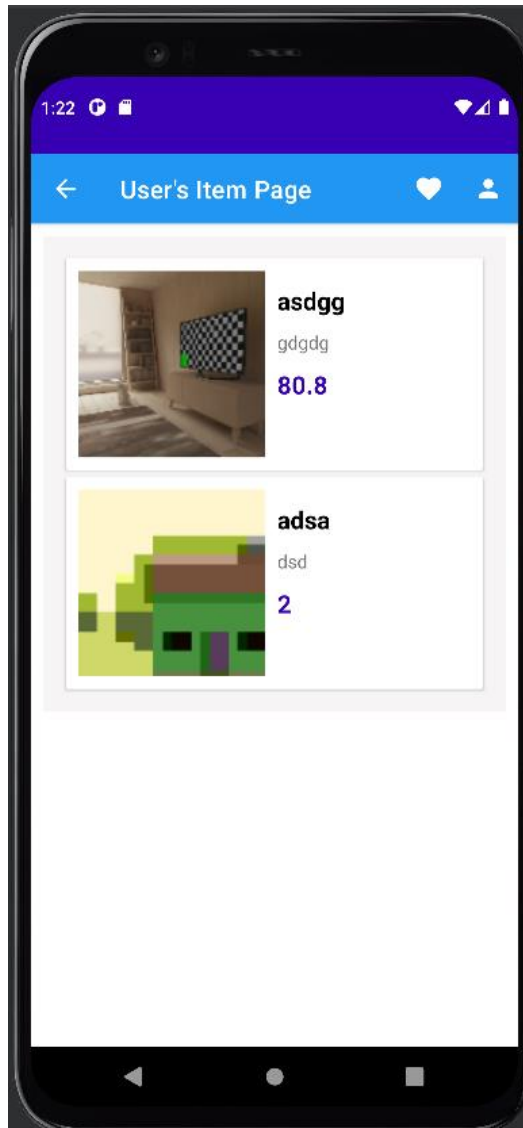
Appendix - A 5: Sell Option Page

When the users click the Sell button in Appendix A-4: Main Page, it will directly link to this page. Users can upload a new item by clicking the Upload New Item button which links to Appendix A-6: Upload New Item Page and viewing the item that had been uploaded by the user which links to Appendix A-7: Item Page. Users are able to donate items by clicking the Donate Item button and links to Appendix A-17: Donate Page.



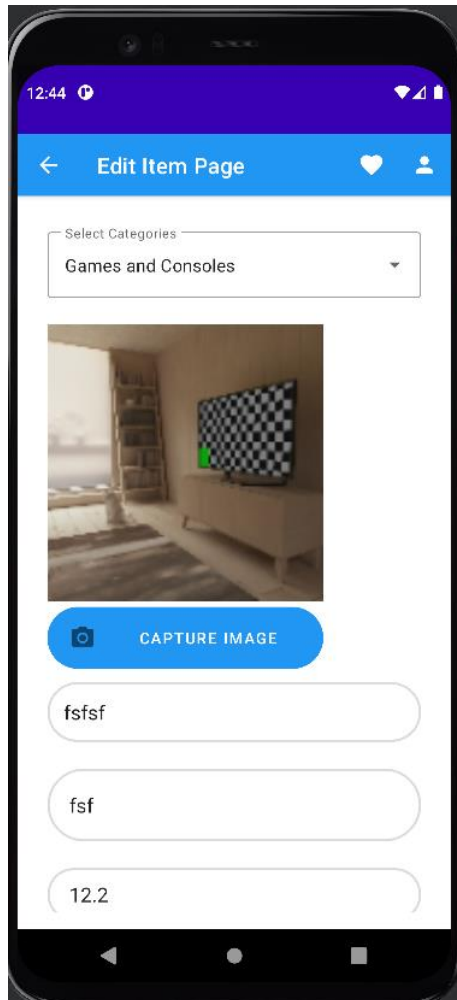
Appendix - A 6: Upload New Item Page

When the users click the Upload New Item button in Appendix A-5: Sell Option Page, it will directly link to this page. The user is required to enter item information and capture pictures or select pictures from the gallery. After entering all the information, the user can click the Upload button to upload the item information to the database.



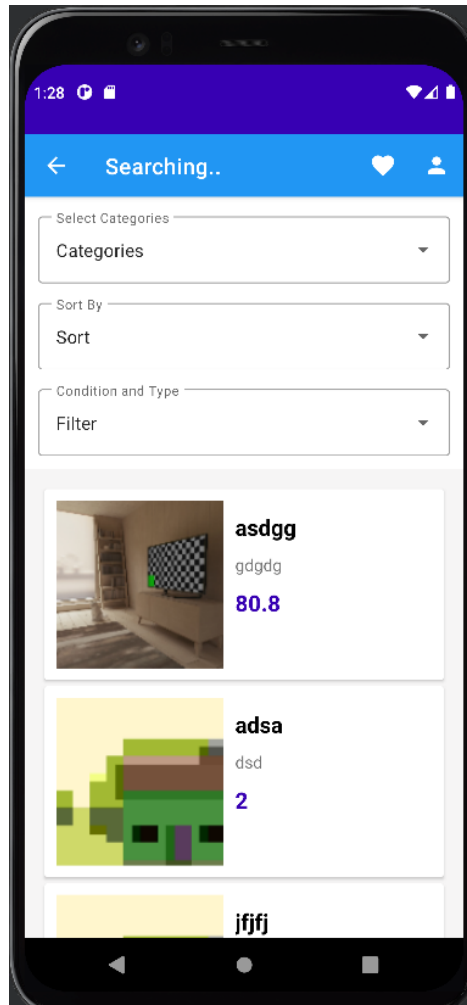
Appendix - A 7: Item Page

When the users click the Item Page button in Appendix A-5: Sell Option Page, it will directly link to this page. The user can click the item card to redirect to the edit item page.



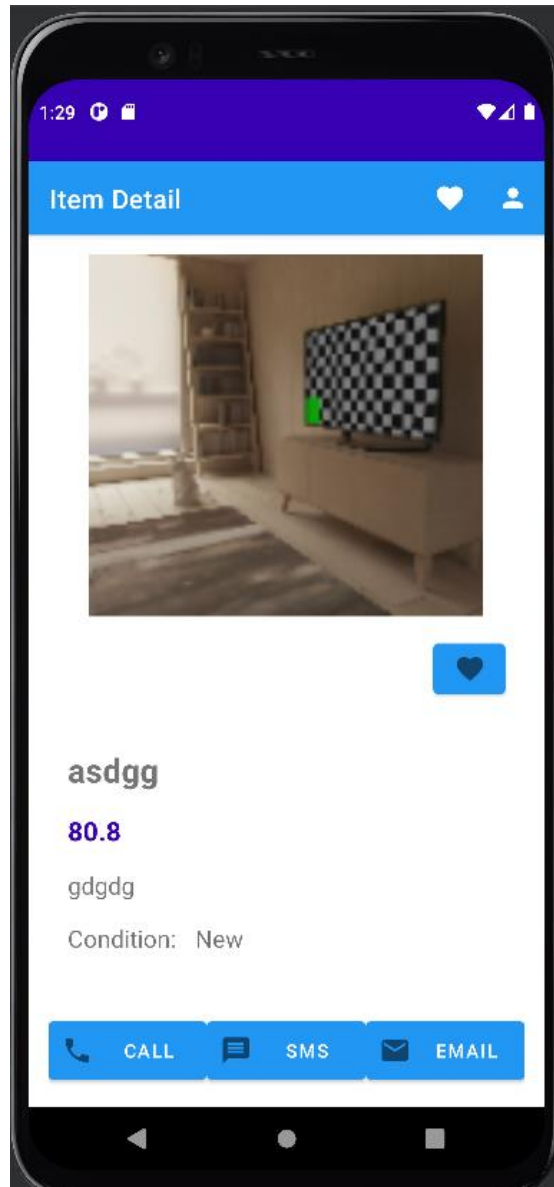
Appendix - A 8: Edit Item Page

When the users click the Edit button in Appendix A-7: Item Page, it will directly link to this page. The system will display all the information of the item like category, image, title, description, price, condition, type and status of the item. User can enter the new item information and click the update information to update the item information. User also allow to click the remove item button to remove the item from the application. After the action it will redirect to Appendix – A 5: SellOptionPage



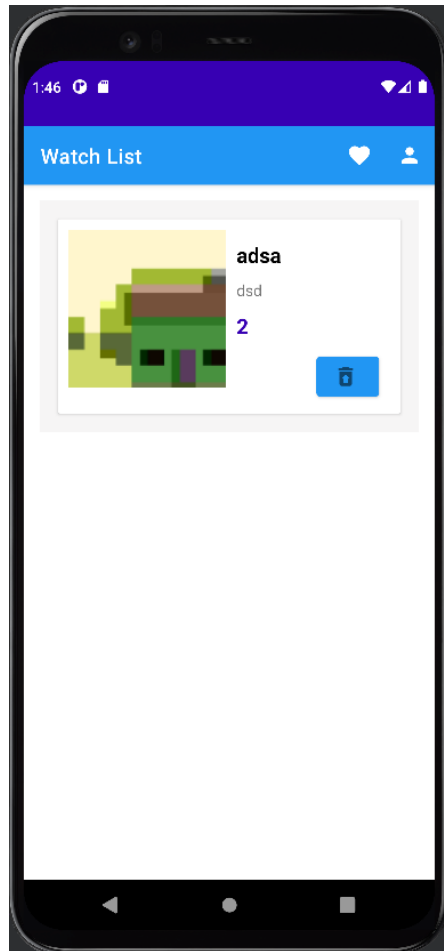
Appendix - A 9: Search Page

When the users enter the item name and click the Search button in Appendix A-4: Main Page, it will directly link to this page. Users can filter the item by selecting the category, condition, and type. Users also can sort the item by selecting the sort. Users can click the image and it will directly link to Appendix A-10: Item Detail Page.



