FREELANCER MOBILE APPLICATION

HO HOCK JIHN

SESSION 2019/2020

FACULTY OF INFORMATION SCIENCE & TECHNOLOGY

MULTIMEDIA UNIVERSITY

MARCH 2020

FREELANCER MOBILE APPLICATION

BY

HO HOCK JIHN

SESSION 2019/2020

THE PROJECT REPORT IS PREPARED FOR

FACULTY OF INFORMATION SCIENCE & TECHNOLOGY
MULTIMEDIA UNIVERSITY
IN PARTIAL FULFILLMENT
FOR

BACHELOR OF INFORMATION TECHNOLOGY
B.I.T. (HONS) SECURITY TECHNOLOGY

FACULTY OF INFORMATION SCIENCE & TECHNOLOGY

MULTIMEDIA UNIVERSITY

MARCH 2020

DECLARATION

I hereby declare that the work have been done by myself and no portion of the work contained in this thesis has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

_____

HO HOCK JIHN

Faculty of Information Science & Technology
Multimedia University

Date:

ACKNOWLEDGEMENT

For this project, I would like to express my thanks and gratitude to the various individual and institutions that have contributed to the successful completion of this project. Without them and their support and help, this project would not be complete. Appreciation and all the thanks go to my supervisor, Ms. Chong Lee Ying for her endless guidance and enormous patience throughout the development of the research.

It is an honour of having the chance to conduct this project whereby this is the occasion for me to gain extra knowledge of the project I am conducting. I'd appreciate that Ms. Chong Lee Ying gave me this open door by willing to have me as one of her students for this Final Year Project.

My appreciation goes to the Multimedia University for all the facilities provided and co-operation in allowing me to use the laboratory and the research centre so that I am able to produce a positive outcome in the project. Lastly, not forgetting my loving parent and fellow course mates who provide moral support, encouragement and attention throughout this project. Thank you.

ABSTRACT

In this modern day, technology has been invaded into our daily life in all ways. The access to education, medicine, industry, transportation and so on has been simplified due to modern day technology. Technology influences people throughout the world, both positively and negatively. Technology that influences people the most is smartphones. Since Apple introduced their first smartphone, which is the iPhone, the smartphone industries are getting higher demand and higher supply since it is easier more function to use. Since smartphones are getting more and more variance, it is also cheap, affordable and sophisticated smartphones. The reason smartphone is widely use by people in the world is because it has many functionalities such as make and receive phone calls, browsing internet, listening to music and etc.

When people are free, they might use their phone to entertain themselves by browsing Facebook or watching YouTube. What if they can utilize their free time by doing some project that can improve their skills and earn some side income that can found on the smartphone. By doing so, they can have a better resume when looking for job, or they can make this as their main income and have a flexible working hour. They can be their own boss to work whenever and wherever they want. Apart from that, student of university also can earn some extra money during their free time. Since they are quite free after their class, they can use their skill that learned from their class to help other people complete their task. Meanwhile the money they earn can pay their university's tuition fees to lighten their parents' burden.

This project aims to build a mobile application which allow them to promote themselves and find project to work on. This mobile application allows people to work flexibly and earn some side income using the most user-friendly and simplest way for them. Before developing this application, some of the similar existing applications are reviewed and studied. This project is developed using Android mobile development tools in light of the fact that the number of smartphone users are increasing since the past few years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS/ SYMBOLS

DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
FIST	Faculty of Information Science and Technology
FYP	Final Year Project
GIMP	GNU Image Manipulation Program
MMU	Multimedia University
OS	Operating System
SDLC	System Development Life Cycle
SMS	Short Message Service

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

With the evolution of technology, the society has been dramatically changed. One of the most popular technology that change the way society lives is smartphone. The technology of smartphones is enhancing and most probably it will replace laptop in the future. Almost every person in the world owns a smartphone. People use smartphone to manage their activity in their daily life. People are becoming more and more dependent on smartphone.

Smartphone is an essential item to be owned in the 21st century. Without smartphone, the connectivity between people become less. People unable to connect through Facebook or Instagram anywhere and anytime. This causes the communication between people become inconvenience. With smartphones, it allows people to access the internet whenever and wherever they are. It also allows people to conduct transaction on the internet to buy or sell whatever products or services they desire. Besides, there are people with various types of computing skills such as graphic design, programming etc. to promote themselves on the internet to earn side income. Also, it allows people to find expert in computing skills to help them to complete their task. However, without a proper application, none of this connectivity and messaging can be worked smoothly. This is why mobile application is very important in a smartphone. Other than this, the feature in the smartphone such as alarm clock, camera, calculator and others improve the life quality of the society.

With the evolution of smartphones and the Internet, it creates an opportunity to allow this project to be implemented to help people around the world. This project is to develop a mobile application for freelancer to sell their skill. This application is a user-friendly application as well as easy to learn. Moreover, buyer can use this application to look for freelancer that suit them to help them with their project by paying fees that set by the freelancer.

1.2 Problem Statement

In this era, when people want to find someone who have skills, they have to browse either through social media or through flyer. Although they can certainly find the expert they are looking for at some point, but it consumes a lot of time to find in the social media since social media are generally used for entertainment. Besides, when they found suitable freelancer on social media, they might end up being swindle by the freelancer since everyone can easily create a fake account and use a fake identity or reputable freelancer's identity. Other than that, freelancer might not receive payment when they complete client's project and this waste their time. That is why social media is not a suitable place to perform a business transaction as social media does not have a safe transaction procedure or function.

On the other hand, before the rise of smartphone, people communicate via Short Message Service (SMS) or through e-mail. Both of these communication mediums are not practical. The reason behind this is because SMS in the past is costly. Therefore, if people found the freelance service through flyer, they would most probably use SMS to communicate to discuss how they want they service is done. By the time they finish discussing, they spend maybe quarter of their budget on the SMS cost. Besides, using e-mail to communicate is not and ideal option because both parties does not know when the other party is available to discuss. Communicating via e-mail consume plenty of time.

1.3 Project Objectives

This application should be an effective tool for user, which able to provide the following objectives:

- **To design a mobile application that enable freelancer to sell their skill by providing double layers protection on the transaction between the freelancer and the buyer.**

  Developing this mobile application enable freelancers to have a platform to promote themselves and bring convenience to the buyer, as they know where to find the expert with specific skill to solve their problem. The way this application provide double layers protection is the money is transferred to the server platform when buyer paid for the order they have made. After the order is paid, the seller is notified on the order and is required to complete the order within the completion time. After seller has completed the order, buyer verify the order by pressing the "RECEIVED" button. After the button is pressed, the server only transfers the total fund deducted the transaction fees of 6% to the seller's e-wallet.

- **To create a digital platform that enables the communication between freelancer and buyer before a service is delivered.**

  This mobile application allows both parties (freelancer and buyer) to communicate with each other in order to get to know the details of the project in order to maximize the satisfactory of the client and quality of the project.

- **Prepare an admin to monitor the identity of seller to prevent fraud.**

  In order to sell service in this mobile application, seller has to verify their identity by providing image of their Identification Card and their Identification Card number. After submitting their image and number, admin verify their identity by matching the submitted image and number. Admin can accept the application or reject the application. After accepting, an e-mail is sent to their email to notify them that they can start adding service to this mobile application. If admin reject their application, an e-mail is sent to the user's email to notify them that their application is rejected.

1.4 Project Scope

The purpose of developing this application is to help ease the buyer and seller (or freelancers). In the past, freelancers do not have a platform to promote themselves and find work. They have to promote themselves through social media or by distributing flyer. By developing this application, it creates an opportunity to help freelancer earn side income by allowing them to promote themselves based on their skills, which is sorted by category. The higher the rating of their profile, the higher their chance to show up in search result. Besides, this application allows both parties to perform a safer transaction to avoid defrauding from either side. It helps freelancers to avoid wasting their time working on a project that eventually not be paid and avoid buyers paid for nothing.

Besides, buyer can grab this opportunity to utilize this application by searching for suitable freelancer to help them complete their project. Before this proposed application is developed, buyer search the expert through social media. By using this application, they are able to find the expert they need through the category sorting in the proposed application. This help to save their time and their problem can be solved as soon as possible. Besides, they can put their mind at ease about swindling when performing transaction through this application. When buyer transfer the total fund, it does not directly debit to the seller account. However, after the seller has complete the project, and buyer can confirm it by clicking the “Received” button, only then a transaction fee of six percent (6%) is deducted from the fund and is debit to the seller’s e-wallet.

1.5 Report Organisation

This report consists of five chapter as shown below:

- Chapter 1: Introduction
- Chapter 2: Literature Review
- Chapter 3: Methodology
- Chapter 4: Proposed Solution and Implementation Plan
- Chapter 5: The Solution
- Chapter 6: The Implementation Process / Result
- Chapter 7: Testing / Evaluation of Findings
- Chapter 8: Conclusion

Chapter 1 consist of the concept and idea of the project. In this chapter, it consists of Project Overview, Problem Statement, Project Objective, and Project Scope.

Chapter 2 contain the comparison of five existing application which have the similar function that available in the Google PlayStore. The listing of strengths and weaknesses of each application helps understand and improve the idea to develop this project. Besides, this chapter also contain screenshots of the functions in the application and explanations based on the screenshot. Furthermore, comparison table to highlight the different features in those existing application also include in this chapter.

Chapter 3 consists of the methodology used for this project and is studied further for improvement of managing this project, whereby the models in System Development Life Cycle (SDLC) is studied and choose the Waterfall Model to carry out this project. Additionally, this chapter also provides information of all the software tools that required in this project, along with the programming language needed to execute the application.

In Chapter 4, the illustration of system design is illustrated by using Unified Modelling Language (UML) diagrams which consists of Context Diagram, Data Flow Diagrams, Use Case Diagram, Flowchart Diagram and System Organization Chart. Besides that, this chapter also consists of Database Design which include Business Rule, Entity Relationship Diagram, List of Tables, Table Design, and Data Dictionary. Apart from that, this chapter also consist of the prototype of the application interface design.

Chapter 5 consists of the preparation and setup of Firebase and Android Studio. Besides, this chapter also consists of print screen of actual system interface of this project alongside with the brief description of each print screen. Apart from that, this chapter also contain the summary and overview diagram of the solution.

Chapter 6 consists of the comparison between original goal and current system. This comparison is to show that the some of the original goal is excluded and new feature included from this project to enhance the functionality of this mobile application. Besides, this chapter also contain new features added into this project with brief description to explain the features. Moreover, extra features that are not mention before also included in this chapter.

In Chapter 7, the testing outline is conducted to ensure the mobile application is working perfectly as intended and operate smoothly for every user. Besides, this chapter also contain the print screen of errors while developing this mobile application alongside with the solution to solve the errors. In Chapter 8, which is the last chapter of this report, it is the conclusion of the whole report.

CHAPTER 2

LITERATURE REVIEW

2.1 History of Freelancer

The word freelance is very popular in the information age. When people hear the word ‘freelancing’, what appears into their mind is internet, computer or smartphone. This is quite natural as these are basically the tools of the 21st century freelancer. Some people may think that this word is fairly new. However, the word ‘freelance’ is quite an old word or to say even older than the word ‘internet’.

To start with historical background, the word “free” comes from Germanic origin and means “to love” and the word “lance” comes from French origin and means to throw, discharge or hurl with force (Andor, 2018). When the two words combined, the word “to freelance” signifies a person discharges the force of his or her skills into the work they love. Besides, the word first came into English in the early 1800s, it was used to refer to a medieval mercenary who would fight for whichever nation or person paid them the most (History of ‘Freelance’, n.d.). The oldest written use of this word was found in Sir Walter Scott’s novel, ‘Ivanhoe’ as shown in Figure 2.1. There is a short historical background of what Walter Scott implied by utilizing that word. There were two kinds of soldiers in the middle Ages; some used to work for a king and others worked for whoever paid. Hence, free-lancers were the free mercenary soldiers who used to sell their skills (Laurinavicius, 2016). As years and centuries passed, the word overpassed the military field and found a prominent place in the business sector. However, there were no major changes in meaning — people still compete in the market, sell their work to various businesses, and in return, they get paid.

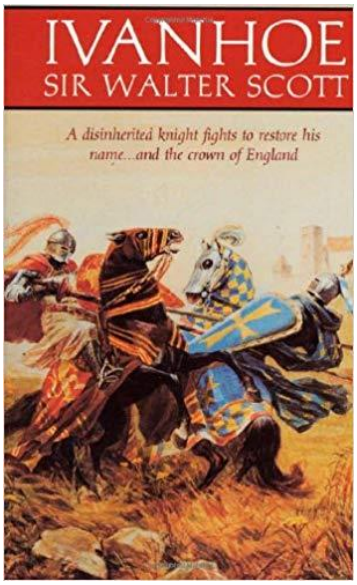


Figure 2.1: Book Cover of Ivanhoe

Before the internet is accessible throughout the world, freelancers had to promote themselves by distributing flyer or advertise on magazine. Although newspaper was considered the best medium that allow information to reach around the world, it is not an optimal option because it is quite costly to advertise daily. For that reason, they have to contact magazine company to pay for advertising or print out flyer of their information and distribute it to people or go to every house and slide in their mail box. Besides that, they had to ask their friends or family to promote them to other friends or relative to gain reputation. Either promoting method is by vocal or passing around their business card.

In the information age, in order for them to have a project to work on, they are required to find a project themselves on the internet. One of the ways to promote themselves on the internet is by creating a website. A website act as an online portfolio for freelancer to show their past work to potential client. Still they have to promote their website through friends and social media to build up reputation. Apart from creating website, they also able to promote themselves in social media such as Instagram, Facebook, and Twitter etc. Social media allow them to post picture of their works and show it to others. Once they have someone interested to have their project done from them, both parties can communicate through messaging feature integrated in the social media or by instant messaging applications such as WhatsApp or WeChat etc.

2.2 Comparison of Existing Applications

There are plenty of similar applications exists in Google Play Store and five applications have been selected for comparison purpose. It includes a brief introduction for each application in order to understand the background of these applications. These five applications are analysed based on functionalities, strengths, and weaknesses to provide a better understanding, knowledge, and skills to develop a better application. To conclude, there is comparison table to show the functionalities of each application along with their strengths and weaknesses.

2.2.1 Fiverr

Figure 2.2: Logo of Fiverr
(Adopted from Fiverr, 2010)

Fiverr was founded and launched in February 2010 by Micha Kaufman and Shai Wininger (Fiverr, 2010). The logo of Fiverr is as shown in Figure 2.2. The company is located in Tel Aviv, Israel and owned by Fiverr International Limited. The founders developed the idea of a marketplace that provide a two-sided platform for people to buy and sell a variety of digital services typically offered by freelancers. Digital services include graphic design, video editing, programming, writing and translation. Fiverr's services start as a minimum of USD$5 and can go up to thousands of dollars (Fiverr 2010).

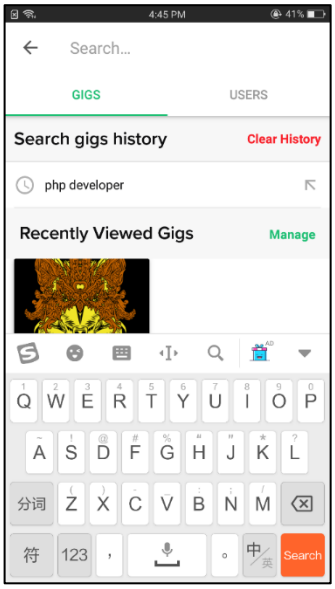


Figure 2.3: Search Function in Fiverr

As shown in Figure 2.3, this is the search function provided by the Fiverr to look for gigs, and users. It contains a search history and recently viewed gigs shown to users to bring convenience to users. Besides, user is able to clear the search history.

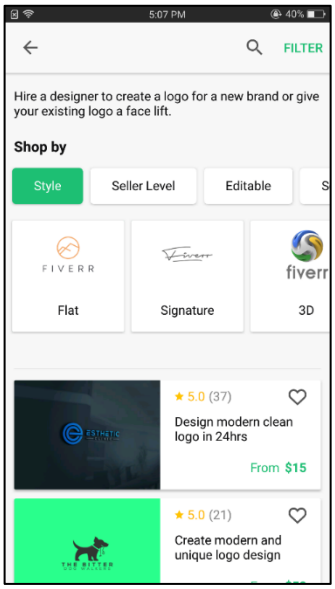


Figure 2.4: Browse Function in Fiverr

Figure 2.4 shows the browse function in Fiverr which allow user to browse the type of work. User can browse the term they searched and the result can be sorted by categories.