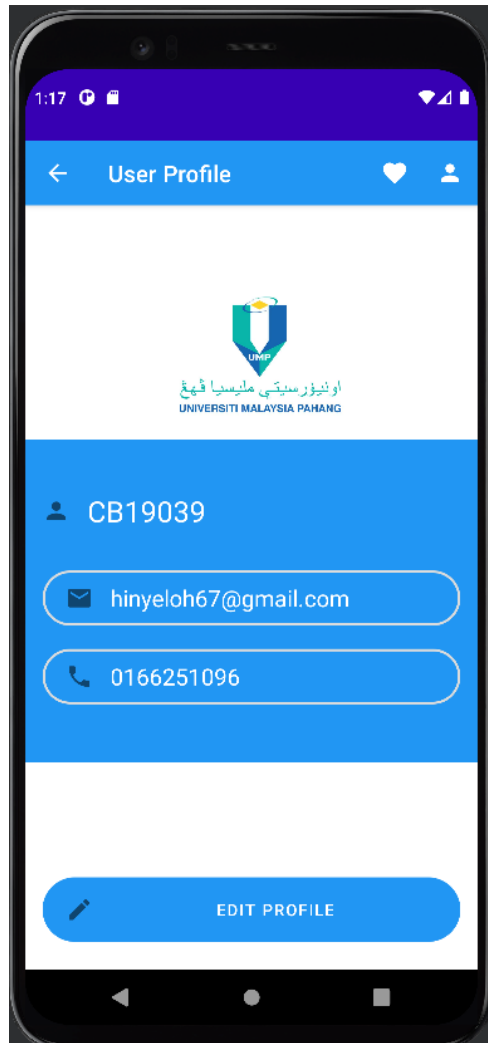
Appendix - A 10: Item Detail Page

When the click the image in Appendix A-9: Search Page, it will directly link to this page. Users can select the method for contacting sellers like call, SMS, and email. Users can view all the information of the item that selected by user. User can click the favor button to add the item to the watch list.



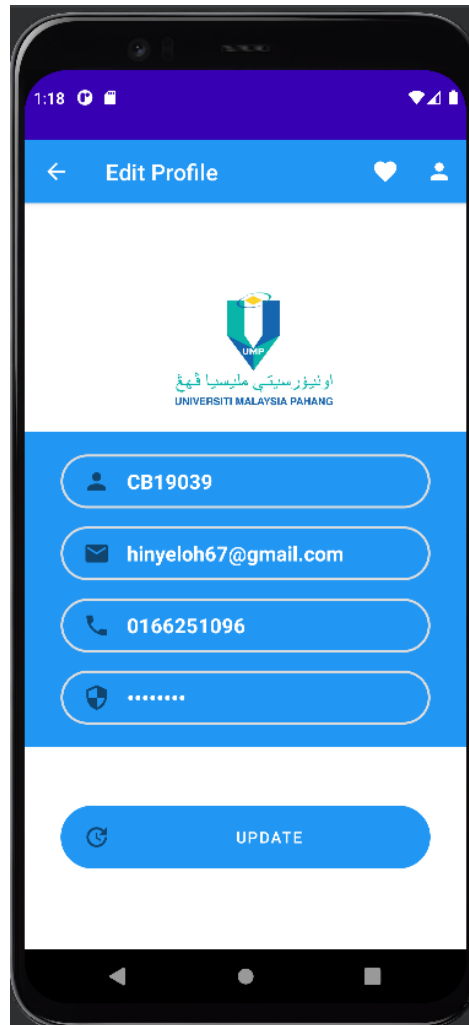
Appendix - A 11: Watch List Page

When the click the favour icon in action bar, it will directly link to this page. Users can view all the items that were added to the watch list. Users can remove the item from the watch list by clicking the remove button. Users can click the image and link to Appendix A-10: Item Detail Page.



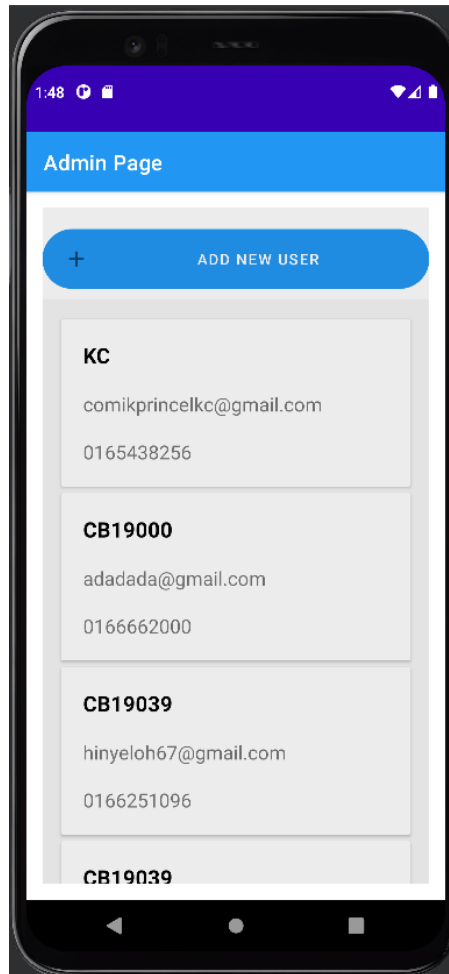
Appendix - A 12: Profile Page

When the click the person icon in action bar, it will directly link to this page. Users can view the email and phone number of the user. Users can edit the profile information by clicking the Edit Profile button and linking to Appendix A-13: Edit Profile Page.



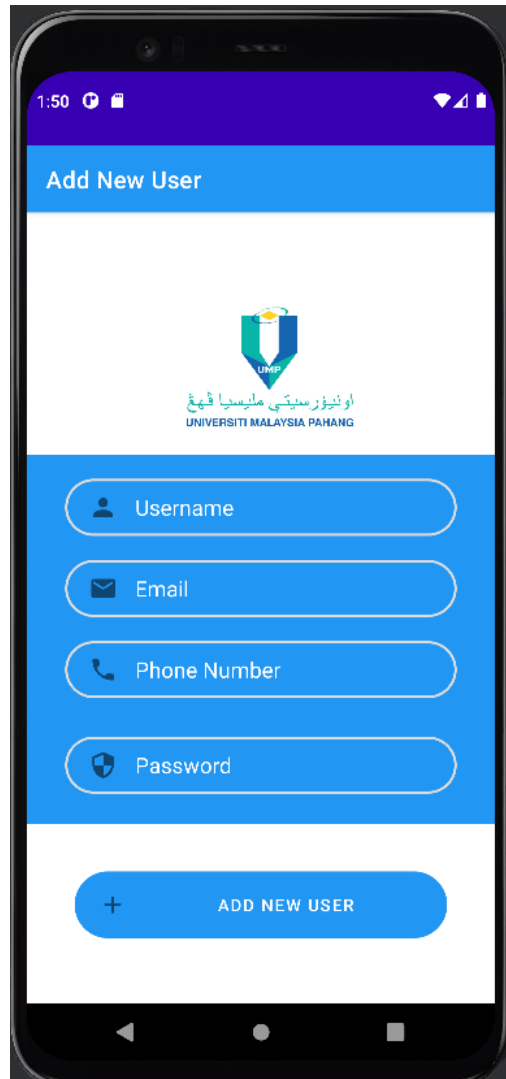
Appendix - A 13: Edit Profile Page

When the click the Edit Profile button in Appendix A-12: Profile Page, it will directly link to this page. Users can enter the new username, and phone number to update the user information and click the Update button to update the user profile.



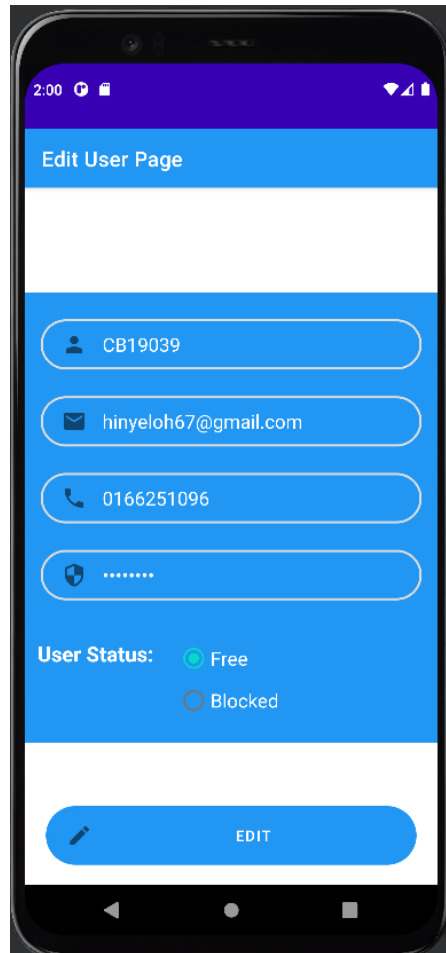
Appendix - A 14: Admin Page

When the click the admin logged in to the system, it will directly link to this page. Admin can view the user information on this page. Admin can click the add new user button and link to Appendix A-15: Add new user. Admin can click the user to link to Appendix A-16: Edit user to edit the user status.



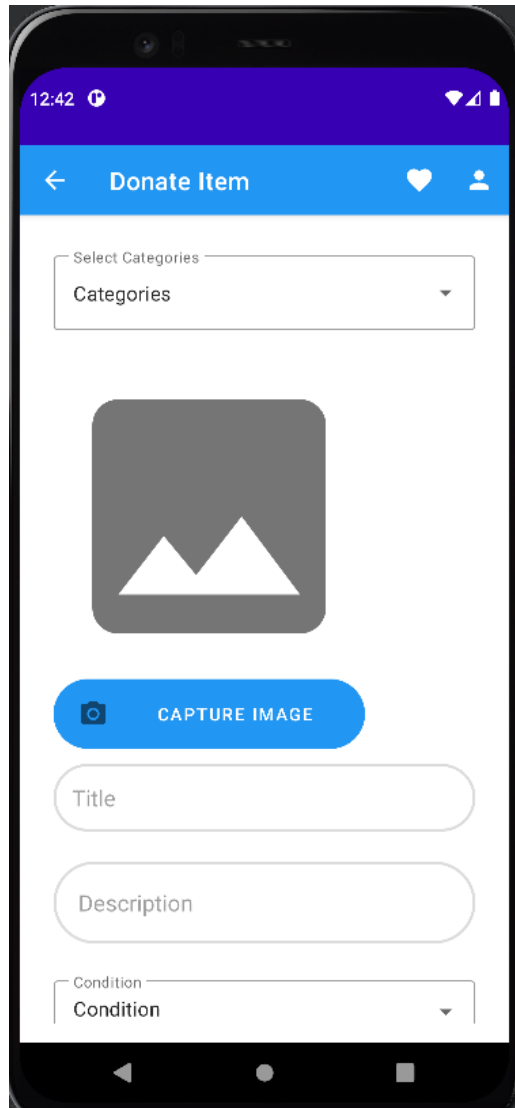
Appendix - A 15: Add New User

When admin click the add new user button from Appendix A-14: Admin Page, it will redirect to this interface. Admin required to enter the user information and click the Add new user button to add a new user to the database.