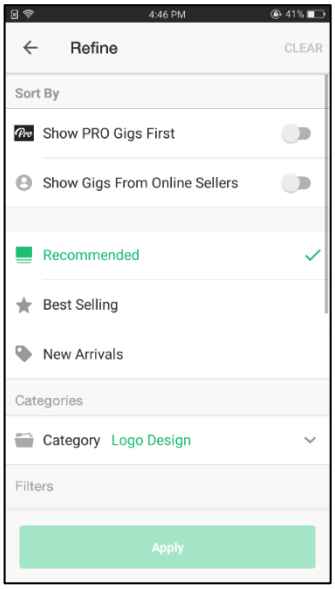


Figure 2.5: Filter Function in Fiverr

Figure 2.5 shows that Fiverr contain filter function to allow user to filter for price, categories etc. To use this function, user has to select which they want to see, after that click the apply button. Once clicked, it is sort according to what users has selected.

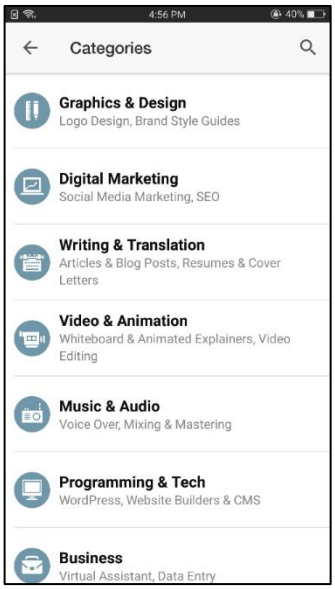


Figure 2.6: Categories in Fiverr

In Figure 2.6, user able to view type of categories in Fiverr, allowing them to choose which type of categories for them to view. The category has sub categories for user to choose specific category.

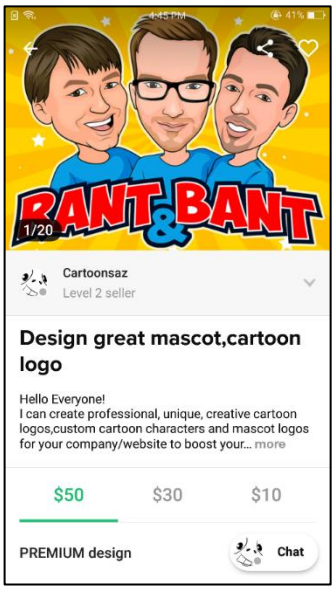


Figure 2.7: Information of Freelancer in Fiverr

Figure 2.7 shows the information of freelancer allow user to get to know the freelancer they wish to work with. Information such as price rate and reviews. This information allow user to decide whether they are suitable with their project.

Figure 2.8: Chat Function in Fiverr

As shown in Figure 2.8, Fiverr contain chat function to allow seller and buyer to chat. Besides chatting, this function also includes insert attachment function. It allows user to upload image or file and send to other party.

Strength:

- User friendly
- Allow browsing for categories and searching
- Contain e-wallet features
- Viewing profile of freelancer

Weakness:

- Automatically accepts jobs on the seller side.
- Fiverr charge 20% commission for seller.

2.2.2 Upwork

Figure 2.9: Logo of Upwork

(Adopted from Upwork, 2015)

Upwork previously known as Elance-oDesk is a global freelancing platform where businesses and independent professionals associate and work together remotely (Upwork, 2015). The company was once an independent company, which is Elance, founded in 1999 by Breerud Sheth, Srini Anumolu and Sanjay Noronha and oDesk, founded in 2003 by Odysseas Tsatalos and Stratis Karamanlakis. Both of the company announced their merger on December 2013 and formed Elance-oDesk. It was rebranded to Upwork on May 2015 (Profeldt, 2015). The logo of Upwork is as shown in Figure 2.9. It is based in Mountain View, California, U.S. (Upwork, 2015)

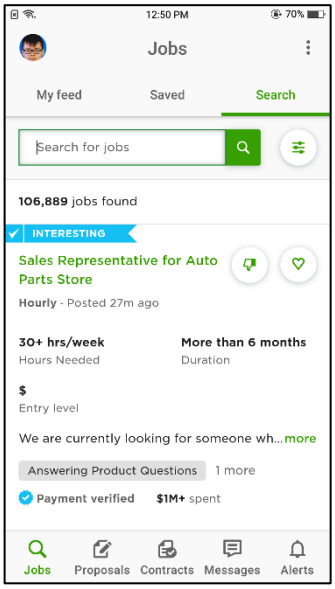


Figure 2.10: Search Function of Seller Side in Upwork

Figure 2.10 shows the search function on the seller side. This function allows the seller also known as freelancer to search for available project for them to work on. Seller can filter and sort the search result by their preference. Besides, seller also able save the job that interest them.

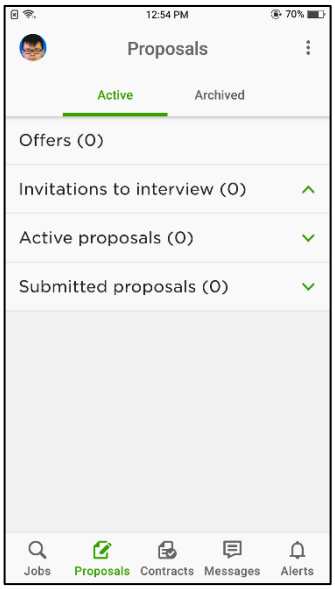


Figure 2.11: Proposal Function of Seller Side in Upwork

As shown in Figure 2.11, this proposal function allow seller to view the offer for them to accept the project or reject the project. It also allows seller to be interview by the client to allow them to handle the project.

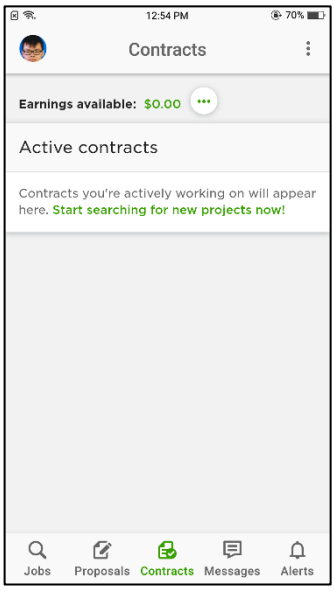


Figure 2.12: Contract Function of Seller Side in Upwork

Figure 2.12 shows that once the seller accepts the project, it is shown in the contract tab to allow them to view. From there, seller is able to click the project they accepted and view the details.

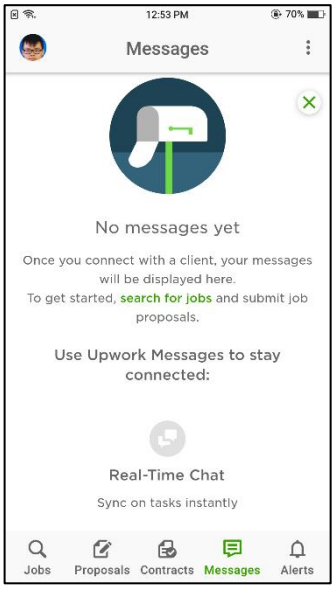
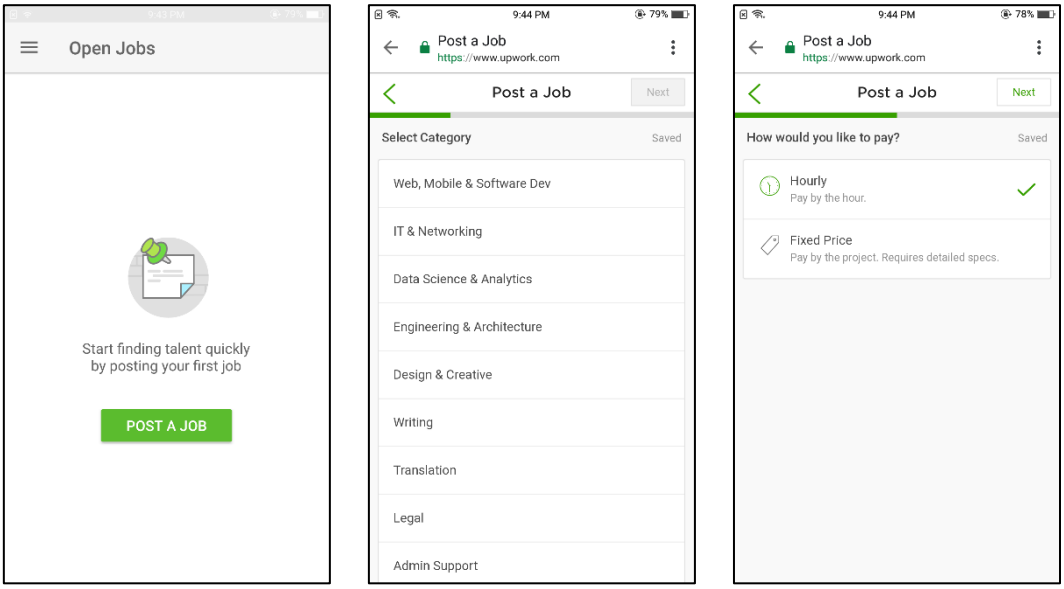


Figure 2.13: Message Tab of Seller Side in Upwork

Upwork also includes the instant message feature as shown in Figure 2.13 for seller to communicate with the client. The message features are arranged according to project. Besides, the features also can view the client status whether they are online or offline.



(a) **(b)** **(c)**

Figure 2.14 (a-c): Post Job Feature of Client Side in Upwork

Figure 2.14 shows that in the client side, client is able to post for job they wish to be complete. It contains different type of categories to be posted for seller to view and work on the project. They also can choose whether the project going to be paid by hourly or a fixed price.

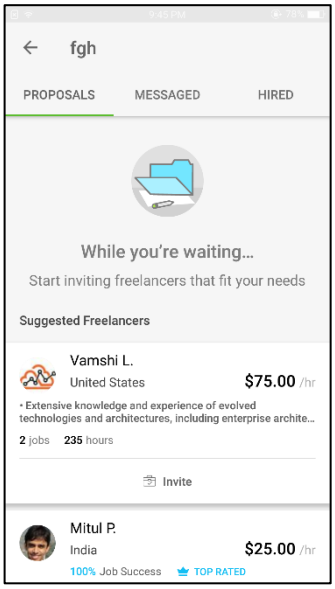


Figure 2.15: Proposal View of Client Side in Upwork

Figure 2.15 shows that client are able to view several proposal from the seller for them to decide which seller they wish to work with. Client can view the profile of suggested freelancers and invite them to complete the project for them.

Strength:

- User friendly
- Able to browse for jobs in seller side
- Seller contain e-wallet

Weakness:

- Require two separate application, one for seller and one for buyer.
- Only able to browse for freelancer after posting jobs in client side

2.2.3 Freelancer.com

Figure 2.16: Logo of Freelancer.com
(Adopted from Freelancer.com, 2009)

Freelancer.com is a global crowdsourcing marketplace, which allow potential employers to post work for freelancers while bidding on in a competitive tender process. Based in Sydney, Australia, Freelancer.com was founded in 2009 by Matt Barrie and also serving as a CEO in the company (Freelancer.com, 2009). Figure 2.16 is the logo of Freelancer.com. The company also has offices in Southern California, Vancouver, London, Buenos Aires, Manila, and Jakarta (Freelancer.com, 2009).

Figure 2.17: First Page of Freelancer.com

Figure 2.17 shows the very first page when enter Freelancer.com application. A message feature allows seller and buyer to communicate with each other. The message feature also includes the function of insert attachment where user can attach a file, image or take picture and send it through the message feature.

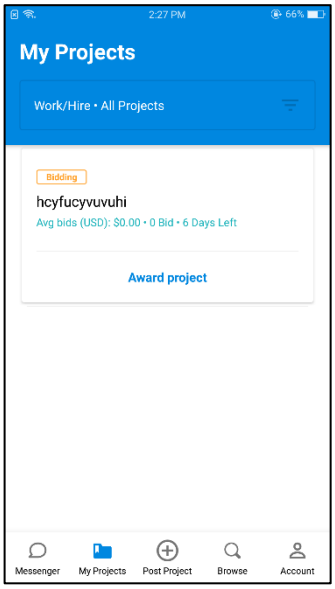


Figure 2.18: Project Tab in Freelancer.com

As shown in Figure 2.18, the project tabs allow buyer to view the project they posted. Here, buyer is able to view various proposal submit by freelancer and award them the project if the freelancer interest them.

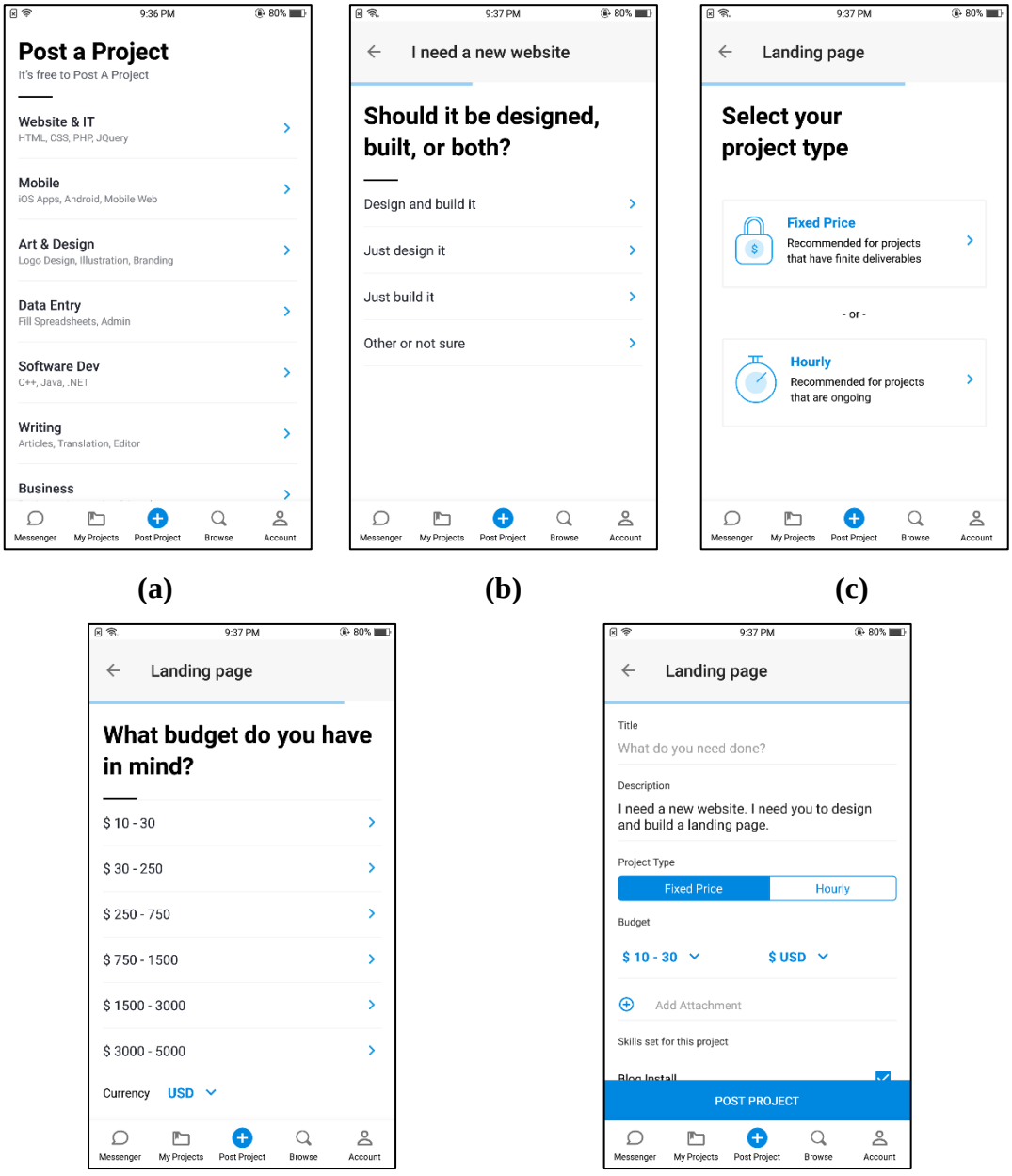
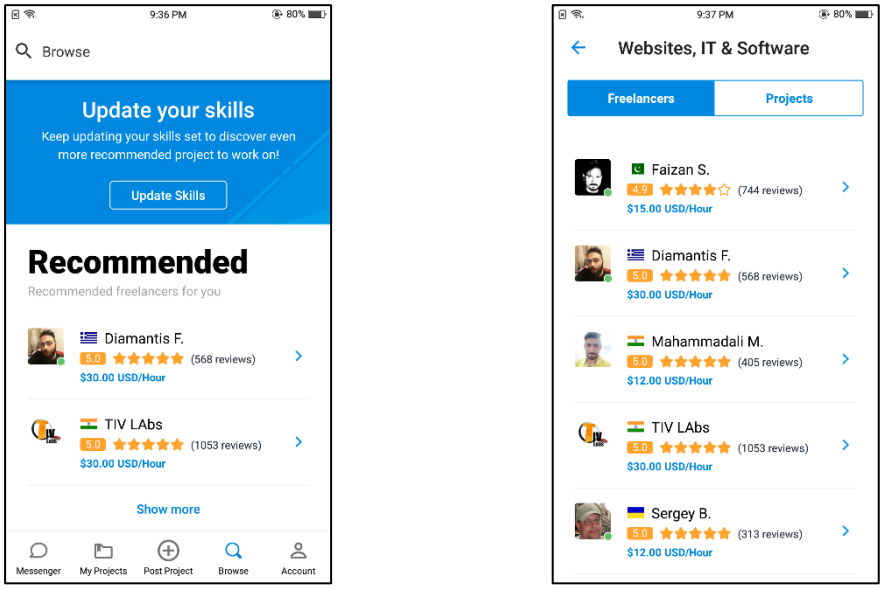


Figure 2.19 (a-e): Posting Jobs in Freelancer.com

Figure 2.19 shows the sequence in order from (a) to (e) of posting a job in Freelancer.com. User can choose which type of category the project fall in, and choose whether they will be paid hourly or fixed price.