Appendix - A 16: Edit User

When admin click the edit button from Appendix A-14: Admin Page, it will redirect to this interface. The admin can click the edit button to update the new information of the user.



Appendix - A 17: Donate Page

Appendix A-17: Donate Page which able seller to donate their item to someone else that needed the item. Sellers are required to enter the required data of the item except the price of the item.

3.2 HARDWARE AND SOFTWARE SPECIFICATION

Table 3.1 Software and Hardware Specification

Name	Version	Type	Description	Purpose of Use
Android Studio	4.2.2	Software	In the Java and Kotlin languages, an integrated development environment for Android application development is offered.	Develop the whole interface and function of the system.
Android Emulator	Oreo	Software	In the Android Studio, a system that operates like an Android application.	To run and test the system virtually during implementation.
HP Pavilion Laptop 15-cs2xxx	Intel Core i5-8265U	Hardware	A laptop is a compact computer that may be carried around with you. They are intended to be more portable than standard desktop computers while yet providing many of the same functions.	Run all the software required to build this application.

Microsoft Office	2016	Software	Microsoft Office is a collection of client applications, server software, and services.	Prepare all the documentation and design.
------------------	------	----------	---	---

APPENDIX B
SDD

2022

SOFTWARE DESIGN DESCRIPTION (SDD)

[UMP Item Trading Application]



DOCUMENT APPROVAL

	Name	Date
Authenticated by: _____ Name		
Approved by:		

<hr/> <p>Client</p>		
---------------------	--	--

Software :

Archiving Place :

TABLE OF CONTENT

CONTENT	PAGE
DOCUMENT APPROVAL	II
TABLE OF CONTENT	IV
LIST OF FIGURES	VII
LIST OF TABLES	VIII
LIST OF APPENDICES	XI
1	CHAPTER 1 1
1.1 PROJECT DESCRIPTION	1
1.2 SYSTEM IDENTIFICATION	2

1.3	ARCHITECTURE / BLUE PRINT	2
1.3.1	General Architecture	2
1.3.2	Package Module	3
1.3.3	Business Service Layer	8
1.4	ARCHITECTURE / BLUEPRINT DESCRIPTION	10
1.4.1	Manage Registration	10
1.4.2	Manage Profile	12
1.4.3	Manage Goods	12
1.4.4	Manage Order	14
1.4.5	Manage User	15
2		CHAPTER 2 16
2.1	DETAILED DESCRIPTION	16
2.1.1	Manage Registration	16
2.1.2	Manage Profile	29
2.1.3	Manage Goods	36
2.1.4	Manage Order	54
2.1.5	Manage User	66
2.2	DATA DICTIONARY	75
2.2.1	User	75
2.2.2	Item	75
2.2.3	Watchlist	77

LIST OF FIGURES

Figure 1.1 General Architecture	3
Figure 1.2 Package Module	3
Figure 1.3 Application Layer	4
Figure 1.4 Business Service Layer	8
Figure 1.5 Class Diagram	10
Figure 2.1 Manage Registration Module	16
Figure 2.2 Manage Profile Module	29
Figure 2.3 Manage Goods Module	36
Figure 2.4 Manage Order Module	54
Figure 2.5 Manage User Module	66

LIST OF TABLES

Table 1.1 Application Layer – Manage Registration	5
Table 1.2 Application Layer – Manage Profile	5
Table 1.3 Application Layer – Manage Goods	6
Table 1.4 Application Layer – Manage Order	6
Table 1.5 Application Layer – Manage User	7
Table 1.6 Business Service Layer – Controller	9
Table 1.7 Business Service Layer – Model	9
Table 1.8 Manage Registration Class	10
Table 1.9 Manage Profile Class	12
Table 1.10 Manage Trade Class	12
Table 1.11 Manage Order Class	14
Table 1.12 Manage User Class	15
Table 2.1 LoginPage	16
Table 2.2 registrationPage	17
Table 2.3 MainActivity	18
Table 2.4 ForgotPasswordPage	19
Table 2.5 Manage Registration Controller	20
Table 2.6 Manage Registration Model	28
Table 2.7 Profile Page	30
Table 2.8 Edit Profile Page	30

Table 2.9 Manage Profile Controller	31
Table 2.10 Manage Profile Model	35
Table 2.11 SellPage	36
Table 2.12 ItemPage	38
Table 2.13 EditItemPage	39
Table 2.14 SellOptionPage	40
Table 2.15 DonatePage	41
Table 2.16 Manage Goods Controller	42
Table 2.17 Manage Trade Model	52
Table 2.18 WatchListPage	54
Table 2.19 ItemDetailPage	55
Table 2.20 SearchPage	56
Table 2.21 Manage Order Controller	58
Table 2.22 Manage Order Model	65
Table 2.23 AdminPage	67
Table 2.24 EditUser	68
Table 2.25 AddNewUser	69
Table 2.26 Manage User Controller	70
Table 2.27 Manage User Model	74
Table 2.28 Data Dictionary-User	75
Table 2.29 Data Dictionary-Item	75
Table 2.30 Data Dictionary-Watchlish	77

LIST OF APPENDICES

1 CHAPTER 1

1.1 PROJECT DESCRIPTION

UMP Items Trading Application is an online mobile application that will be used by students of University Malaysia Pahang where it is specializing in purchases and selling items among UMP students. This application will be used by the sellers to sell their items on the platform and buyers able to buy the items uploaded by the sellers. This application can keep huge data organized and secured.

This application will be an online commerce system. The application will be designed to allow all the UMP students to use the system to manage and review the relevant information. The functionality requirement of the ITA is to manage registration, manage profiles, manage trades, and manage orders.

The first module is to manage registration. This module will be used by users to register an account for the application. After the users get the account from the application, users can log in to the system and get the application's main function like searching for items and selling items in the application.

The second module is to manage profiles. This module will be used by users to edit their profiles and update their information in the application database. Users can change their user name, and phone number in their profile.

The third module is to manage goods. This module will be used by sellers to manage the items that want to sell. Sellers can upload items, edit items and delete items of their items. Sellers uploading items are required to fulfill all the required information for the items to sell.

The fourth module is to manage orders. This module will be used by buyers to manage the item that they want to buy. Buyers can search for items and add the items to the watchlist. Buyers can make orders with the seller by communicating with the seller by email, SMS, or call, to make orders directly with the seller.

1.2 SYSTEM IDENTIFICATION

System Title: UMP Items Trading Application

System Abbreviation: ITA

System Identification Number: ITA v1 2022

1.3 ARCHITECTURE / BLUE PRINT

1.3.1 General Architecture

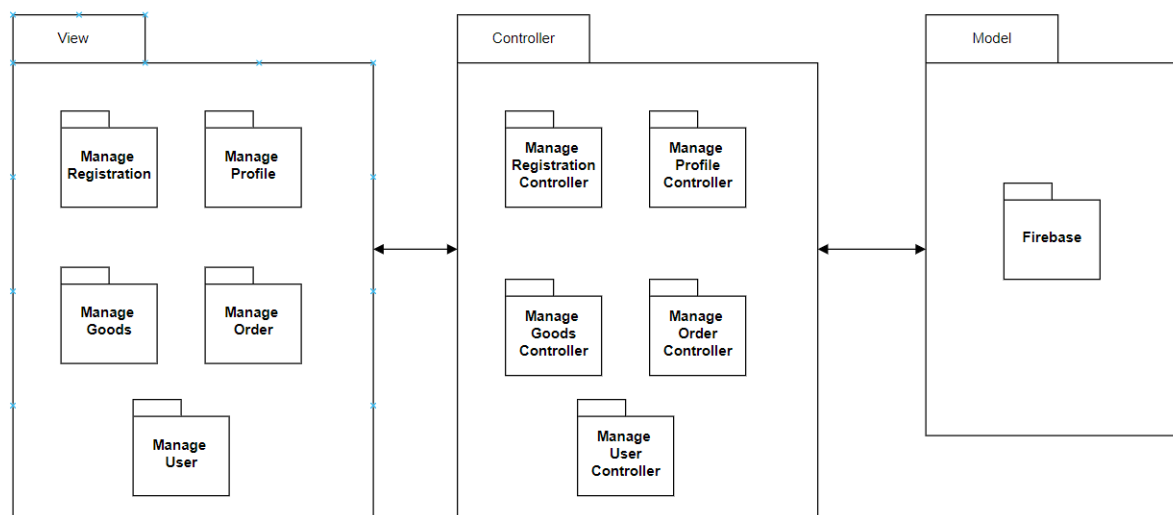


Figure 1.1 General Architecture

The system uses MVC architecture where it consists of three layers, M stands for Model, V stands for Views and C stands for controllers.

1.3.2 Package Module

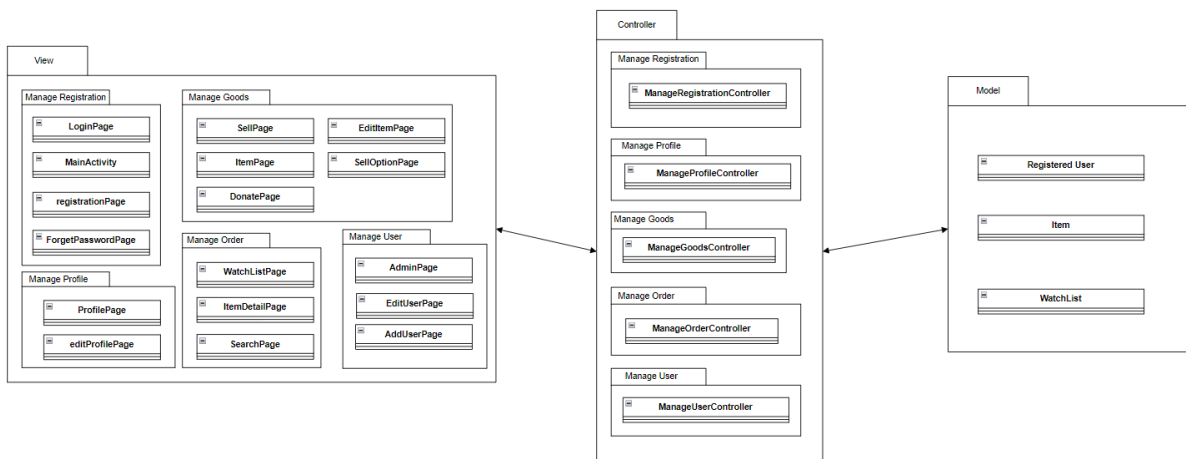


Figure 1.2 Package Module

1.3.2.1 Application Layer

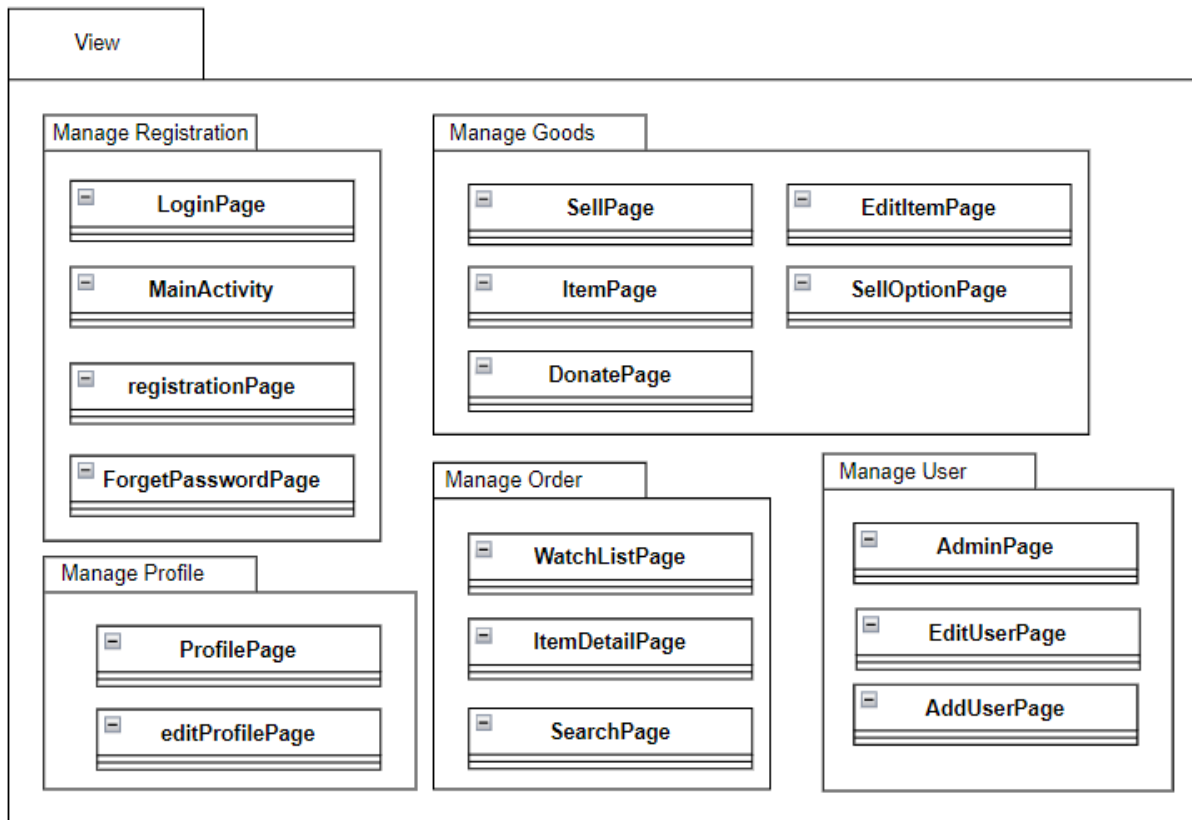


Figure 1.3 Application Layer

Figure 1.3 shows the interfaces categorized into four modules: manage registration, manage profile, manage trade, and manage order. The description of the interfaces are shows as tables below.

1.3.2.2 Manage Registration

Table 1.1 Application Layer – Manage Registration

Class Name	Description
MainActivity	This interface allows user to search item and sell item.
LoginPage	This interface allows the user to login.
registrationPage	This interface allows the user to register their account.
ForgetPasswordPage	This interface allows the user to reset their password.

1.3.2.3 Manage Profile

Table 1.2 Application Layer – Manage Profile

Class Name	Description
ProfilePage	This interface allows user to view user's profile detail.
editProfilePage	This interface allows user to edit user's profile information.

1.3.2.4 Manage Goods

Table 1.3 Application Layer – Manage Goods

Class Name	Description
SellPage	This interface allows user to upload all the information about the item that want to sell.
SellOptionPage	This interface allows users to select upload new item or modify existed item
EditItemPage	This interface allows user to edit the information of the item.
ItemPage	This interface allows user to view all the items that had been uploaded and be able to delete and edit them.
DonatePage	This interface allows user to donate their item to the application by entering the required information.

1.3.2.5 Manage Order

Table 1.4 Application Layer – Manage Order

Class Name	Description
SearchPage	This interface allows user to view all the item that fulfill the item name that the user searches for data and able to filter and sort item in this page.
ItemDetailPage	This interface allows the user to view the item detail information and contact the seller by selecting method at the bottom navigation bar.
WatchListPage	This interface allows the user to view all the item that had been added into the watchlist.

1.3.2.6 Manage User

Table 1.5 Application Layer – Manage User

Class Name	Description
AdminPage	This interface allows admin to view all the user information.
AddNewUser	This interface allows admin to add a new user to the system.

<p>EditUser</p>	<p>This interface allows admin to edit the existing user's status.</p>
-----------------	--

1.3.3 Business Service Layer

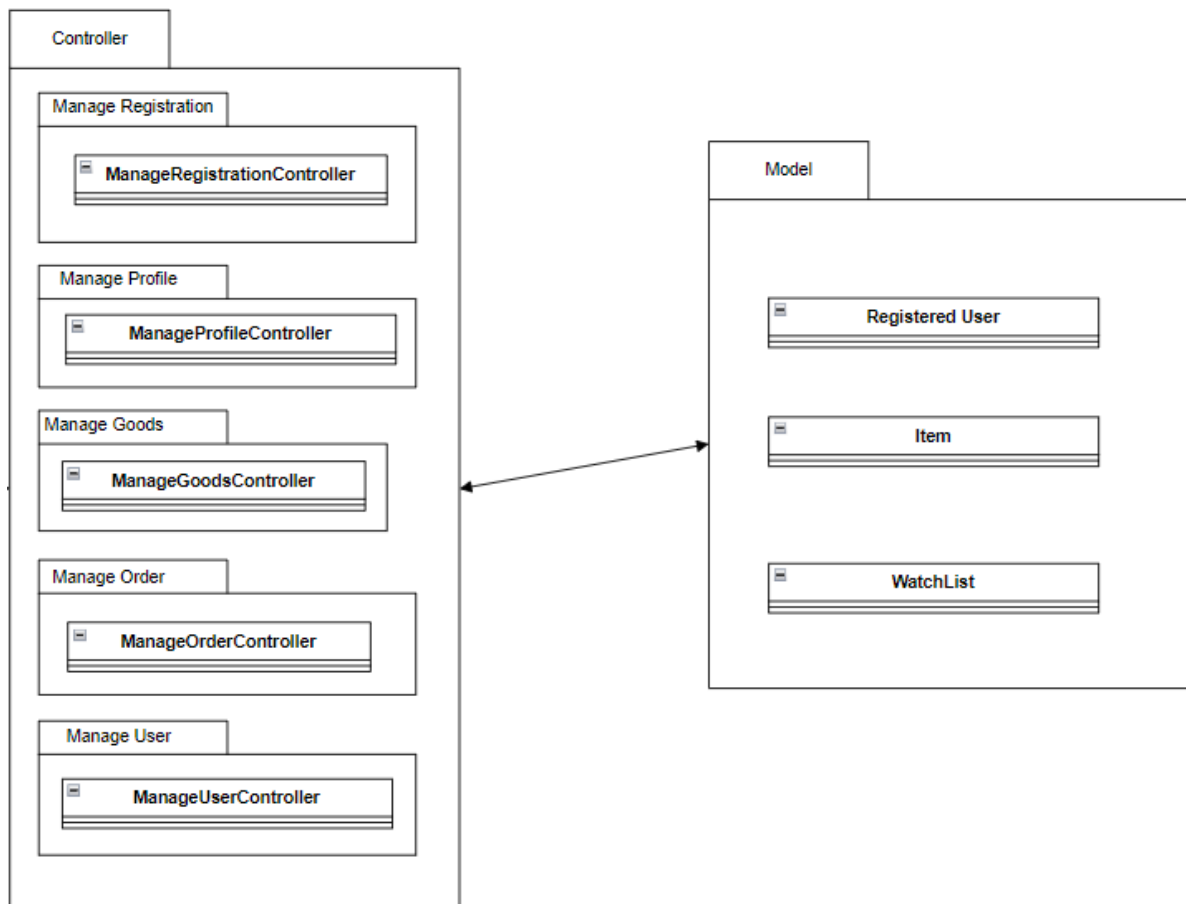


Figure 1.4 Business Service Layer

1.3.3.1 Controller

Table 1.6 Business Service Layer – Controller

Class Name	Description
Manage Registration Controller	To manage and process the registration data.
Manage Profile Controller	To manage and process the profile data.
Manage Goods Controller	To manage and process the item information that uploaded by seller.
Manage Order Controller	To manage and process the item information and seller contact information.
Manage User Controller	To manage and process the user information.

1.3.3.2 Model

Table 1.7 Business Service Layer – Model

Class Name	Description
Registered User	To store user data.
Item	To store item data.
WatchList	To store watch list data.

1.4 ARCHITECTURE / BLUEPRINT DESCRIPTION

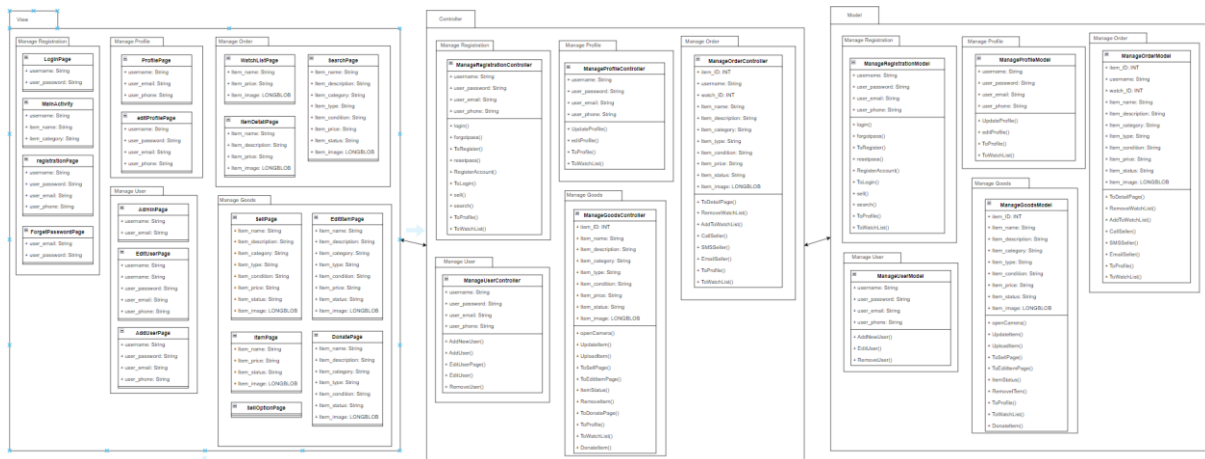


Figure 1.5 Class Diagram

Figure 1.5 shows the class diagram that consists four modules which are manage registration, manage profile, manage trade, and manage order.