(a) **(b)**

Figure 2.20 (a-b): Browse Function in (a) and Search Result in (b)

Figure 2.20 shows the browse function in (a) and search function in (b). It allows user to browse for type of freelancer and search for freelancer to work with. Besides searching for freelancer, user can also search for project.

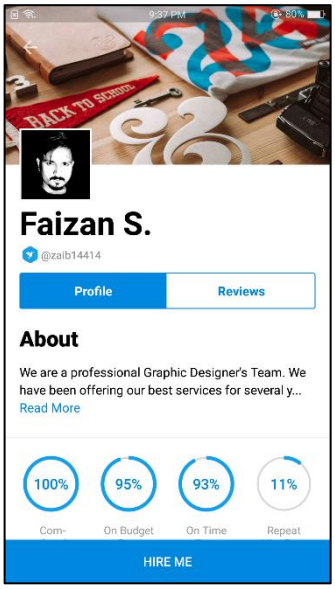


Figure 2.21: Information of Freelancer in Freelancer.com

Figure 2.21 shows the profile of a freelancer, it has information of the freelancer and able to hire them to work without posting any project in the application. When the "HIRE ME" button is clicked, user can choose whether to pay the freelancer by hourly or fixed price, then describe the job and hire them.

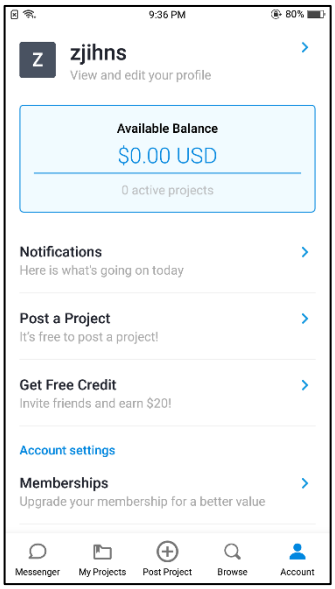


Figure 2.22: E-wallet Function in Freelancer.com

Figure 2.22 show the e-wallet in Freelancer.com. Buyer can deposit funds by using Credit/Debit card or by using PayPal.

Strength:

- User friendly
- Contain e-wallet
- Allow browsing and searching
- Viewing profile of freelancer

Weakness:

- Freelancer have a limited number of bids.
- Freelancer have to purchase premium package in order to bid more.
- Take commission from freelancer before starting a project.

2.2.4 PeoplePerHour

Figure 2.23: Logo of PeoplePerHour

(Adopted from PeoplePerHour, 2007)

PeoplePerHour was established in 2007 by Xenios Thrasyvoulou and Simos Kitiris, the company headquarter is in London, United Kingdom (PeoplePerHour, 2007). Figure 2.23 illustrate the logo of PeoplePerHour. PeoplePerHour function as an online platform providing businesses access to freelancers rather than hiring in-house or via agencies. Jobs can start from little as an hour and can be increased up as needed to build whole team online. They operate as part of what has progressed toward becoming the “talent cloud” with large of freelancers, offerings a range of skills on a flexible and efficient basis. (PeoplePerHour, 2007)

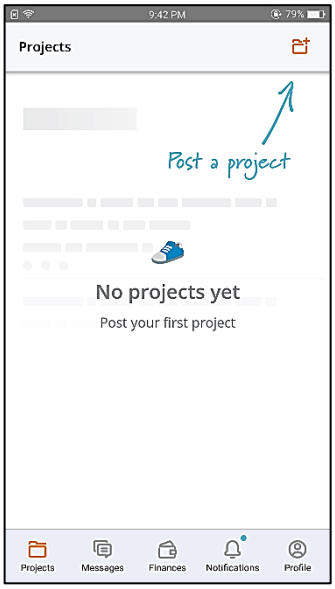


Figure 2.24: First Page of PeoplePerHour

Figure 2.24 shows the first page of the application once enter the application. It is the project page and allow user to post project and view posted project. Before a project is available for viewing and posted, the project needs to be validated by the moderators.

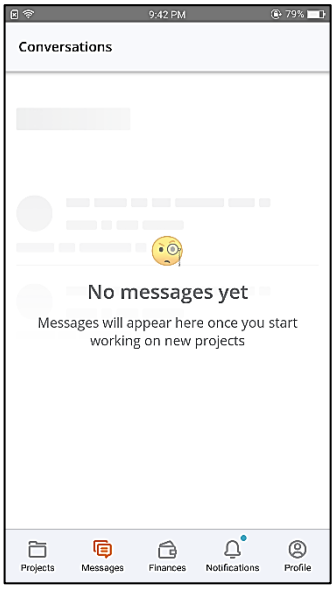


Figure 2.25: Message Features of PeoplePerHour

This application contains the message features as shown in Figure 2.25 that allow user to communicate with freelancer. When the project is accepted by a freelancer, their profile will appear in here, which allow user to easily communicate with them.

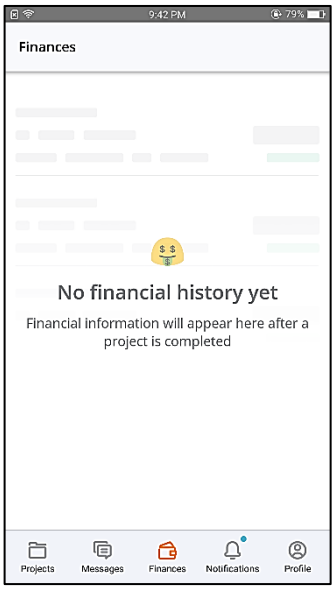


Figure 2.26: Finance Tab in PeoplePerHour

Figure 2.26 shows that PeoplePerHour contain a finance tab that allow user view the financial history on the past project. This tab shows how much has the user earn by completing projects.

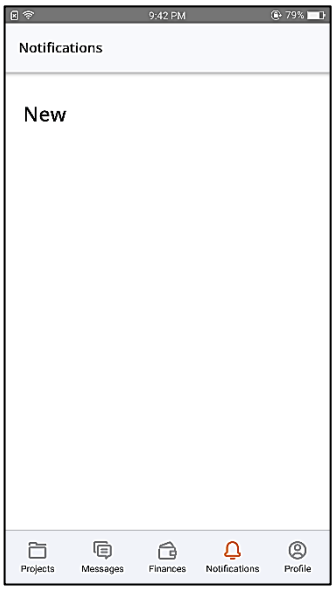


Figure 2.27: Notification Tab in PeoplePerHour

As shown in Figure 2.27, it is the notification tab. If there is an update on the project, the application alerts the user on the phone. Then the notification is shown in this tab.

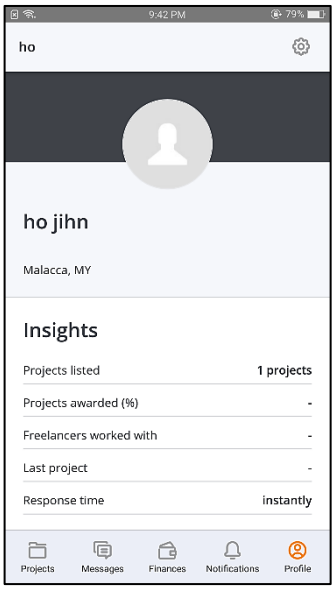


Figure 2.28: Profile Setting in PeoplePerHour

Figure 2.28 shows that the profile setting in PeoplePerHour. User can edit password or email in this tab. Besides, user can also view the insights of their account.

Strength:

- User friendly

Weakness:

- Unable to browse for project or freelancer in the application
- Does not contain e-wallet in the application

2.2.5 Feehour

Figure 2.29: Logo of Feehour
(Adopted from Feehour, n.d.)

The headquarter of Feehour is based in Chuadanga, Khulna. Logo of Feehour is shown in Figure 2.29. Feehour is the fastest growing digital service marketplace for freelance talents across the globe. By using Feehour's platform, companies, entrepreneurs and individuals can hire freelancers and select from the world's largest category of services ranging from graphic design to video editing to online marketing and content writing and a lot more. (LinkedIn, n.d.)

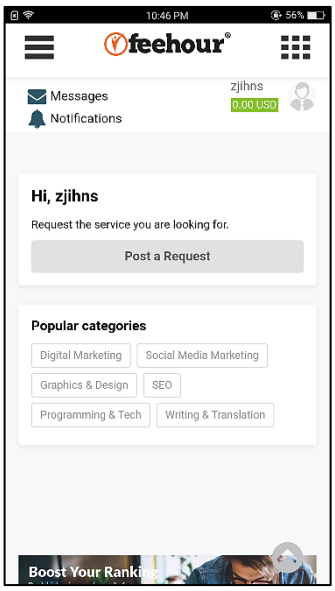


Figure 2.30: First Page of Feehour

As shown in Figure 2.30, it is the first page when accessing the application. Here, user can post a project and set the title, description and price of the project, or browse the category based on popularity.

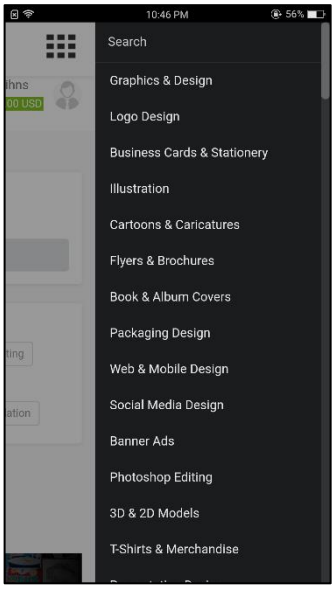


Figure 2.31: Categories Function of Feehour

Figure 2.31 shows that the categories feature that allow user to browse various type of category and search for category. When one of the categories are clicked, the category tab is not automatically closed, user have to close manually and then the app refreshes the page and show the category clicked.

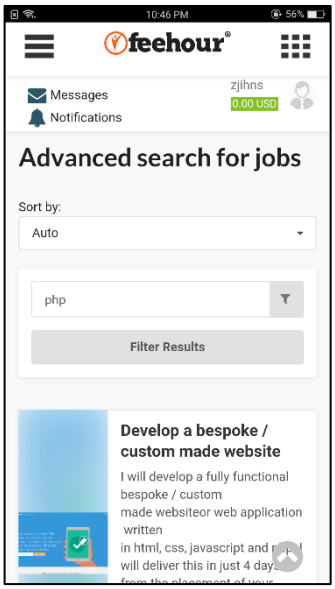


Figure 2.32: Search Function in Feehour

Figure 2.32 shows the search function of Feehour, which allows user to search for different type of job by title. Besides searching, user can also filter the search result to reduce the result.

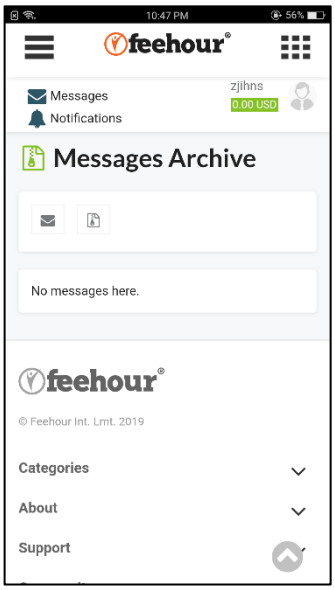


Figure 2.33: Message Features in Feehour

As shown in Figure 2.33, it is the message features in Feehour. It allows user to view message in inbox and create message to chat with other users.

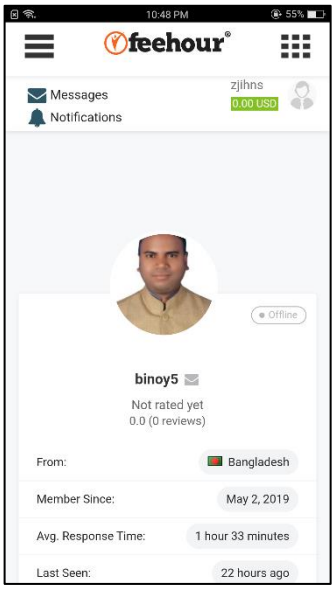


Figure 2.34: Profile of Freelancer in Feehour

Figure 2.34 shows the information of the freelancer. The information shown are where is the seller from, when did they join the platform, their response time etc. It also allow to directly contacting the freelancer.

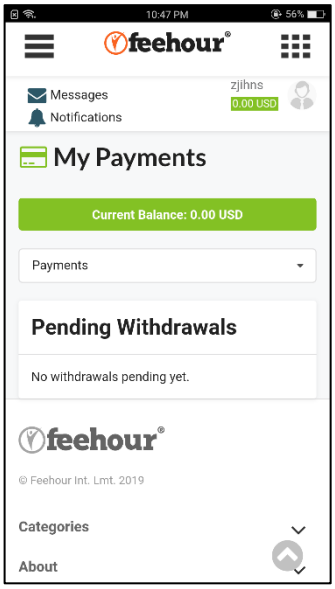


Figure 2.35: E-wallet Feature in Feehour

Figure 2.35 shows the e-wallet feature in Feehour. It is where user can view the balance of fund and status. User can make withdrawals but are required a minimum balance of USD$4.

Strength:

- Able to browse and search for jobs
- Able to view profile of freelancer

Weakness:

- Unable to search for freelancer
- Application is based on web version
- Not user friendly due to application interface is based on the web version.

2.3 Comparison Table

Table 2.1: Comparison between Applications

Applications / Functions	Fiverr	Upwork	Freelancer.com	PeoplePerHour	Feehour
Is the app easy to navigate?	✓	✓	✓	✓	✗
Does the app contain e-wallet feature?	✓	✓	✓	✗	✓
Does the app allow browsing for project?	✓	✓	✓	✗	✓
Does the app allow searching for users?	✓	✗	✓	✗	✗
Does the app allow job searching?	✓	✓	✓	✗	✓
Does the app use image for project/category?	✓	✗	✗	✗	✓
Does the app allow user to view other user's profile?	✓	✓	✓	✗	✓
Does the app charge reasonable commission from freelancer?	✗	✗	✗	✓	✓

From the five applications reviewed, not all has a good design and functions. Table 2.1 presents the comparison between the five applications reviewed. The table includes the comparison of some of the features such as user friendliness, e-wallet features, searching function, profile viewing and commission charges. Some of the application are a based on the web, this cause the navigation through the application harder. The functions from this comparison is used by this project, which are browsing, search function, e-wallet and profile viewing. Besides, to overcome the weaknesses from the five applications, this project include two more function which are set availability of service, set duration of service and e-mail notification for convenience to buyer and seller.

CHAPTER 3

METHODOLOGY

3.1 System Development Life Cycle (SDLC)

3.1.1 Introduction

In order to achieve the assessed objectives and goals in an information system development, it is recommended to implement a scheme for managing and controlling the overall project since the beginning of the project until the end. Supervision or structure that can help synchronize project progression is needed facilitate project operation, including analysing, designing, and other development tasks. The required structure is the one that differentiate all tasks required to develop, launch, and preserve an information system.

Apart from that, the structure also should include all aspect of system analysis, application design, which is the user interface, system programming to make the program work, system testing to ensure the system is bug free, and system maintenance until the system is launched and implemented. This structure is named System Development Life Cycle (SDLC) for this situation. SDLC is a framework determining activities undertaken during each step in the software development process. SDLC is a structure within the software organization which is implemented by a development team. This contains a detailed plan outlining how specific software should be developed, maintained and replaced. The life cycle defines a methodology for enhancing the software quality and the overall development process. (SDLC, n.d.). Figure 3.1 is the example of stages in SDLC.