1.4.1 Manage Registration

Table 1.8 Manage Registration Class

Manage Registration Controller	
login()	Allow the user to login to the application.
forgotpass()	Allow the user link to ForgotPasswordPage.
ToRegister()	Allow the user link to registrationPage.
resetpass()	Allow the user to reset password.
RegisterAccount()	Allow the user to register their account.
ToLogin()	Allow the user link to LoginPage.
sell()	Allow the user link to SellOptionPage.
search()	Allow the user to search item name and link to SearchPage.
ToProfile()	Allow the user to view their profile information
ToWatchList()	Allow the user to view the item that had been added into the watch list.

1.4.2 Manage Profile

Table 1.9 Manage Profile Class

Manage Profile Controller	
UpdateProfile()	Allow user to update the user's profile information
ToProfile()	Allow the user to view their profile information
ToWatchList()	Allow the user to view the item that had been added into the watch list.
editProfile()	Allow the user link to EditProfilePage.

1.4.3 Manage Goods

Table 1.10 Manage Trade Class

Manage Goods Controller	
openCamera()	Allow the user to open camera for take pictures.

UpdateItem()	Allow the user to update item information.
UploadItem()	Allow the user to upload item to the database.
ToSellPage()	Allow the user link to SellPage.
ToItemPage()	Allow the user link to ItemPage.
ToEditItemPage()	Allow the user link to EditItemPage.
ItemStatus()	Allow the user to modify the status of the item.
RemoveItem()	Allow the user to delete the item.
ToDonatePage()	Allow the user to link to Donate Page
DonateItem()	Allow the user to donate their item.
ToProfile()	Allow the user to view their profile information
ToWatchList()	Allow the user to view the item that had been added into the watch list.

1.4.4 Manage Order

Table 1.11 Manage Order Class

Manage Order Controller	
ToDetailPage()	Allow the user link to ItemDetailPage.
RemoveWatchList()	Allow the user to remove the item in watch list
AddToWatchList()	Allow the user to add the item to the watch list.
CallSeller()	Allow the user to use the call method to contact the seller.
SMSSeller()	Allow the user to use the SMS method to contact the seller.
EmailSeller()	Allow the user to use the email method to contact the seller.
ToProfile()	Allow the user to view their profile information
ToWatchList()	Allow the user to view the item that had been added into the watch list.

1.4.5 Manage User

Table 1.12 Manage User Class

Manage Trade Controller	
AddNewUser()	Allow the admin to add new user to the system
AddUser()	Allow the admin link to Add New User interface
EditUser()	Allow the admin to edit the existing user's information
ToEditUser()	Allow the admin link to Edit User interface
RemoveUser()	Allow the admin to remove the existing user.

2 CHAPTER 2

2.1 DETAILED DESCRIPTION

2.1.1 Manage Registration

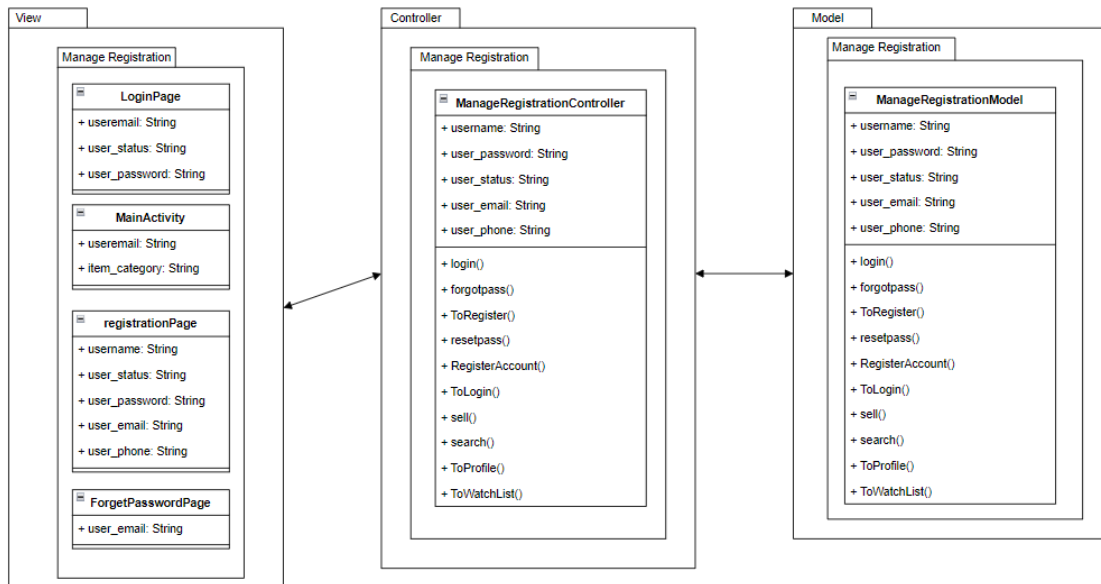


Figure 2.1 Manage Registration Module

2.1.1.1 LoginPage

Table 2.1 LoginPage

Class Type	Boundary Class
------------	----------------

Responsibility	This interface allows the user to login.	
Attributes	Attributes Name	Attributes Type
	user_email	String
	user_password	String
	User_status	String
Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.1.2 registrationPage

Table 2.2 registrationPage

Class Type	Boundary Class
Responsibility	This interface allows the user to register their account.

Attributes	Attributes Name	Attributes Type
	username	String
	user_password	String
	user_status	String
	user_email	String
	user_phone	String
Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.1.3 MainActivity

Table 2.3 MainActivity

Class Type	Boundary Class

Responsibility	This interface allows user to search item and sell item.	
Attributes	Attributes Name	Attributes Type
	useremail	String
	Item_category	String
Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.1.4 ForgetPasswordPage

Table 2.4 ForgetPasswordPage

Class Type	Boundary Class
Responsibility	This interface allows the user to reset their password.

Attributes	Attributes Name	Attributes Type
	user_email	String
Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.1.5 Manage Registration Controller

Table 2.5 Manage Registration Controller

Class Type	Controller Class	
Responsibility	To manage and process the registration data	
Attributes	Attributes Name	Attributes Type
	username	String

	user_password	String
	user_status	String
	user_email	String
	user_phone	String
Method	Method Name	Description
	login()	Allow the user to login to the application.
	forgotpass()	Allow the user link to ForgotPasswordPage.
	ToRegister()	Allow the user link to registrationPage.
	resetpass()	Allow the user to reset password.
	RegisterAccount()	Allow the user to register their account.

	ToLogin()	Allow the user link to LoginPage.
	sell()	Allow the user link to SellOptionPage.
	search()	Allow the user to search item name and link to SearchPage.
	ToProfile()	Allow the user to view their profile information
	ToWatchList()	Allow the user to view the item that had been added into the watch list.
Algorithms	<pre> login() BEGIN IF username and password correct THEN redirect to User or Admin Homepage ELSE display error message </pre>	

	<pre>END forgot_password() BEGIN IF Click forgot password Redirect to ForgetPasswordPage END RegisterAccount() BEGIN IF username missing input THEN display 'Name is required' error message Else IF email missing input</pre>
--	--

	<p>THEN display 'Email is required' error message</p> <p>Else IF email already exists</p> <p>THEN display the 'Email already in use' error message</p> <p>Else IF password missing input</p> <p>THEN display 'Password is required' error message</p> <p>Else IF phone number missing input</p> <p>THEN display 'Phone Number is required' error message</p> <p>Else send username, email and password to model.</p> <p>END</p> <p>ToRegister()</p> <p>BEGIN</p> <p>IF Click Sign Up Account</p>
--	--

	<p>Redirect to registrationPage</p> <p>END</p> <p>resetpass()</p> <p>BEGIN</p> <p>READ user_email</p> <p>READ new_user_pass</p> <p>IF click the “Reset Password” button</p> <p>UPDATE user_pass to new_user_pass</p> <p>Redirect to LoginPage</p> <p>END</p> <p>ToLogin()</p>
--	---

	<p>BEGIN</p> <p>IF aClick Already Have an Account</p> <p>Redirect to LoginPage</p> <p>END</p> <p>Sell()</p> <p>BEGIN</p> <p>IF click “Sell” button</p> <p>Redirect to SellOptionPage</p> <p>END</p> <p>Search()</p> <p>BEGIN</p>
--	--

	<p>READ item_name</p> <p>IF click “Search” button</p> <p>SELECT * from item where item_name LIKE %item_name%</p> <p>Passing data to SearchPage</p> <p>Redirect to SearchPage</p> <p>END</p> <p>ToProfile()</p> <p>BEGIN</p> <p>IF click the person icon on the action bar</p> <p>Redirect to ProfilePage</p> <p>END</p>
--	---

	<p>ToWatchList()</p> <p>BEGIN</p> <p>IF click the favorite icon on the action bar</p> <p>Redirect to WatchListPage</p> <p>END</p>
--	---

2.1.1.6 Manage Registration Model

Table 2.6 Manage Registration Model

Class Type	Entity class	
Responsibility	Model	
Attributes	Attributes Name	Attributes Type
	username	String