SDLC's most popular models include the Waterfall Model, Spiral Model, Iterative Model, V-Shaped Model, and Agile Model. Fully described in 1971, the term created in the 1960s when mainframe computers filled whole rooms and developed a pressing need to define procedure and equipment centred on creating large-scale business systems (Guide to SDLC, n.d.). Back then, teams were small-scale, centralized, and clients were 'less' requesting. This sort of circumstances meant that

refined methodologies were not really needed to drive the life of system development. Nevertheless, technology evolved rapidly, systems have become increasingly complex, and clients have become familiar to technology that works well. Models and frameworks were created to lead business through a well-ordered system development life cycle. Today, the traditional ways of dealing with the development of technology system have been adapted to meet the continuous changes, complicated needs of each distinctive organization and their users.

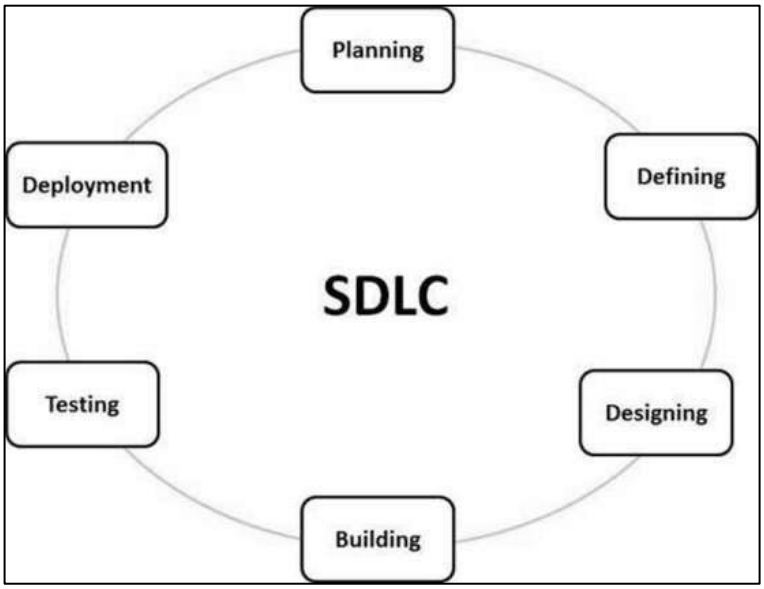


Figure 3.1: Stages of SDLC

3.1.2 Waterfall Model

The Waterfall Model was the first ever to introduce a Process Model. It is also alluded as a model of a linear-sequential life-cycle. Comprehension and utilization are extremely easy. In this model, each phase must be accomplished before the next phase can start, and the phases do not overlap. The Waterfall Model is the earliest SDLC approach which was utilized to develop software. The Waterfall Model demonstrates the development of software in a linear sequential flow as shown in Figure 3.2. This means that any phase in the development process begins only when the preceding phase is finished.

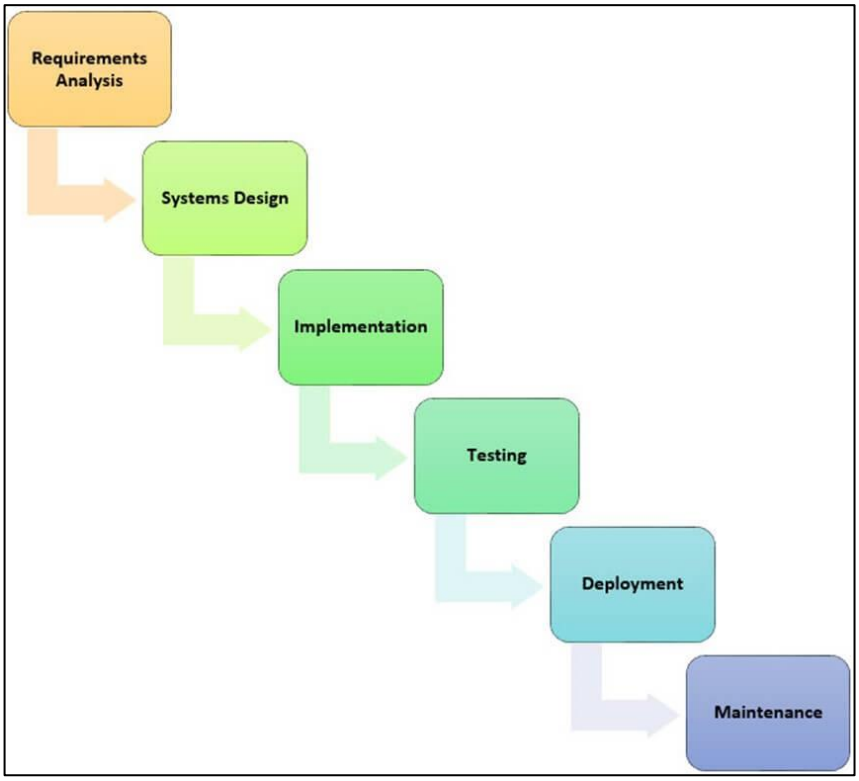


Figure 3.2: Phases of Waterfall Model

Potential requirements, deadlines, and guidelines for the project are studied and put into a functional specification in the Requirements Analysis phase. This stage handles project definition and planning without mentioning specific processes. In Systems Design phase, the requirement specifications from first phase are analysed in this phase and the system design is produced. This system design helps in stating hardware and system needs and aides in interpreting the overall system architecture. Next, the source code is developed in the Implementation phase using the models, logic and requirements identified in the preceding stages. The system is typically designed in smaller piece, or component, before being jointly implemented.

In Testing phase, the entire units developed during implementation phase are combine into a system after all units are tested. Post integration the whole system is evaluated for any shortcoming and failures. In Deployment phase, once testing of functional and non-functional is carried out; the product is deployed in the consumer setting or introduced onto the market. Eventually, corrective, functional and perfective maintenance is carried out continuously in the Maintenance phase to strengthen, upgrade and optimize the end product. This could include the initiation of software patches, or new releases.

3.1.3 Activities in Phases

3.1.3.1 Requirements Analysis

In this phase, meet the supervisor to review and decide the main task of this project. Proposed title, reason I choose to develop this system, how the system work and what is the outcome of this project is explained in this meeting. After acquiring the clear idea of the requirement, we did a research to determine the possibility to develop this system to solve the stated problems. Based on the finding, the required tasks and works to achieve the goals of this project is stated and finalized. Next, a plan where it manages the available time to do each task, task durations and priority is designed.

The next step is to gather information by studying existing applications. Literature review is organized. Since this kind of application is quite common, it contains comparison of five existing applications. These existing applications is studied to determine and understand the features and functions. The strengths and weaknesses are recorded. After that, based on the information gather from literature review, the overall features of the system should be clear at this point, and the process can carry on.

3.1.3.2 Systems Design

In this phase, diagrams are created to assists developer to develop the application. Data flow diagram shows in more details all the functions that are provided by the system and the inputted and outputted data. Use case diagram shows what user can operate in the system. Flow chart diagram shows the steps for buyer to buy services from seller and seller to sell services to buyer.

3.1.3.3 Implementation

In this implementation phase, all the design and prototype are applied and turn into the system. Coding is performed, and the specifications for physical design are transformed into working computer code. Testing is carried out in parallel with coding to make sure the system meets the requirements laid down. Lastly, installation where the original system is replaced by the new one.

3.1.3.4 Testing

In this phase, the system is tested to determine whether it has met with the requirements and specifications of this project. The system are fixed if any error occurred. After that, it is ready to be tested by selected end-users for Alpha version..

3.1.3.5 Deployment

In Deployment phase, after the system is tested and ready to go live, it is uploaded to Google Play Store for people to download.

3.1.3.6 Maintenance

In the last phase of Waterfall Model, we have to make sure our application is up and running. Every function is working, as it should be. If user report a bug, find solution and fix immediately. If the bug is fixed and new features are incorporate, notify the users.

3.2 Tools and Techniques

3.2.1 Android Studio

Figure 3.3: Logo of Android Studio

(Adopted from Android Studio, 2014)

Android Studio is the official integrated development environment (IDE) for Android application development. As shown in Figure3.3 is the logo of Android Studio. Based on the IntelliJ IDEA, a Java built-in software development environment, and includes its developer tools and code editing. Android Studio uses a Gradle-based build system, code templates, GitHub integration, and emulator to help application development within the Android operating system. Each Android Studio project has at least one modality that includes source code and resource files. These modalities comprise of modules for the Library modules, Android App, and modules for the Google App Engine. The software was introduced in May 2013 at Google I/O, with the first stable build was launched in December 2014. Android Studio is accessible for desktop platforms running on Windows, Mac and Linux.

3.2.2 Java

Figure 3.4: Logo of Java

Java is a popular, object-oriented and class-based programming language. Logo of Java is as shown in Figure 3.4. Java is developed in 1995 and owned by Oracle (Java Introduction, n.d.). After the Java code has been compiled into bytecode, it is run securely on all the other platforms that has Java Virtual Machine (JVM) (Rouse, n.d.). Java is chosen for being secure, fast and reliable. In fact, several applications and even websites that are created much more daily need to have Java installed in order to be able to run. Java is all over the place. Mobile phones, game consoles, workstation, scientific computers, laptops, internet. All these things have Java installed and nearly all of their applications and programs are Java-based. In order to code a Java program, Java Development Kit (JDK) is required. The Java Development Kit an environment for software development utilized to build Java applications and applets. It comprises the Java Runtime Environment (JRE), an interpreter/loader (java), an archiver (jar), compiler (javac), a documentation generator (javadoc) and other tools required for Java development.

3.2.3 Firebase

Figure 3.5: Logo of Firebase

(Adopted from Firebase, 2011)

Firebase is a Backend-as-a-Service — BaaS — that began as a YC11 company and grew up into a cutting-edge app-development platform on Google Cloud Platform. Figure 3.5 illustrate the logo of Firebase. Firebase is utilized for database purpose in this project as this platform provides numerous services where it helps developers to build a better application, enhance the application's quality, and expand the business widely.

3.2.4 GIMP

Figure 3.6: Logo of GIMP

(Adopted from GIMP, 1996)

GNU Image Manipulation Program (GIMP) is a cross-platform image editor accessible for GNU/Linux, Windows, OS X and more operating systems (GIMP, 1996). As shown in Figure 3.6 is the logo of GIMP. It is release in February 1996. It is an open source and a freely distributed program for tasks such as image composition, photo retouching, and image authoring. It has many abilities. It can be utilized as a basic painting program, an expert quality photo retouching program, a large scale production image renderer, an online batch processing system, an image format converter, etc. GIMP is chosen because it is free and offer a variety of features.

3.2.5 Justinmind

Figure 3.7: Logo of Justinmind

(Adopted from Justinmind, 2015)

Justinmind is a prototyping tool that enables users to create interactive, realistic simulations of mobile applications and web (Justinmind, 2015). It offers abilities typically found in the diagramming tools, such as placement of drag and drop, rescaling, formatting and import/export of widgets.

CHAPTER 4

PROPOSED SOLUTION AND IMPLEMENTATION PLAN

4.1 Functional Design

4.1.1 Context Diagram

The context diagram shows the overview of the entire information system and the method of interacting with external entities.

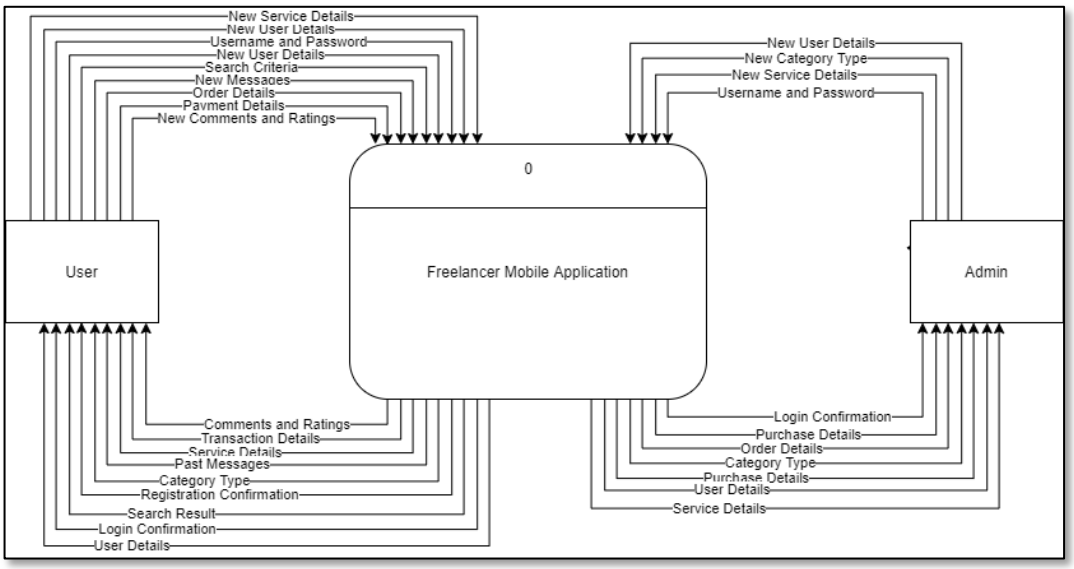


Figure 4.1: Context Diagram

4.1.2 Data Flow Diagram

Data Flow Diagram illustrates the flows of information through a process or system in terms of inputs and outputs.

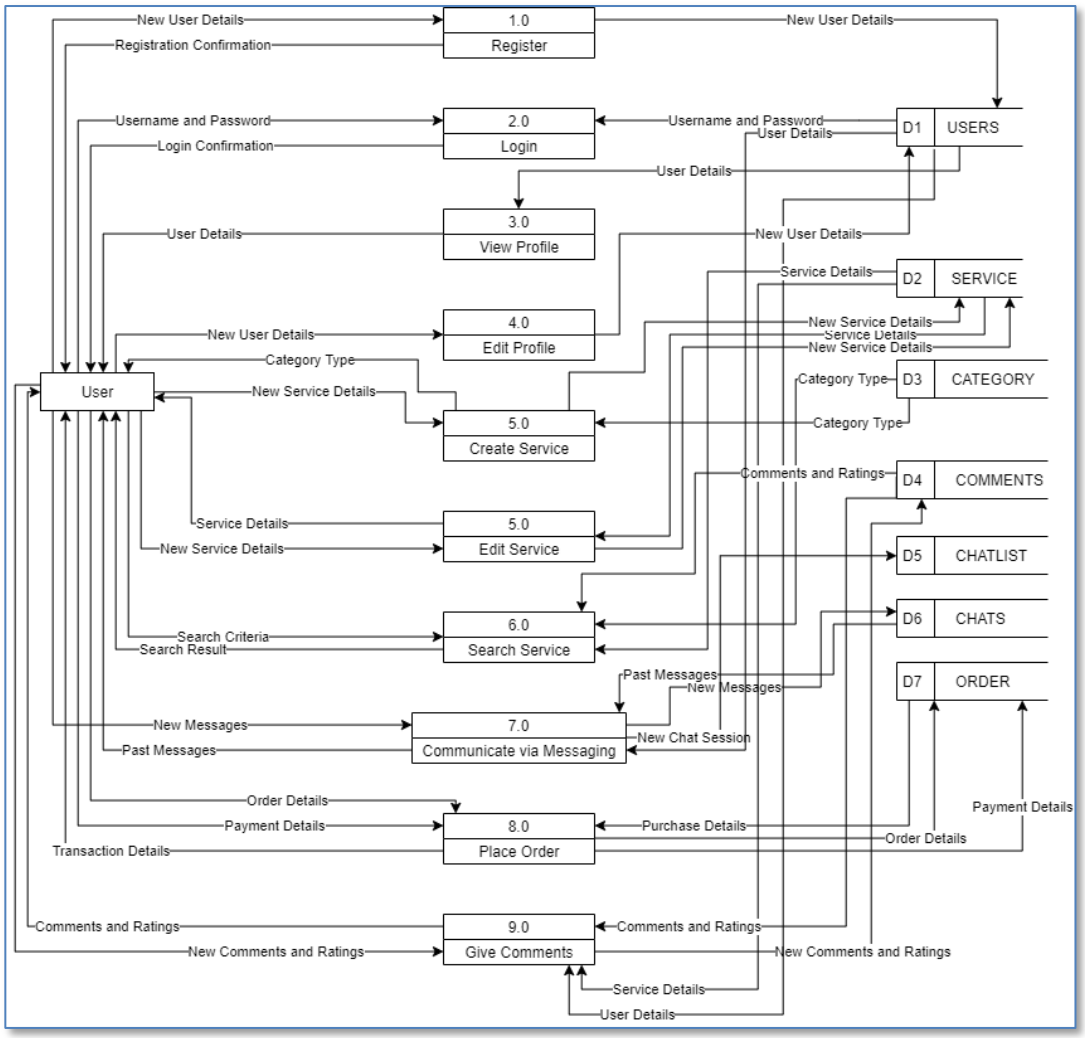


Figure 4.2: Data Flow Diagram for User

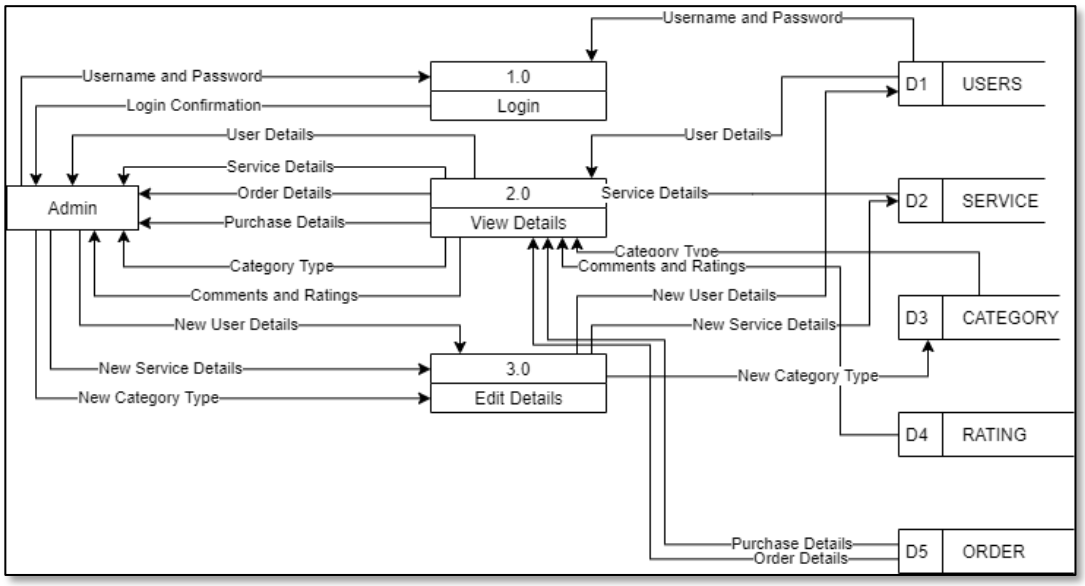


Figure 4.3: Data Flow Diagram for Admin

4.1.3 Use Case Diagram

A use case diagram is to describe the behaviour of the target system by the view of external point of view.

Figure 4.4: Use Case Diagram

4.1.4 Flow Chart

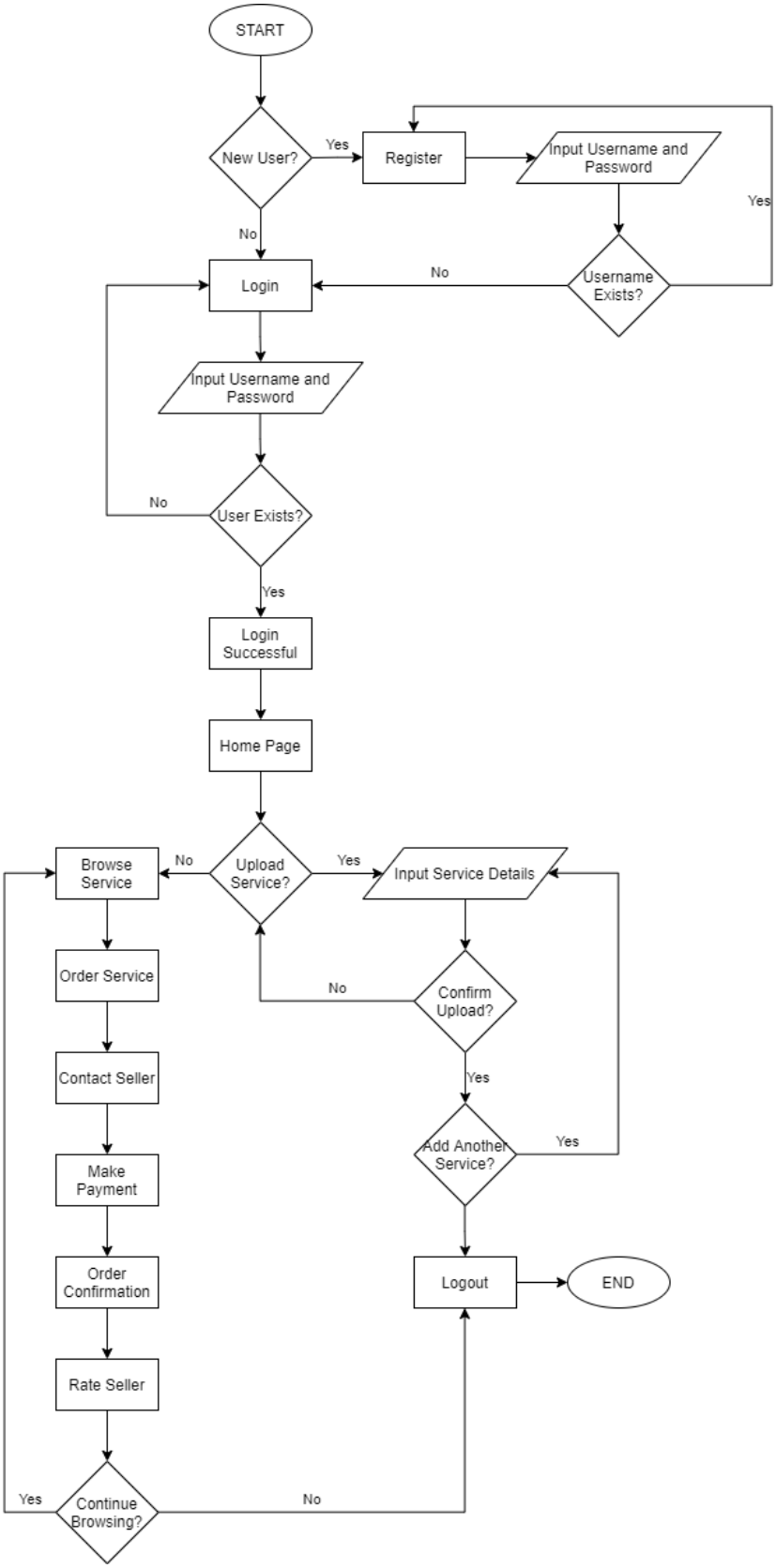


Figure 4.5: Flow Chart Diagram

4.1.5 System Organization Chart

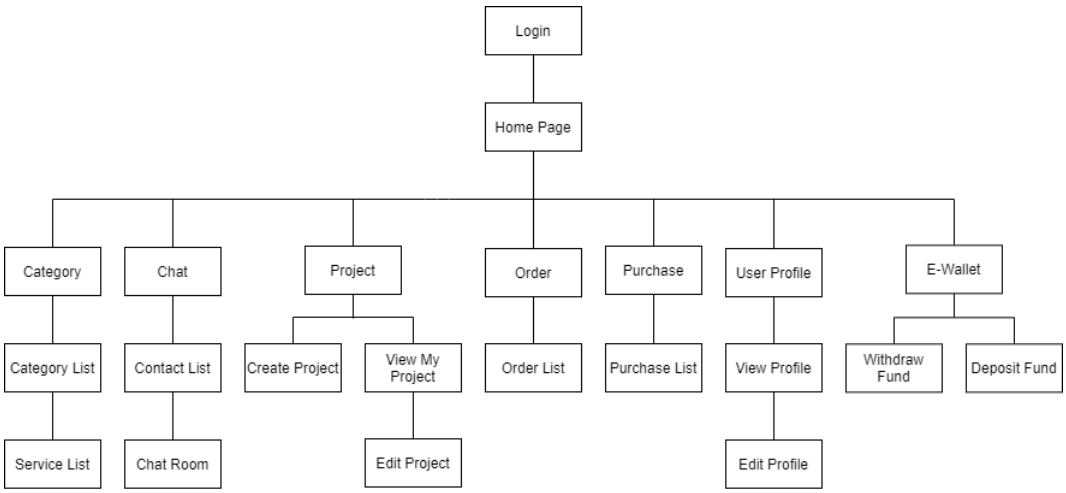


Figure 4.6: System Organization Chart

4.2 Database Design

4.2.1 Business Rule

I. One User can make one or many Order.
II. Each Order is made by one User.
III. One User can create one or many Service.
IV. Each Service belong to one User.
V. One Order can contain one or many Service.
VI. One Service belong to one or many Order.
VII. One Category can contain one or many Service.
VIII. One Service belong to one Category.
IX. One User can create one or many Chatlist.
X. Each Chatlist belong to one and only one User
XI. One Chatlist can contain one or many Chat.
XII. Each Chatlist belong to one and only one Chatlist.
XIII. User can make one or many Rating.
XIV. Each Rating belong to one and only one User.
XV. One User has zero or many Rating.
XVI. Each Rating belong to one and only one User.
XVII. One Chat creates one and only one Token.
XVIII. Each Token belong to one and only one Chat.

4.2.2 ER Diagram

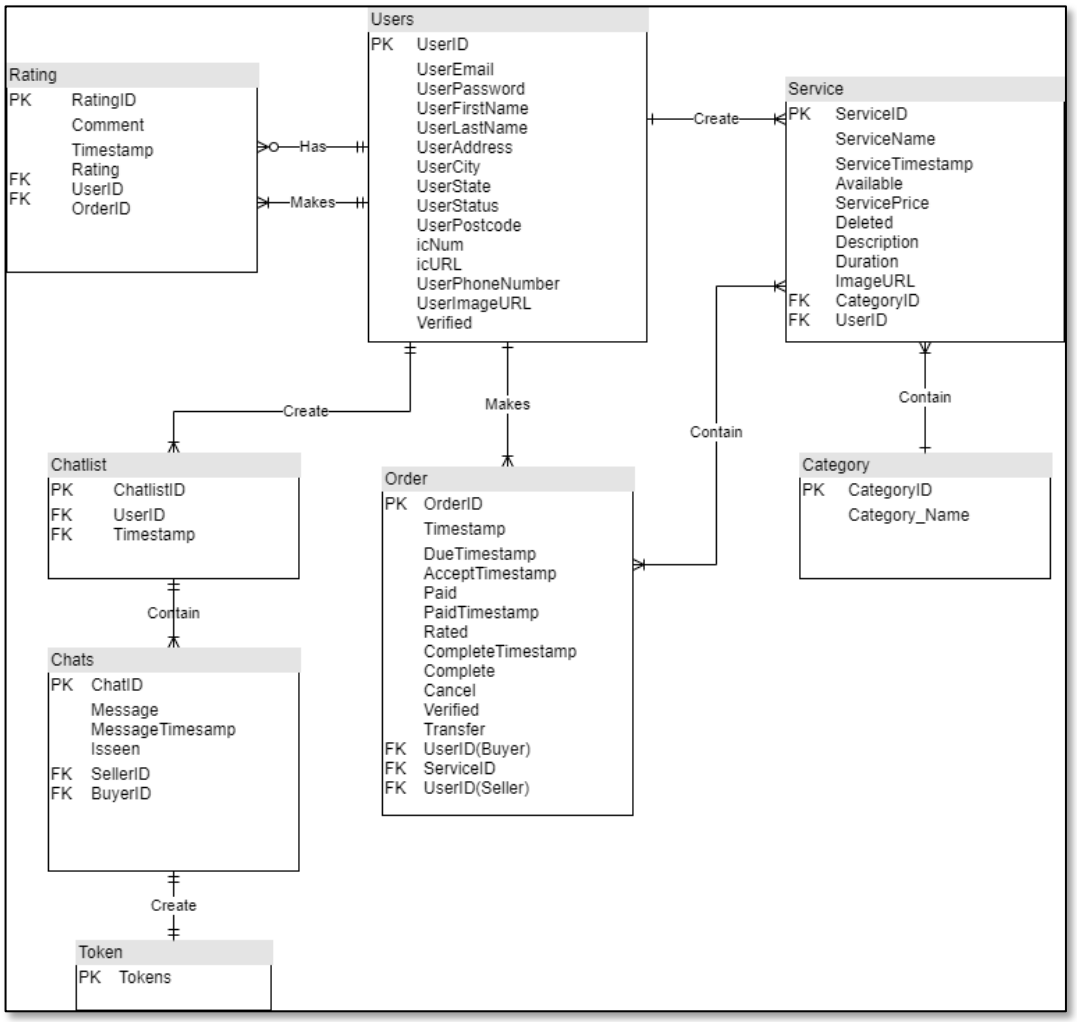


Figure 4.7: Entity Relationship Diagram

4.2.3 List of Tables

Table Name	Table Description
USER	This table is used to store the information of user.
SERVICE	This table is used to store the details of the services upload by the seller.
ORDER	This table is used to store details of order made by buyer.
CATEGORY	This table is used to store the list of category added by the admin.
RATING	This table is used to store the comments and ratings given by the buyer to seller.
CHATS	This table is used to store the chat's message between the users.
CHATLIST	This table is used to store the chat session of the user.
TOKENS	This table is used to store the notification tokens.