	User_status	String
	user_password	String
	user_email	String
	user_phone	String

2.1.2 Manage Profile

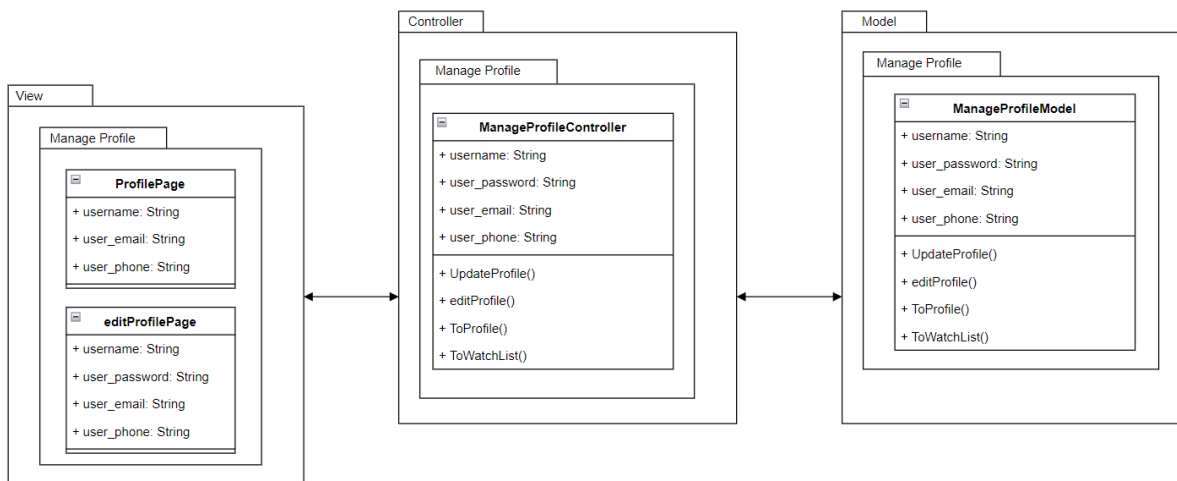


Figure 2.2 Manage Profile Module

2.1.2.1 ProfilePage

Table 2.7 Profile Page

Class Type	Boundary Class	
Responsibility	This interface allows the user to view the user's profile information.	
Attributes	Attributes Name	Attributes Type
	user_name	String
	user_phone	String
	user_email	String
Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.2.2 editProfilePage

Table 2.8 Edit Profile Page

Class Type	Boundary Class	
Responsibility	This interface allows the user to edit profile information.	
Attributes	Attributes Name	Attributes Type
	user_name	String
	user_password	String
	user_phone	String
	user_email	String
Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.2.3 Manage Profile Controller

Table 2.9 Manage Profile Controller

Class Type	Controller Class	
Responsibility	To manage and process the profile data	
Attributes	Attributes Name	Attributes Type
	username	String
	user_password	String
	user_email	String
	user_phone	String
Method	Method Name	Description
	UpdateProfile()	Allow user to update the user's profile information
	ToProfile()	Allow the user to view their profile information
	ToWatchList()	Allow the user to view the item that had been added into the watch list.

	editProfile()	Allow the user link to EditProfilePage.
Algorithms	<pre> UpdateProfile() BEGIN READ user_name READ user_email READ user_phone READ user_password IF Click "Update" button UPDATE user_name, user_email, user_phone and user_password Redirect to ProfilePage END </pre>	

	<pre>editProfile() BEGIN IF click "Edit Profile" Redirect to EditProfilePage END ToProfile() BEGIN IF click the person icon on the action bar Redirect to ProfilePage END ToWatchList()</pre>
--	---

	<p>BEGIN</p> <p>IF click the favorite icon on the action bar</p> <p>Redirect to WatchListPage</p> <p>END</p>
--	--

2.1.2.4 Manage Profile Model

Table 2.10 Manage Profile Model

Class Type	Entity class	
Responsibility	Model	
Attributes	Attributes Name	Attributes Type
	username	String
	user_password	String
	user_email	String

	user_phone	String
--	------------	--------

2.1.3 Manage Goods

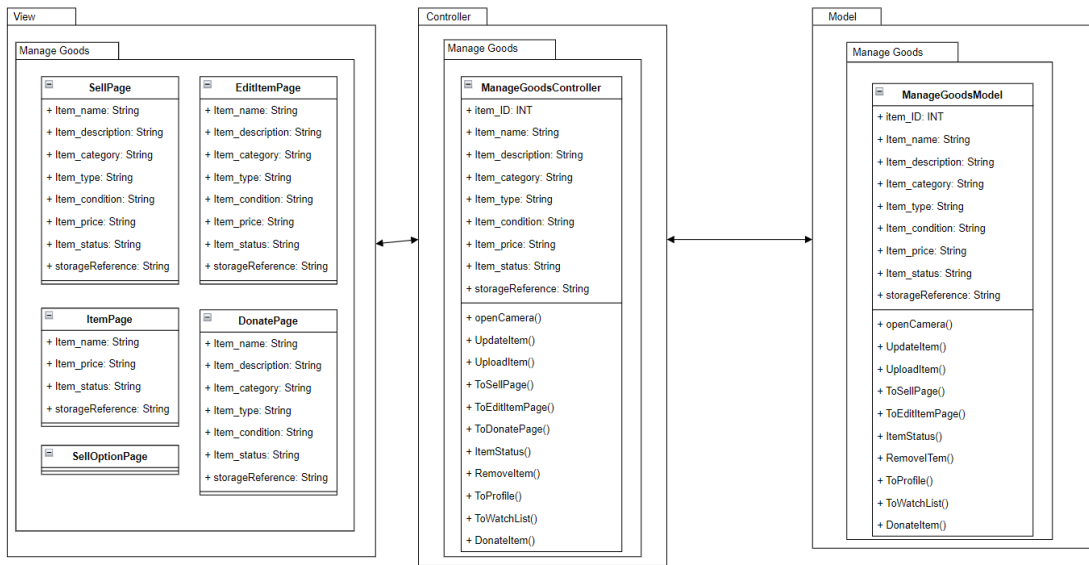


Figure 2.3 Manage Goods Module

2.1.3.1 SellPage

Table 2.11 SellPage

Class Type	Boundary Class
Responsibility	This interface allows the user to upload a new item.

Attributes	Attributes Name	Attributes Type
	Item_name	String
	Item_description	String
	Item_category	String
	Item_type	String
	Item_condition	String
	Item_price	String
	Item_status	String
	storageReference	String
Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.3.2 ItemPage

Table 2.12 ItemPage

Class Type	Boundary Class	
Responsibility	This interface allows the user to view the uploaded item.	
Attributes	Attributes Name	Attributes Type
	Item_name	String
	Item_price	String
	Item_status	String
	storageReference	String
Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.3.3 EditItemPage

Table 2.13 EditItemPage

Class Type	Boundary Class	
Responsibility	This interface allows the user to edit item information.	
Attributes	Attributes Name	Attributes Type
	Item_name	String
	Item_description	String
	Item_category	String
	Item_type	String
	Item_condition	String
	Item_price	String
	Item_status	String
	storageReference	String

Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.3.4 SellOptionPage

Table 2.14 SellOptionPage

Class Type	Boundary Class	
Responsibility	This interface allows the user to select a view item or upload a new item.	
Attributes	Attributes Name	Attributes Type
	N/A	N/A
Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.3.5 DonatePage

Table 2.15 DonatePage

Class Type	Boundary Class	
Responsibility	This interface allows the user to upload a new item.	
Attributes	Attributes Name	Attributes Type
	Item_name	String
	Item_description	String
	Item_category	String
	Item_type	String
	Item_condition	String
	Item_status	String
	storageReference	String
Method	Method Name	Description

	N/A	N/A
Algorithms	N/A	

2.1.3.6 Manage Goods Controller

Table 2.16 Manage Goods Controller

Class Type	Controller Class	
Responsibility	To manage and process the goods data	
Attributes	Attributes Name	Attributes Type
	Item_ID	String
	Item_name	String
	Item_description	String
	Item_category	String
	Item_type	String

	Item_condition	String
	Item_price	String
	Item_status	String
	storageReference	String
Method	Method Name	Description
	openCamera()	Allow the user to open camera for take pictures.
	UpdateItem()	Allow the user to update item information.
	UploadItem()	Allow the user to upload item to the database.
	ToSellPage()	Allow the user link to SellPage.
	ToItemPage()	Allow the user link to ItemPage.

	ToEditItemPage()	Allow the user link to EditItemPage.
	ItemStatus()	Allow the user to modify the status of the item.
	RemoveItem()	Allow the user to delete the item.
	ToDonatePage()	Allow the user to link to Donate Page
	DonateItem()	Allow the user to donate their item
	ToProfile()	Allow the user to view their profile information
	ToWatchList()	Allow the user to view the item that had been added into the watch list.
Algorithms	openCamera() BEGIN	

	<p>If Click the “Capture Picture” button</p> <p>Open the Camera and capture picture</p> <p>Passing the picture to item_image</p> <p>END</p> <p>UpdateItem()</p> <p>BEGIN</p> <p>READ item_name</p> <p>READ item_description</p> <p>READ item_category</p> <p>READ item_type</p> <p>READ item_condition</p> <p>READ item_price</p>
--	---

	<pre>READ item_status READ item_image If click "Update" UPDATE item_name, item_description, item_category, item_type, item_condition, item_price, item_status, item_image END UploadItem() BEGIN READ item_name READ item_description READ item_category READ item_type READ item_condition</pre>
--	--

	<pre>READ item_price READ item_status READ item_image If click "Upload" INSERT item_name, item_description, item_category, item_type, item_condition, item_price, item_status, item_image into item table END ToSellPage() BEGIN If click "Upload New Item" button Redirect to SellPage END</pre>
--	---

	<p>ToItemPage()</p> <p>BEGIN</p> <p>If click “Item Page” button</p> <p>Redirect to ItemPage</p> <p>END</p> <p>ToEditItemPage()</p> <p>BEGIN</p> <p>If click “Edit” button</p> <p>Redirect to EditItemPage</p> <p>END</p>
--	--

	<pre>ItemStatus() BEGIN READ item_ID If click "Sold Out" switch UPDATE item_status END RemoveItem() BEGIN READ item_ID If click "Delete" button DELETE item_ID END</pre>
--	---

	<pre>ToDonatePage() BEGIN If click "Donate Item" button Redirect to DonatePage END DonateItem() BEGIN READ item_name READ item_description READ item_category READ item_type</pre>
--	--

	<pre>READ item_condition READ item_status READ item_image If click "Upload" INSERT item_name, item_description, item_category, item_type, item_condition, item_status, item_image into item table END ToProfile() BEGIN IF click the person icon on the action bar Redirect to ProfilePage END</pre>
--	--

	<p>ToWatchList()</p> <p>BEGIN</p> <p>IF click the favorite icon on the action bar</p> <p>Redirect to WatchListPage</p> <p>END</p>
--	---

2.1.3.7 Manage Goods Model

Table 2.17 Manage Trade Model

Class Type	Entity class	
Responsibility	Model	
Attributes	Attributes Name	Attributes Type
	Item_ID	INT
	Item_name	String