4.2.4 Table Design

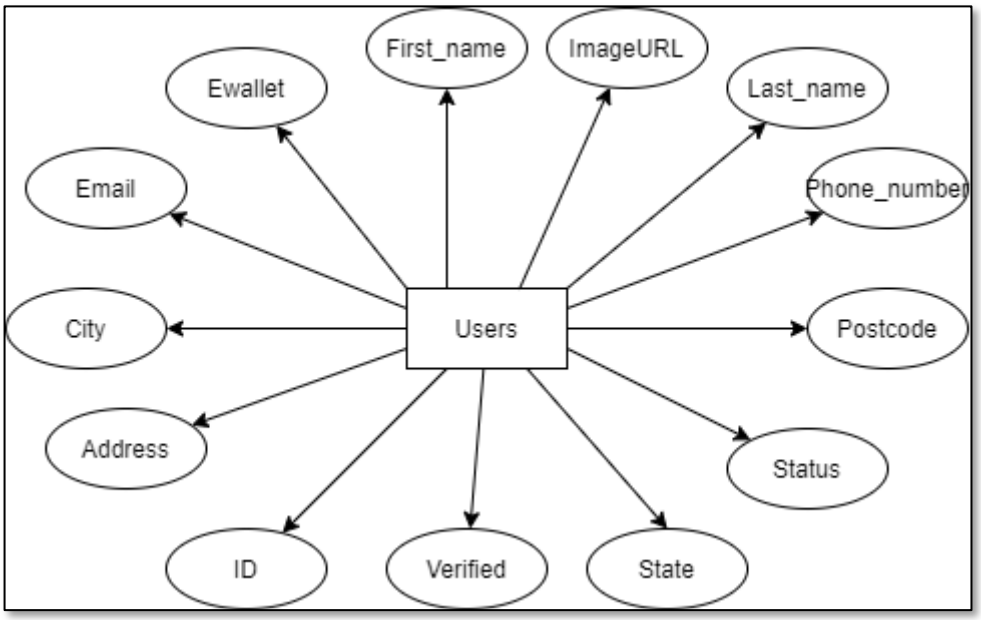


Figure 4.8: Table Design for Users

Figure 4.9: Table Design for Category

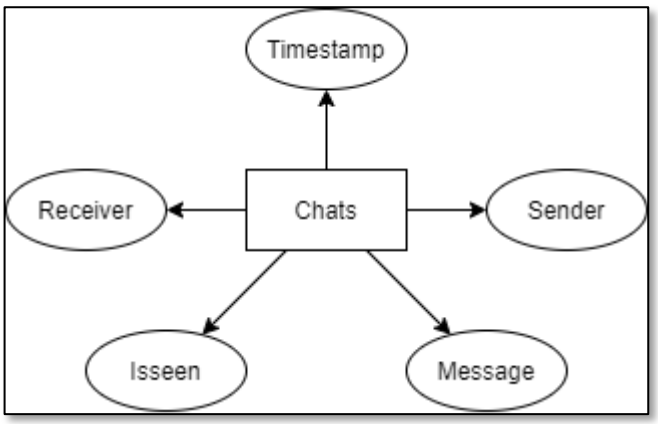


Figure 4.10: Table Design for Chats

Figure 4.11: Table Design for Chatlist

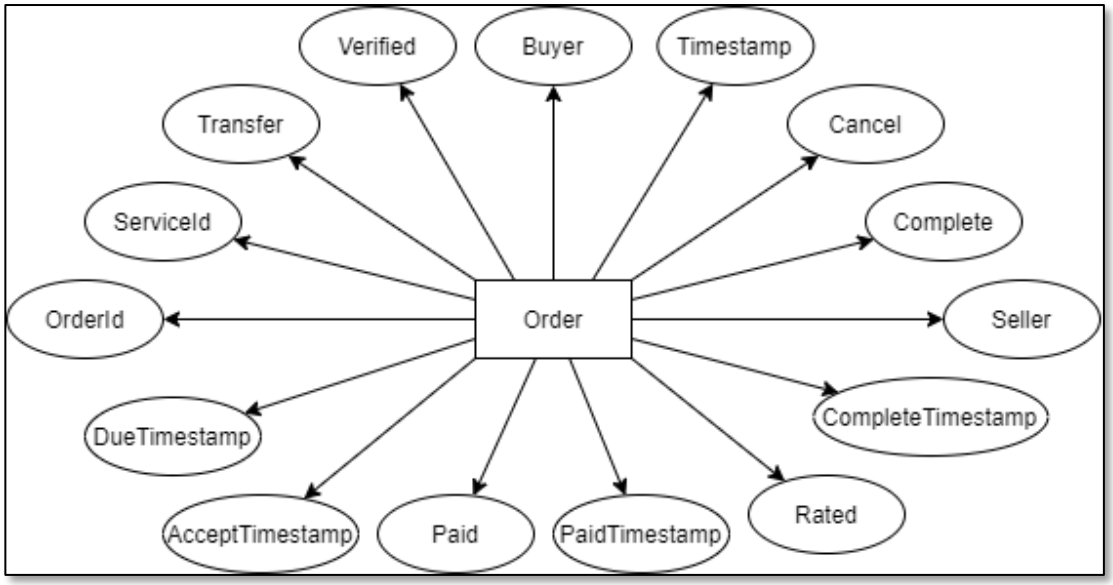


Figure 4.12: Table Design for Order

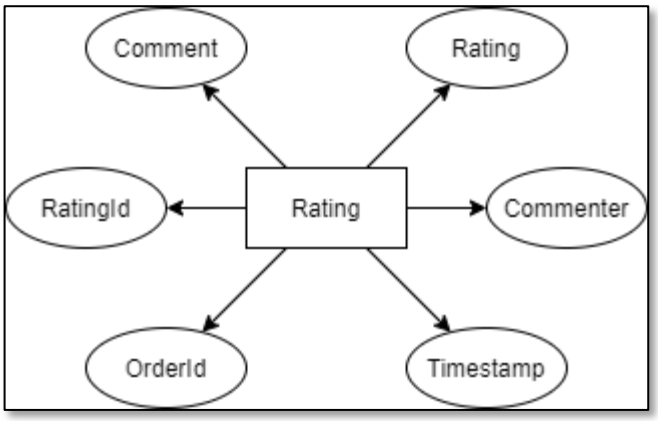


Figure 4.13: Table Design for Rating

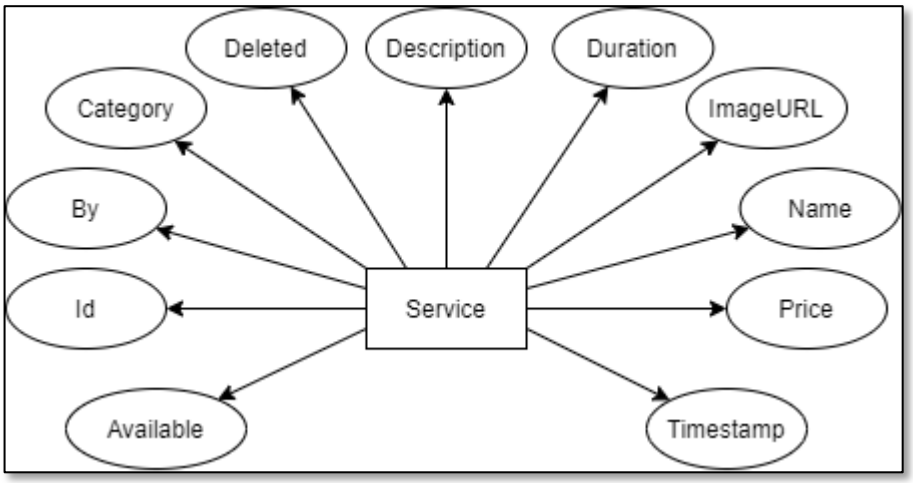


Figure 4.14: Table Design for Service

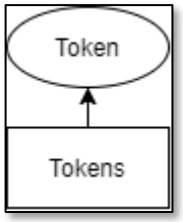


Figure 4.15: Table Design for Tokens

4.2.5 Data Dictionary

Table Name	Attribute Name	Content	Type	Format	Range	Required	PK & FK	FK Reference Table
Users	Id	User's ID	String	xxxxxxxxxxxx		YES	PK	
	Address	User's Address	String	xxxxxxxxxxxx		YES		
	City	User's City	String	xxxxxxxxxxxx		YES		
	Email	User's Email	String	xxxxxxxxxxxx		YES		
	Ewallet	User's Ewallet	Float	99999	1-99999	YES		
	First_name	User's First Name	String	xxxxxxxxxxxx		YES		
	ImageURL	User's Profile Image	String	xxxxxxxxxxxx		YES		
	icNum	User's Identification Number	String	xxxxxxxxxxxx		YES		
	icURL	User's Identification Image	String	xxxxxxxxxxxx		YES		
	Last_name	User's Last Name	String	xxxxxxxxxxxx		YES		
	Phone_number	User's Phone Number	String	xxxxxxxxxxxx		NO		
	Postcode	User's Post Code	String	xxxxxxxxxxxx		YES		
	State	User's State	String	xxxxxxxxxxxx		YES		
	Status	User's Status	String	xxxxxxxxxxxx		YES		
	Verified	User's Verification Status	Boolean	True/False		YES		
Category	Id	Category's ID	String	xxxxxxxxxxxx		YES	PK	
	Name	Category's Name	String	xxxxxxxxxxxx		YES		
Chats	Isseen	Chats' Isseen Flag	Boolean	True/False		YES		
	Message	Chats' Message	String	xxxxxxxxxxxx		YES		
	Receiver	Chats' Receiver's ID	String	xxxxxxxxxxxx		NO	FK	Users
	Sender	Chats' Sender's ID	String	xxxxxxxxxxxx		YES	FK	Users
	Timestamp	Chats' Timestamp	String	xxxxxxxxxxxx		YES		
Chatlist	Id	Chatlist's ID	String	xxxxxxxxxxxx		YES	PK	
	Timestamp	Chatlist's Timestamp	String	xxxxxxxxxxxx		YES		
Order	OrderId	Order's ID	String	xxxxxxxxxxxx		YES	PK	
	AcceptTimestamp	Order's Accept Timestamp	String	xxxxxxxxxxxx		YES		
	Buyer	Order's Buyer ID	String	xxxxxxxxxxxx		YES	FK	Users
	Cancel	Order's Cancel Flag	Boolean	True/False		YES		
	Complete	Order's Complete Flag	Boolean	True/False		YES		
	CompleteTimestamp	Order's Complete Timestamp	String	xxxxxxxxxxxx		YES		
	DueTImestamp	Order's Due Timestamp	String	xxxxxxxxxxxx		YES		
	Paid	Order's Paid Flag	String	xxxxxxxxxxxx		YES		
	PaidTimestamp	Order's Paid Timestamp	String	xxxxxxxxxxxx		YES		
	Rated	Order's Rated Flag	Boolean	True/False		YES		
	Seller	Order's Seller's ID	String	xxxxxxxxxxxx		YES	FK	Users
	ServiceId	Order's Service's ID	String	xxxxxxxxxxxx		YES	FK	Service
	Timestamp	Order's Timestamp	String	xxxxxxxxxxxx		YES		
	Transfer	Order's Transfer Flag	Boolean	True/False		YES		
	Verified	Order's Verified Flag	Boolean	True/False		YES		
Rating	RatingId	Rating's ID	String	xxxxxxxxxxxx		YES	PK	
	Comment	Rating's Comment	String	xxxxxxxxxxxx		YES		
	Rating	Rating's Value	Float	99999		YES		
	Commenter	Rating's Commenter's ID	String	xxxxxxxxxxxx		YES	FK	Users
	Timestamp	Rating's Timestamp	String	xxxxxxxxxxxx		YES		
	OrderId	Rating's Order's ID	String	xxxxxxxxxxxx		YES	FK	Order
Service	Id	Service's ID	String	xxxxxxxxxxxx		YES	PK	
	Available	Service's Available Flag	Boolean	True/False		YES		
	By	Service's Owner's ID	String	xxxxxxxxxxxx		YES	FK	Users
	Category	Service's Category's ID	String	xxxxxxxxxxxx		YES	FK	Category
	Deleted	Service's Deleted's ID	Boolean	True/False		YES		
	Description	Service's Description	String	xxxxxxxxxxxx		YES		
	Duration	Service's Duration	String	xxxxxxxxxxxx		YES		
	ImageURL	Service's Image	String	xxxxxxxxxxxx		YES		
	Name	Service's Name	String	xxxxxxxxxxxx		YES		
	Price	Service's Price	String	xxxxxxxxxxxx		YES		
	Timestamp	Service's Timestamp	String	xxxxxxxxxxxx		YES		
Tokens	Token	Tokens's Token ID	String	xxxxxxxxxxxx		YES	PK	

Figure 4.16: Data Dictionary

4.3 Screen Design and Description

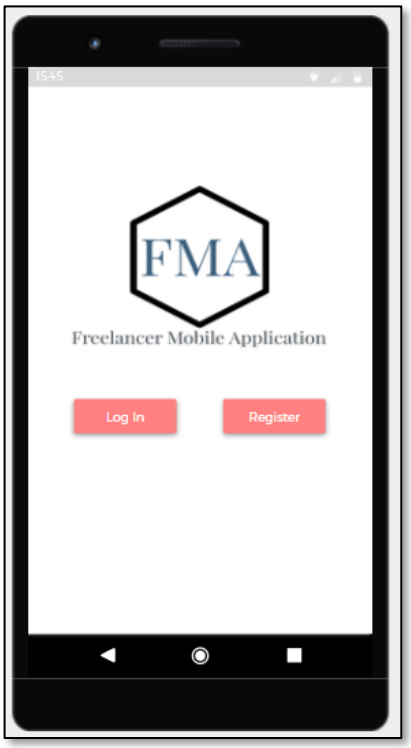
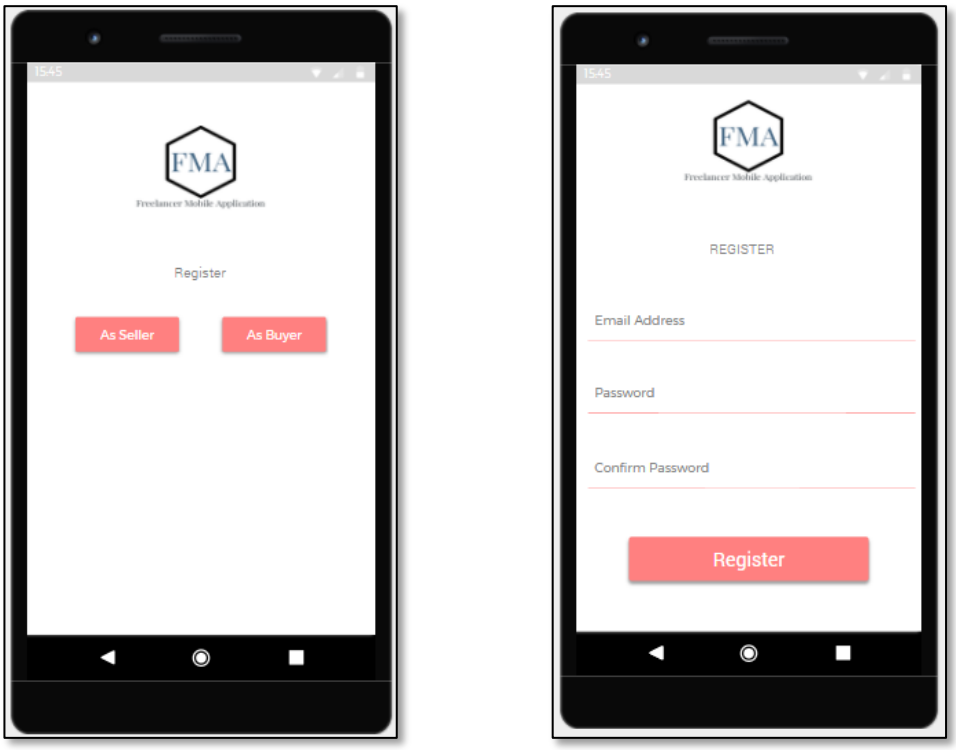


Figure 4.17: Welcome Page

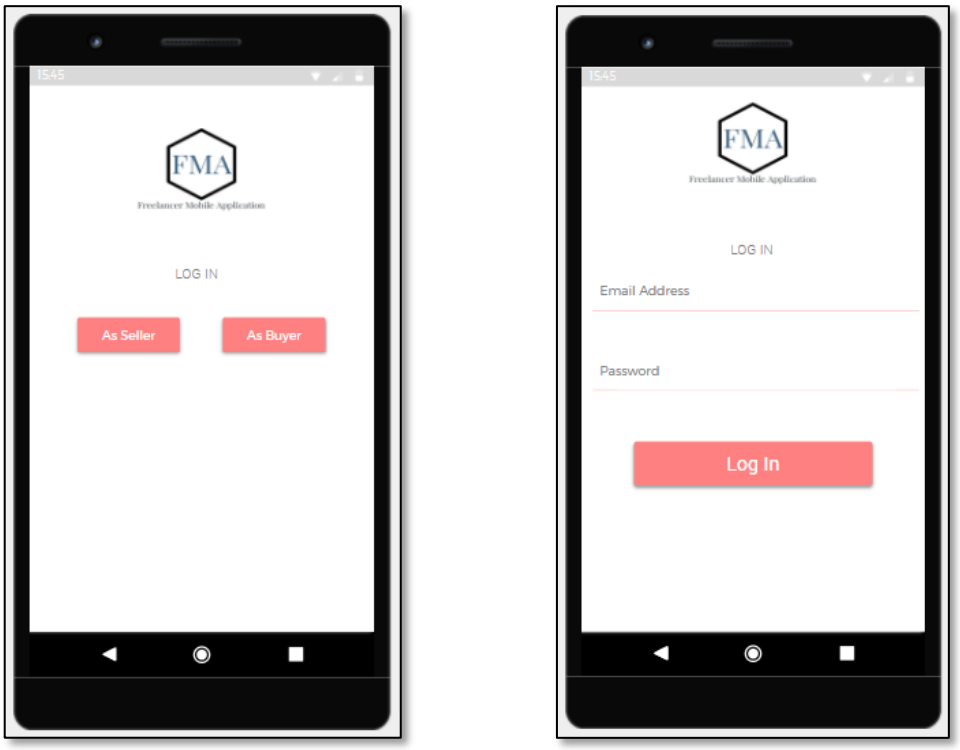
Figure 4.9 show the very first page of the application when it is opened. In this page, there are two option for user to choose, either to register as a user or log in as a user.



(a) **(b)**

Figure 4.18 (a-b): Register Page

When user chose to register, it will allow user to choose user to register as a buyer or a seller as shown in Figure 4.10(a). When user choose either one, it will lead user to registration page as shown in Figure 4.10(b). The application will only ask for e-mail, password and confirmation of password to bring convenience to user. User can enter more of their details in the setting when logged in.



(a) **(b)**

Figure 4.19 (a-b): Login Page

After registration successful, user will be able to log in as a seller or as a buyer as shown in Figure 4.11(a). Figure 4.11(b) shows user only need to enter their e-mail and password in order to log in to this application.

Figure 4.20: Homepage

As shown in Figure 4.12 is the home page of this application. Here, user are able to view the newest addition of category and what they recently viewed. Besides, they able to choose to navigate through the application by using the navigation bar and the bottom or by using menu button in the top left corner.

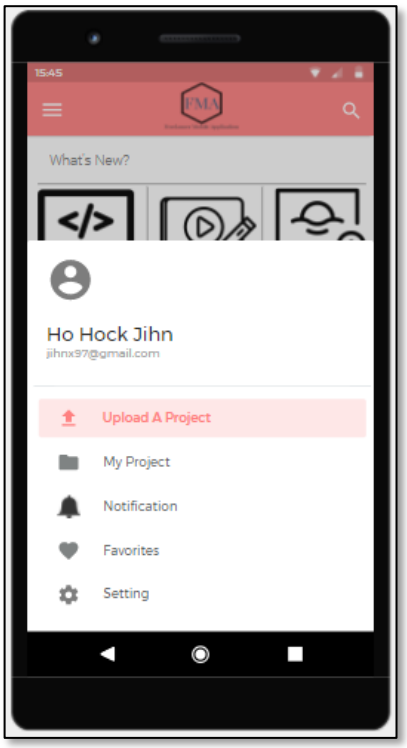
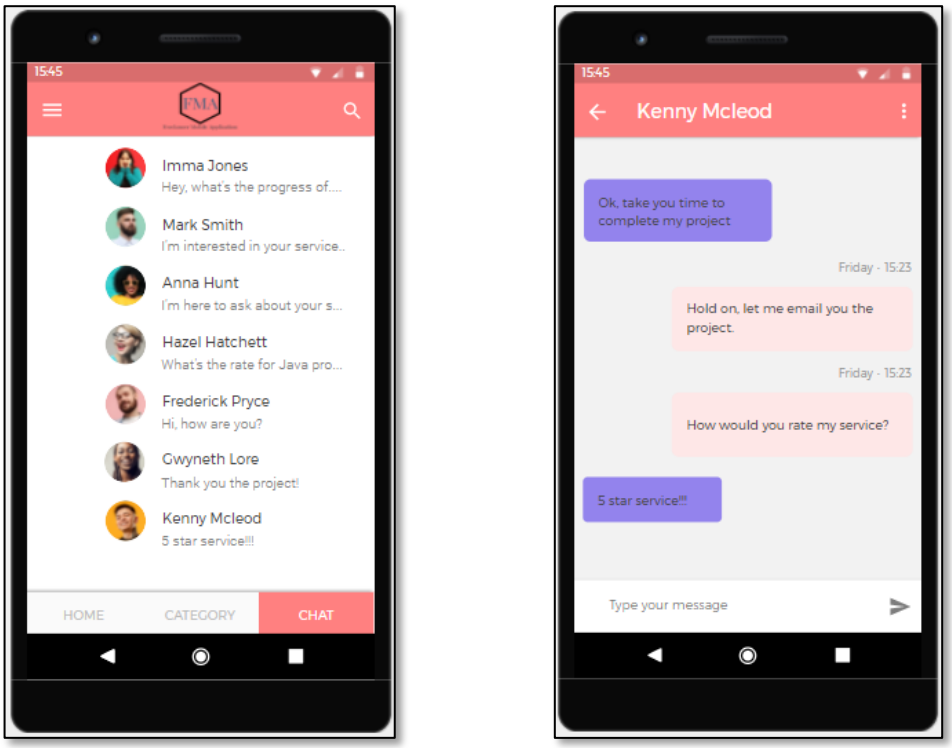


Figure 4.21: Menu

When user press the menu button at the top left corner, it will pop up their brief details about their name and the e-mail as shown in Figure 4.13. In addition, it will also allow user to upload project if user is a seller.

Figure 4.22: Category Page

Figure 4.14 shows the category page of the application. Here, it shows the entire category that allow user to browse through it. The category are added by the admin.



(a) **(b)**

Figure 4.23 (a-b): Chat Page

Figure 4.15 shows the chat function in this application. In Figure 4.15(a) is contact list that the user recently chat with. When clicked one of those contact, it will allow user to chat the them as shown in Figure 4.15(b).

4.4 Implementation Plan

4.4.1 Milestone

The task in the first phase of this project have been listed in the Gantt chart in Figure 4.29 and the task in the second phase of this project have been listed in the Gantt chart in Figure 4.30. The purpose of the diagram is to determine how much progress have been made based on the number of weeks.

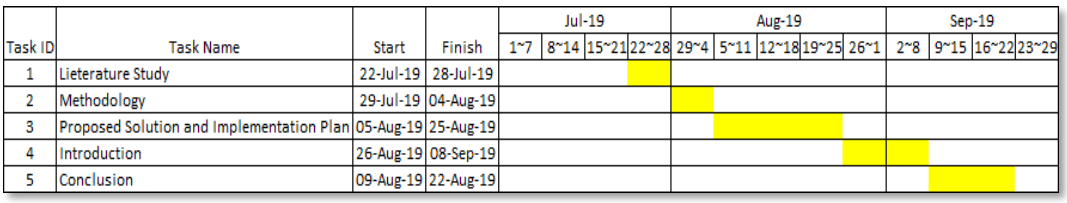


Figure 4.24: First Phase Timeline

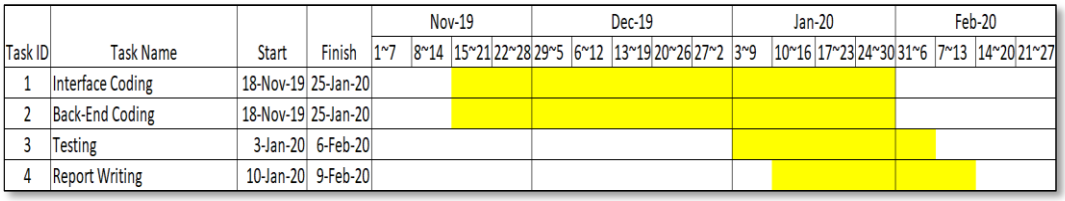


Figure 4.25: Second Phase Timeline

CHAPTER 5

THE SOLUTION

5.1 Preparation and Setup

5.1.1 Android Studio

Android Studio allow developer to develop an android application. To start with it, go to https://developer.android.com/studio and download the latest version.

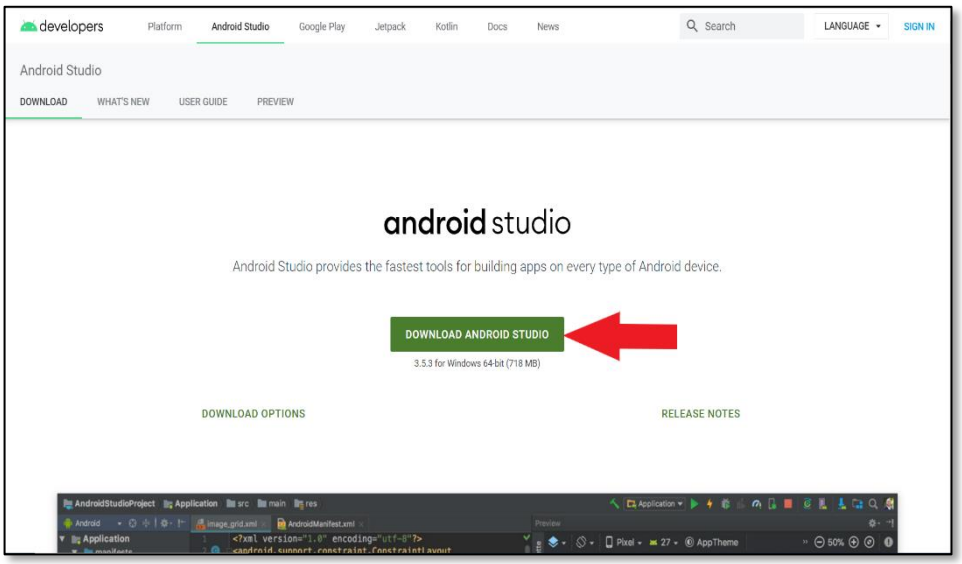


Figure 5.1: Android Studio Webpage

After the download is completed, follow the installation guide and install the program.

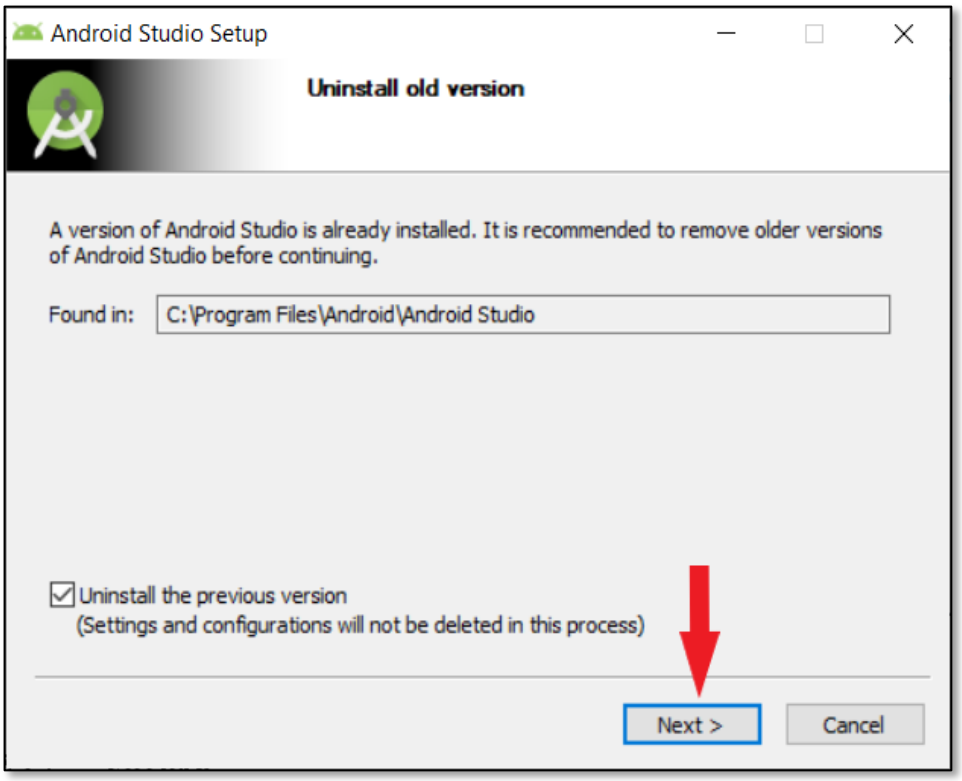


Figure 5.2: Android Studio Installation

After the installation is complete, the program is ready to use and develop an android application.

5.1.2 Firebase

Firebase is a database developed by Google to allow developer to store data in it. To start using it, first go to https://firebase.google.com/ and either sign in using any Google account or sign up if do not have any Google account. After signing in, press "Get Started" to continue with Firebase setup.

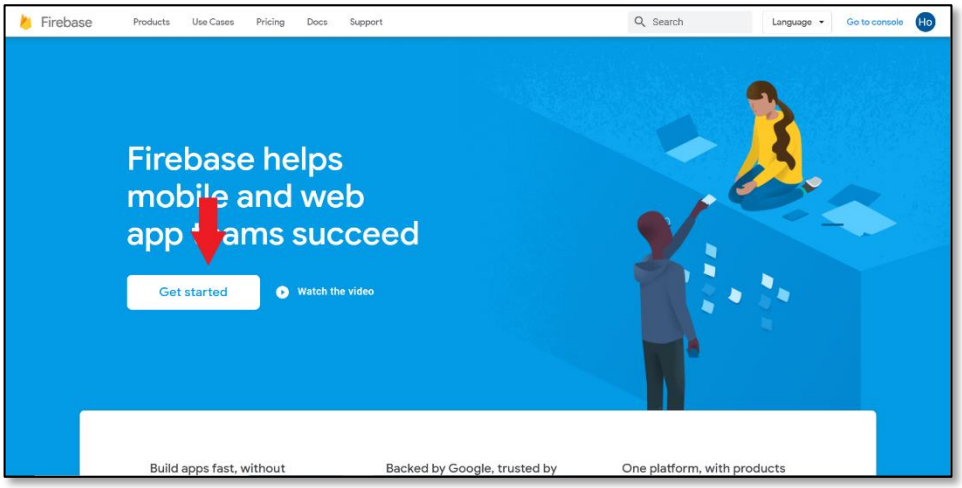


Figure 5.3: Firebase Webpage (Logged In)

Add a new project and enter the project name.

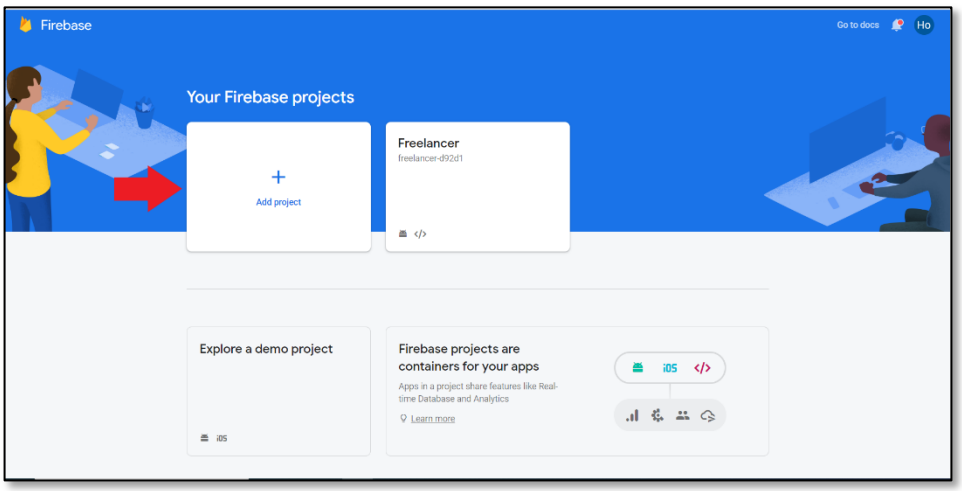


Figure 5.4: Firebase Project Page

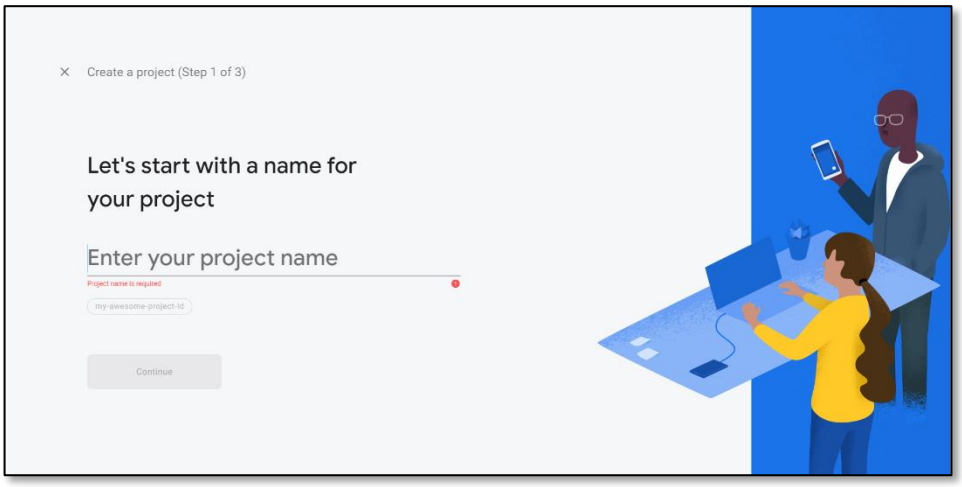


Figure 5.5: Firebase (Insert project name)

Click “Database”. And scroll down to look for Realtime Database and click “Create Database”.

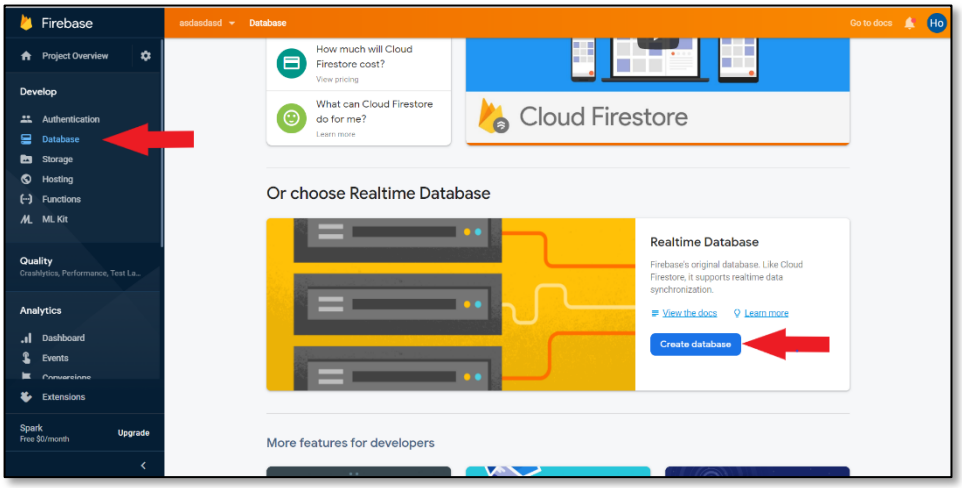


Figure 5.6: Firebase Console Page (Database Page)

Select “Start in test mode”.