	Item_description	String
	Item_category	String
	Item_type	String
	Item_condition	String
	Item_price	String
	Item_status	String
	Item_order	String
	storageReference	String

2.1.4 Manage Order



Figure 2.4 Manage Order Module

2.1.4.1 WatchListPage

Table 2.18 WatchListPage

Class Type	Boundary Class
Responsibility	This interface allows the user to view the item that added to watch list

Attributes	Attributes Name	Attributes Type
	Item_name	String
	Item_price	String
	Item_description	String
	storageReference	String
Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.4.2 ItemDetailPage

Table 2.19 ItemDetailPage

Class Type	Boundary Class
Responsibility	This interface allows the user to view the item detail

Attributes	Attributes Name	Attributes Type
	Item_name	String
	Item_price	String
	Item_condition	String
	storageReference	String
	Item_description	String
Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.4.3 SearchPage

Table 2.20 SearchPage

Class Type	Boundary Class
------------	----------------

Responsibility	This interface allows the user to search item by item name	
Attributes	Attributes Name	Attributes Type
	Item_name	String
	Item_price	String
	storageReference	String
	Item_description	String
	Item_category	String
	Item_type	String
	Item_status	String
	Item_order	String
	Item_condition	String
	Method	Method Name
N/A		N/A

Algorithms	N/A
------------	-----

2.1.4.4 Manage Order Controller

Table 2.21 Manage Order Controller

Class Type	Controller Class	
Responsibility	To manage and process the trade data	
Attributes	Attributes Name	Attributes Type
	Item_ID	String
	User_ID	String
	watchList_ID	String
	Item_name	String
	Item_description	String
	Item_category	String

	Item_type	String
	Item_condition	String
	Item_price	String
	Item_status	String
	storageReference	String
Method	Method Name	Description
	ToDetailPage()	Allow the user link to ItemDetailPage.
	RemoveWatchList()	Allow the user to remove the item in watch list
	AddToWatchList()	Allow the user to add the item to the watch list.
	CallSeller()	Allow the user to use the call method to contact the seller.

	SMSSeller()	Allow the user to use the SMS method to contact the seller.
	EmailSeller()	Allow the user to use the email method to contact the seller.
	ToProfile()	Allow the user to view their profile information
	ToWatchList()	Allow the user to view the item that had been added into the watch list.
Algorithms	<p>ToDetailPage()</p> <p>BEGIN</p> <p>READ item_ID</p> <p>If click the image</p> <p>Redirect to ItemDetailPage</p> <p>END</p>	

	<pre>RemoveWatchList() BEGIN READ watch_ID If click the "Delete" button DELETE * from watchlist where watch_ID = "watch_ID" END AddToWatchList(). BEGIN READ item_ID READ username If click the favorite icon</pre>
--	---

	<p>INSERT item_ID, username into watchlist</p> <p>Redirect to WatchListPage</p> <p>END</p> <p>CallSeller()</p> <p>BEGIN</p> <p>READ user_phone</p> <p>READ item_ID</p> <p>If click the “Call” method</p> <p>CALL user_phone</p> <p>END</p> <p>SMSSeller()</p>
--	---

	<pre>BEGIN READ user_phone READ item_ID If click the “SMS” method MESSAGE user_phone END EmailSeller() BEGIN READ user_email READ item_ID If click the “Email” method EMAIL user_email</pre>
--	---

	<p>END</p> <p>ToProfile()</p> <p>BEGIN</p> <p>IF click the person icon on the action bar</p> <p>Redirect to ProfilePage</p> <p>END</p> <p>ToWatchList()</p> <p>BEGIN</p> <p>IF click the favorite icon on the action bar</p> <p>Redirect to WatchListPage</p> <p>END</p>
--	--

2.1.4.5 Manage Order Model

Table 2.22 Manage Order Model

Class Type	Entity class	
Responsibility	Model	
Attributes	Attributes Name	Attributes Type
	Item_ID	String
	username	String
	watchList_ID	String
	Item_name	String
	Item_description	String
	Item_category	String
	Item_type	String
	Item_condition	String

	Item_order	String
	Item_price	String
	Item_status	String
	storageReference	String

2.1.5 Manage User

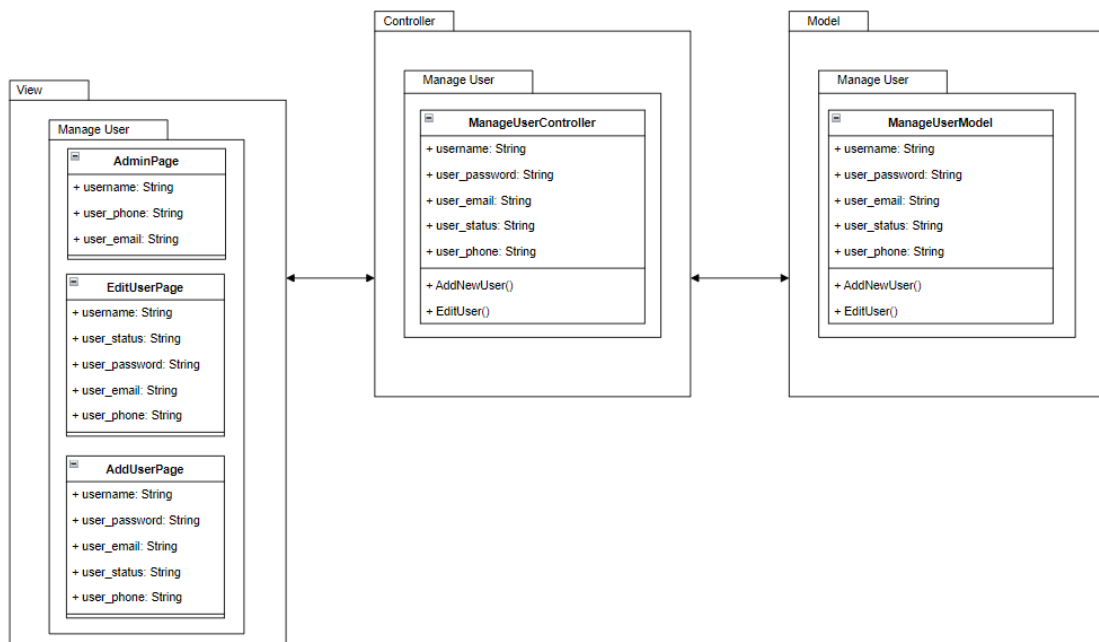


Figure 2.5 Manage User Module

2.1.5.1 AdminPage

Table 2.23 AdminPage

Class Type	Boundary Class	
Responsibility	This interface allows the admin to view the user's information	
Attributes	Attributes Name	Attributes Type
	user_name	String
	user_password	String
	user_status	String
	user_phone	String
	user_email	String
Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.5.2 EditUser

Table 2.24 EditUser

Class Type	Boundary Class	
Responsibility	This interface allows the admin to edit the user's information	
Attributes	Attributes Name	Attributes Type
	user_name	String
	user_password	String
	user_status	String
	user_phone	String
	user_email	String
Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.5.3 AddUserPage

Table 2.25 AddNewUser

Class Type	Boundary Class	
Responsibility	This interface allows the admin to add new user.	
Attributes	Attributes Name	Attributes Type
	user_name	String
	user_password	String
	user_phone	String
	user_status	String
	user_email	String
Method	Method Name	Description
	N/A	N/A
Algorithms	N/A	

2.1.5.4 Manage User Controller

Table 2.26 Manage User Controller

Class Type	Controller Class	
Responsibility	To manage and process the trade data	
Attributes	Attributes Name	Attributes Type
	user_name	String
	user_password	String
	user_phone	String
	user_status	String
	user_email	String
Method	Method Name	Description
	AddNewUser()	Allow the admin to add new user

	AddUser()	Allow the admin to link to Add new user page
	ToEditUser()	Allow the admin to link to Edit user page
	EditUser()	Allow the admin to edit the existing user's status
Algorithms	<p>AddNewUser()</p> <p>BEGIN</p> <p>READ username</p> <p>READ user_email</p> <p>READ user_password</p> <p>READ user_phone</p> <p>If click Add new user button</p>	

	<p>INSERT username, user_email, user_password, user_phone into user</p> <p>Redirect to AdminPage</p> <p>END</p> <p>AddUser()</p> <p>BEGIN</p> <p>If click the Add new User button</p> <p>Redirect to Addnewuser interface</p> <p>END</p> <p>ToEditUser()</p> <p>BEGIN</p> <p>If click the edit button</p>
--	---

	<p>Redirect to EditUserPage</p> <p>END</p> <p>EditUser()</p> <p>BEGIN</p> <p>READ user_ID</p> <p>READ username</p> <p>READ user_email</p> <p>READ user_status</p> <p>READ user_password</p> <p>READ user_phone</p> <p>If click Edit button</p> <p>UPDATE user_status from user WHERE user_id = entered “user_ID”</p>
--	--

	<p>Redirect to AdminPage</p> <p>END</p>
--	---

2.1.5.5 Manage User Model

Table 2.27 Manage User Model

Class Type	Entity class	
Responsibility	Model	
Attributes	Attributes Name	Attributes Type
	user_name	String
	user_password	String
	user_phone	String
	user_status	String
	user_email	String

2.2 DATA DICTIONARY

2.2.1 Registered user

Table 2.28 Data Dictionary-Registered User

Field Name	Description	Data Type	Constraint
User_ID	ID of user	VARCHAR(255)	PK
username	Name of User	VARCHAR(255)	
user_email	Email of user	VARCHAR(255)	
user_phone	Phone Number of user	VARCHAR(255)	
user_password	Password of user	VARCHAR(255)	

2.2.2 Item

Table 2.29 Data Dictionary-Item

Field Name	Description	Data Type	Constraint
Item_ID	ID of item	VARCHAR(255)	PK
User_ID	ID of user	VARCHAR(255)	FK
Item_name	Name of item	VARCHAR(255)	
Item_description	Description of item	VARCHAR(255)	
Item_category	Category of item	VARCHAR(255)	
Item_type	Type of item	VARCHAR(255)	
Item_condition	Condition of item	VARCHAR(255)	
Item_price	Price of item	VARCHAR(255)	
Item_status	Status of item	VARCHAR(255)	
storageReference	Reference of item image storage	VARCHAR(255)	
Item_order	Ordering of item	VARCHAR(255)	

2.2.3 Watchlist

Table 2.30 Data Dictionary-Watchlish

Field Name	Description	Data Type	Constraint
watchList_ID	ID of watch list	VARCHAR(255)	PK
User_ID	ID of User	VARCHAR(255)	FK
Item_ID	ID of item	VARCHAR(255)	FK

APPENDIX C
UAT TEST

UAT Form 1

No.	Module	Activities	Status		Comments
1.	Manage Registration	User login	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
2.		User register	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
3.		User reset password	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
4.	Manage Profile	User views the profile information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
5.		User edits the profile information.	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Can modify the username and phone number
6.	Manage Goods	User open Camera in application	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Need to select the picture from gallery.

7.		User selects category of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	the application has a lot of category for choose.
8.		User selects type of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
9.		User selects condition of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
10.		User adds item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
11.		User edits item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
12.		User views item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
13.		User edits the item status	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
14.		User removes item from database	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

15.		User donates the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
16.	Manage Order	User searches by the item title	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Can search by the one character.
17.		User views the searching result	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
18.		User sorts the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
19.		User filters the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Can filter by categories, types and conditions
20.		User selects method for contacting seller	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Can call, SMS, and email seller
21.		User views the item added into the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

22.		User removes the item from the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
23		User views the donated item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
24.	Manage User	Admin views the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
25.		Admin add the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Same as register
26.		Admin edit the user status	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Just can edit the user status.

This test has been performed by:

Name: LIM KAH CHUN

Signature: *lim kah chun*

Date: 10/1/2023

UAT Form use for testing.

UAT Form 2

No.	Module	Activities	Status		Comments
1.	Manage Registration	User login	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Can easily login
2.		User register	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
3.		User reset password	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
4.	Manage Profile	User views the profile information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
5.		User edits the profile information.	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
6.	Manage Goods	User open Camera in application	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
7.		User selects category of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
8.		User selects type of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

9.		User selects condition of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
10.		User adds item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
11.		User edits item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
12.		User views item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
13.		User edits the item status	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
14.		User removes item from database	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
15.		User donates the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
16.	Manage Order	User searches by the item title	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
17.		User views the searching result	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

18.		User sorts the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
19.		User filters the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
20.		User selects method for contacting seller	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Click the SMS will direct sending sms to seller
21.		User views the item added into the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
22.		User removes the item from the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
23		User views the donated item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
24.	Manage User	Admin views the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
25.		Admin add the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

26.		Admin edit the user status	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
-----	--	----------------------------	---	-----------------------------	--

This test has been performed by:

Name: LOH KIN MING

Signature: *Loh*

Date: 11/1/2023

UAT Form use for testing.

UAT Form 3

No.	Module	Activities	Status		Comments
1.	Manage Registration	User login	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
2.		User register	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
3.		User reset password	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Reset password need to get the email to reset password
4.	Manage Profile	User views the profile information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
5.		User edits the profile information.	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Can not edit the email
6.	Manage Goods	User open Camera in application	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

7.		User selects category of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
8.		User selects type of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
9.		User selects condition of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
10.		User adds item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
11.		User edits item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
12.		User views item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
13.		User edits the item status	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
14.		User removes item from database	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
15.		User donates the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

16.	Manage Order	User searches by the item title	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
17.		User views the searching result	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
18.		User sorts the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
19.		User filters the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
20.		User selects method for contacting seller	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
21.		User views the item added into the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
22.		User removes the item from the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
23.		User views the donated item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
24.	Manage User	Admin views the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

25.		Admin add the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
26.		Admin edit the user status	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

This test has been performed by:

Name: FOONG KIN HONG

Signature: *Foong*

Date: 11/1/2023

UAT Form use for testing.

UAT Form 4

No.	Module	Activities	Status		Comments
1.	Manage Registration	User login	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
2.		User register	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Very good cause of the registered email can not register again
3.		User reset password	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
4.	Manage Profile	User views the profile information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
5.		User edits the profile information.	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
6.	Manage Goods	User open Camera in application	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Require permission

					for using camera
7.		User selects category of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Got a lot of categories
8.		User selects type of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Also can sort the item
9.		User selects condition of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
10.		User adds item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
11.		User edits item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
12.		User views item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Can view own item in item page

13.		User edits the item status	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
14.		User removes item from database	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
15.		User donates the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
16.	Manage Order	User searches by the item title	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
17.		User views the searching result	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Just few item in the system
18.		User sorts the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
19.		User filters the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
20.		User selects method for contacting seller	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
21.		User views the item added into the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

22.		User removes the item from the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
23		User views the donated item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
24.	Manage User	Admin views the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
25.		Admin add the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
26.		Admin edit the user status	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

This test has been performed by:

Name: TEO VOON CHUAN

Signature: *Teo*

Date: 11/1/2023

UAT Form use for testing.

UAT Form 5

No.	Module	Activities	Status		Comments
1.	Manage Registration	User login	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
2.		User register	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
3.		User reset password	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
4.	Manage Profile	User views the profile information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
5.		User edits the profile information.	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
6.	Manage Goods	User open Camera in application	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
7.		User selects category of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
8.		User selects type of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

9.		User selects condition of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
10.		User adds item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
11.		User edits item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
12.		User views item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
13.		User edits the item status	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
14.		User removes item from database	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
15.		User donates the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
16.	Manage Order	User searches by the item title	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
17.		User views the searching result	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

18.		User sorts the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
19.		User filters the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
20.		User selects method for contacting seller	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
21.		User views the item added into the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
22.		User removes the item from the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
23		User views the donated item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
24.	Manage User	Admin views the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
25.		Admin add the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
26.		Admin edit the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

This test has been performed by:

Name: KOH QING ZHE

Signature: 

Date: 12/1/2023

UAT Form use for testing.

UAT Form 6

No.	Module	Activities	Status		Comments
1.	Manage Registration	User login	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
2.		User register	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
3.		User reset password	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Had to use the link to reset the password
4.	Manage Profile	User views the profile information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
5.		User edits the profile information.	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Cannot edit the password and email
6.	Manage Goods	User open Camera in application	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	


7.		User selects category of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
8.		User selects type of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
9.		User selects condition of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
10.		User adds item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
11.		User edits item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
12.		User views item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
13.		User edits the item status	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
14.		User removes item from database	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
15.		User donates the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

16.	Manage Order	User searches by the item title	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Can easily search by one character
17.		User views the searching result	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Will display all the searching result
18.		User sorts the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
19.		User filters the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
20.		User selects method for contacting seller	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
21.		User views the item added into the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
22.		User removes the item from the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

23		User views the donated item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
24.	Manage User	Admin views the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
25.		Admin add the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
26.		Admin edit the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

This test has been performed by:

Name: KONG KEI

Signature: 

Date: 12/1/2023

UAT Form use for testing.

UAT Form 7

No.	Module	Activities	Status		Comments
1.	Manage Registration	User login	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Must login with the email
2.		User register	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
3.		User reset password	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
4.	Manage Profile	User views the profile information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Profile did not have the profile picture
5.		User edits the profile information.	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
6.	Manage Goods	User open Camera in application	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
7.		User selects category of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

8.		User selects type of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
9.		User selects condition of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
10.		User adds item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
11.		User edits item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Can edit item easily
12.		User views item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
13.		User edits the item status	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
14.		User removes item from database	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Will directly delete the item without confirmation
15.		User donates the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

16.	Manage Order	User searches by the item title	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
17.		User views the searching result	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
18.		User sorts the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Can sort the item by prices
19.		User filters the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
20.		User selects method for contacting seller	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
21.		User views the item added into the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
22.		User removes the item from the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
23		User views the donated item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

24.	Manage User	Admin views the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
25.		Admin add the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
26.		Admin edit the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

This test has been performed by:

Name: HO SHU XIAN

Signature: 

Date: 12/1/2023

UAT Form use for testing.

UAT Form 8

No.	Module	Activities	Status		Comments
1.	Manage Registration	User login	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
2.		User register	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
3.		User reset password	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
4.	Manage Profile	User views the profile information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
5.		User edits the profile information.	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
6.	Manage Goods	User open Camera in application	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Able to open camera
7.		User selects category of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Got a lot of category to be selected

8.		User selects type of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
9.		User selects condition of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	Only two condition
10.		User adds item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
11.		User edits item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
12.		User views item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
13.		User edits the item status	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
14.		User removes item from database	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
15.		User donates the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
16.	Manage Order	User searches by the item title	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

17.		User views the searching result	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
18.		User sorts the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
19.		User filters the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
20.		User selects method for contacting seller	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
21.		User views the item added into the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
22.		User removes the item from the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
23		User views the donated item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
24.	Manage User	Admin views the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	User cannot be search
25.		Admin add the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

26.		Admin edit the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
-----	--	---------------------------------	---	-----------------------------	--

This test has been performed by:

Name: LING YEE BING

Signature: *ling*

Date: 12/1/2023

UAT Form use for testing.

UAT Form 9

No.	Module	Activities	Status		Comments
1.	Manage Registration	User login	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
2.		User register	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
3.		User reset password	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
4.	Manage Profile	User views the profile information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
5.		User edits the profile information.	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
6.	Manage Goods	User open Camera in application	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
7.		User selects category of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
8.		User selects type of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

9.		User selects condition of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
10.		User adds item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
11.		User edits item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
12.		User views item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
13.		User edits the item status	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
14.		User removes item from database	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
15.		User donates the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
16.	Manage Order	User searches by the item title	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
17.		User views the searching result	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

18.		User sorts the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
19.		User filters the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
20.		User selects method for contacting seller	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
21.		User views the item added into the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
22.		User removes the item from the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
23		User views the donated item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
24.	Manage User	Admin views the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
25.		Admin add the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
26.		Admin edit the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

This test has been performed by:

Name: HO XUE YING

Signature: *Ho*

Date: 12/1/2023

UAT Form use for testing.

UAT Form 10

No.	Module	Activities	Status		Comments
1.	Manage Registration	User login	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
2.		User register	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
3.		User reset password	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
4.	Manage Profile	User views the profile information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
5.		User edits the profile information.	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
6.	Manage Goods	User open Camera in application	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
7.		User selects category of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
8.		User selects type of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

9.		User selects condition of item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
10.		User adds item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
11.		User edits item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
12.		User views item information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
13.		User edits the item status	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
14.		User removes item from database	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
15.		User donates the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
16.	Manage Order	User searches by the item title	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
17.		User views the searching result	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

18.		User sorts the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
19.		User filters the item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
20.		User selects method for contacting seller	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
21.		User views the item added into the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
22.		User removes the item from the watchlist	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
23		User views the donated item	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
24.	Manage User	Admin views the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
25.		Admin add the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	
26.		Admin edit the user information	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	

This test has been performed by:

Name: KOH WEI QI

Signature: *vicky*

Date: 12/1/2023

UAT Form use for testing.