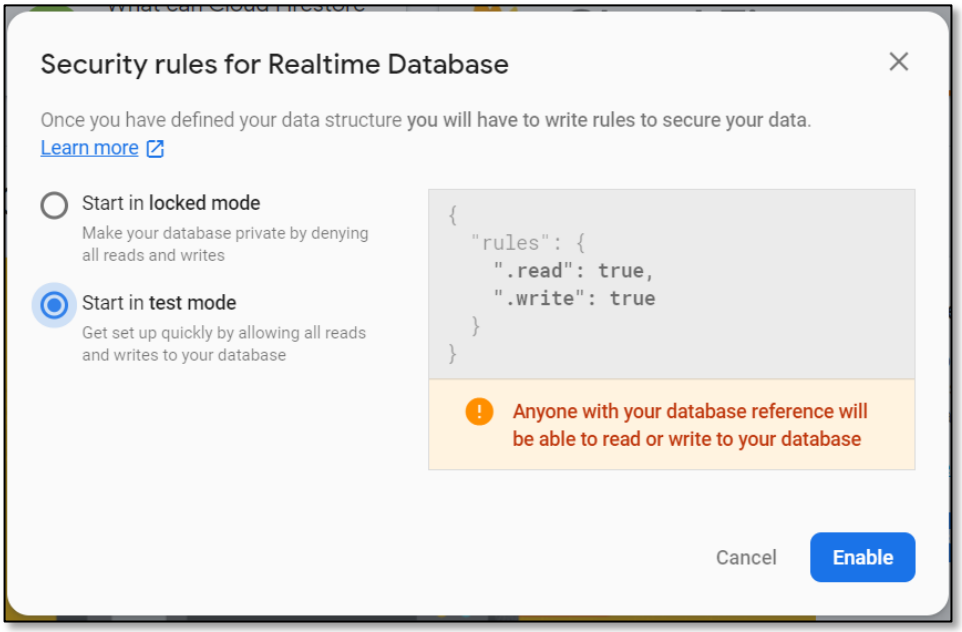


Figure 5.7: Firebase (Setting up database)

Once finish creating, this page is shown.

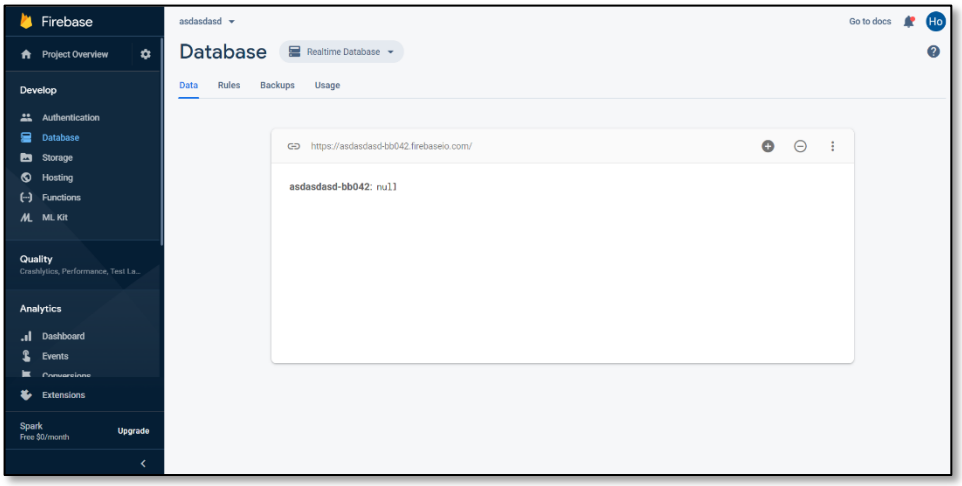


Figure 5.8: Firebase (Database Page)

Go to android studio and click “Tool” and click “Firebase”.

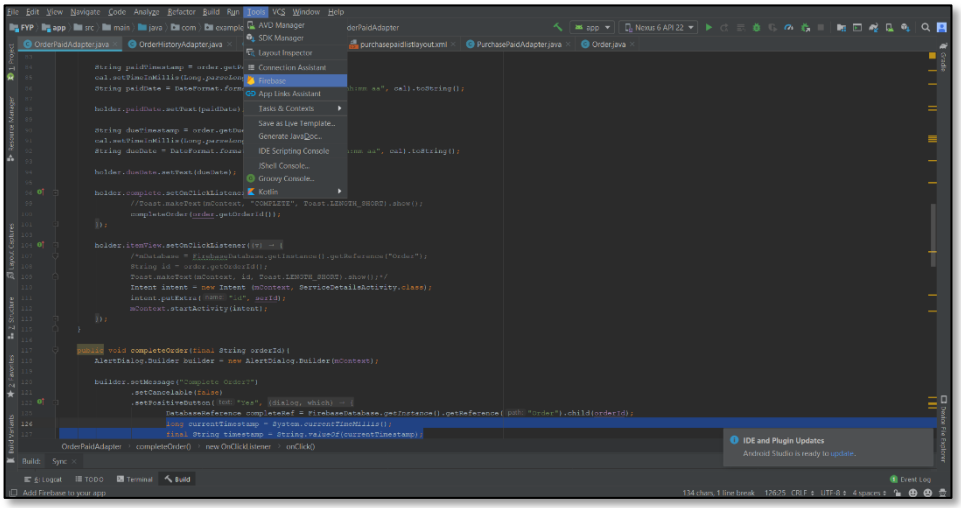


Figure 5.9: Android Studio Overview

After clicking Firebase, this firebase sidebar is shown and expand the Realtime Database. Enable the Realtime Database by signing in with previously created account in Firebase.

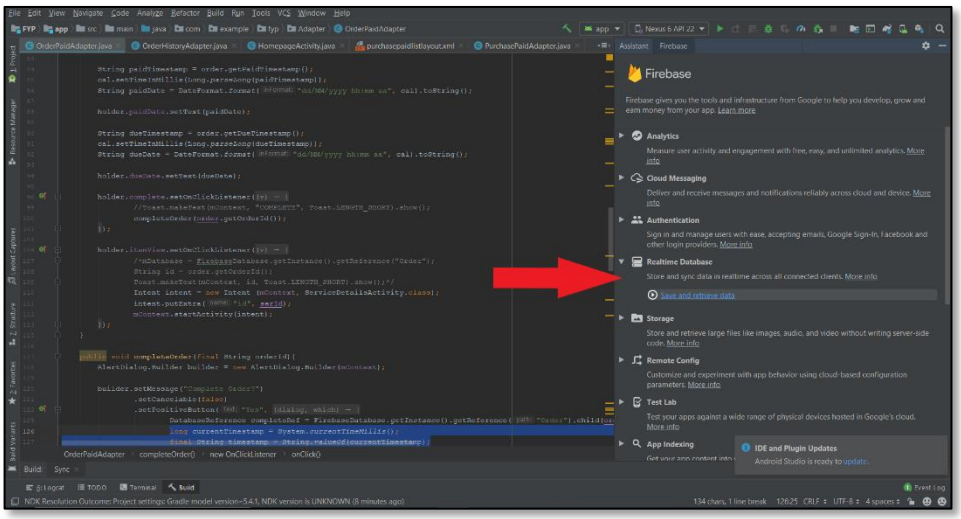


Figure 5.10: Android Studio (Firebase Sidebar)

When all of the setup is completed, Android studio is connected to Firebase and ready to use.

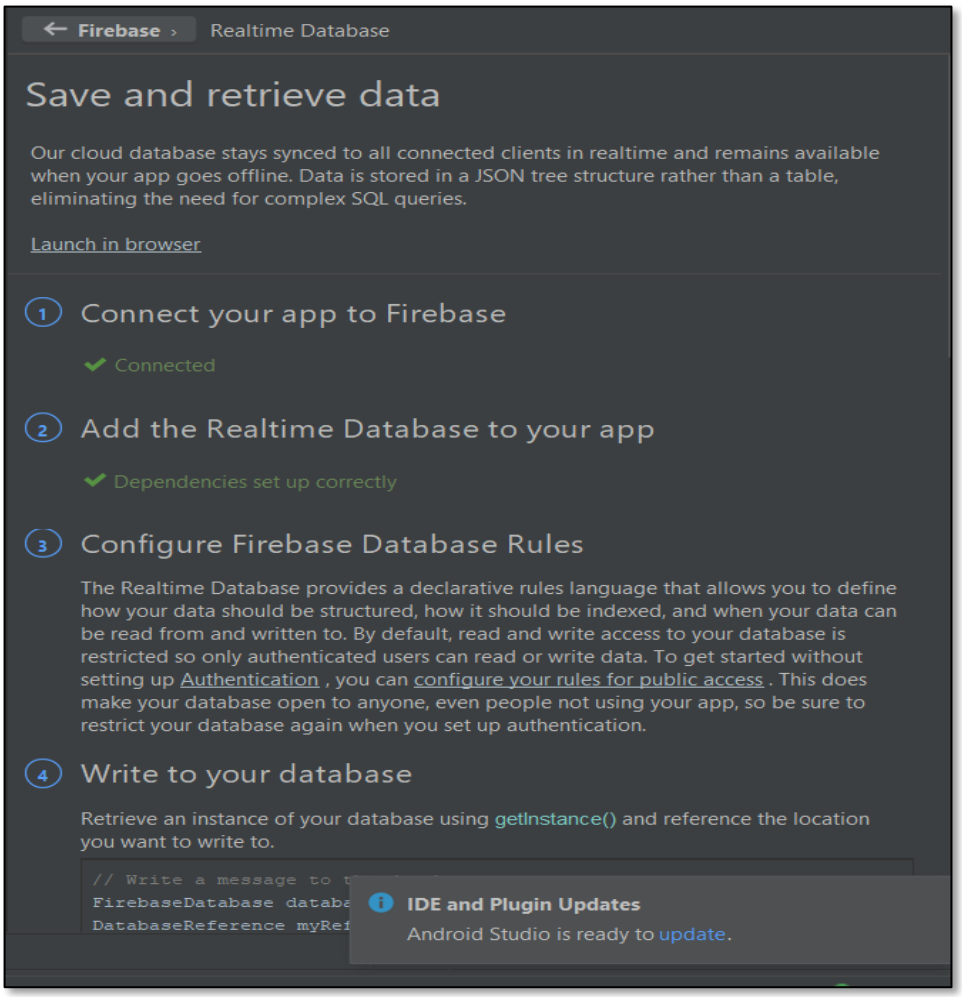


Figure 5.11: Firebase Setup Complete

5.2 Actual System Interface

5.2.1 Mobile Application Interface

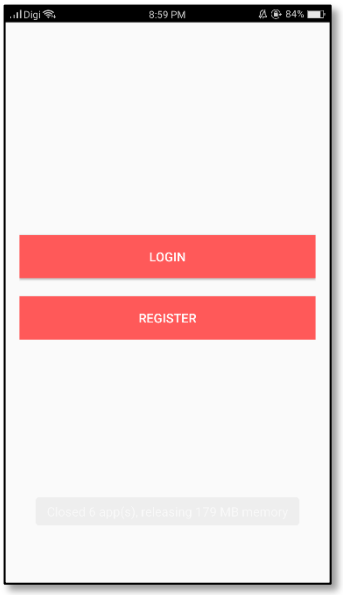


Figure 5.12: Main Page

Figure 5.12 show the main page of the application. User can either choose to login to the application or register as user if does not have any account.

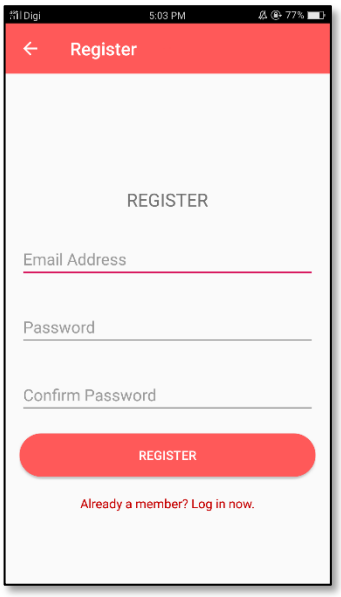


Figure 5.13: Register Page

As shown in Figure 5.13, user can register themselves by providing their e-mail address and password. Once all field is filled in, user can press the register button to register themselves.

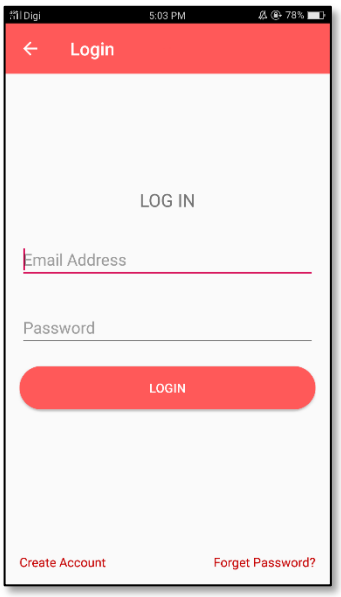


Figure 5.14: Login Page

After successfully register themselves, user have to verify their account through e-mail by clicking the link sent by the application. Once verified, user can login using registered account to the application as shown in Figure 5.14.

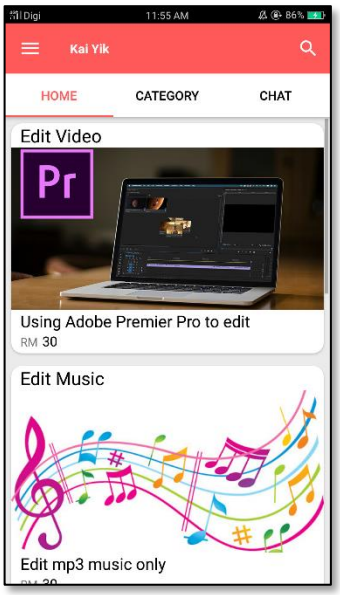


Figure 5.15: Homepage

Figure 5.15 show the first page once the user is logged in, which is the homepage of the application. It shows list of services added by other user for others to buy.

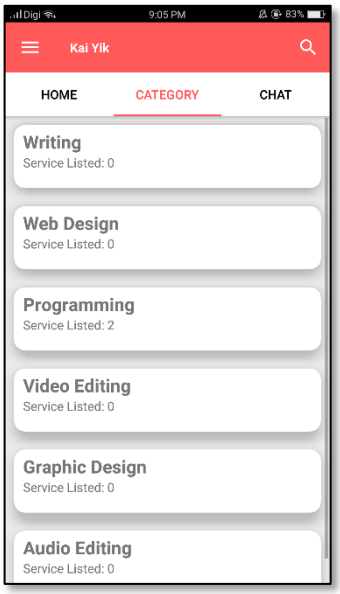
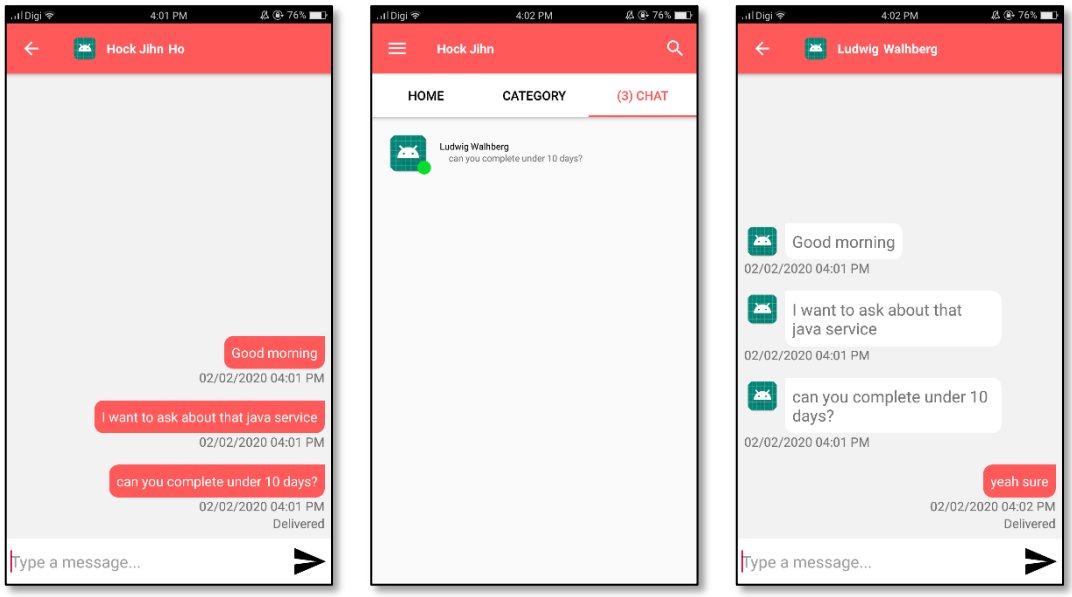


Figure 5.16: Category Page

Figure 5.16 is the category page where all the category is listed here. User can click the category name and it leads to another page where it shows a list of service that is under the category.



(a) **(b)** **(c)**

Figure 5.17 (a-c): Chatting with User

Figure 5.17 shows the chatting features in this application. Figure 5.17 (a) is the perspective from the sender side, Figure 5.17 (b-c) is the perspective of receiver side. It also shows the date and time the message is sent.

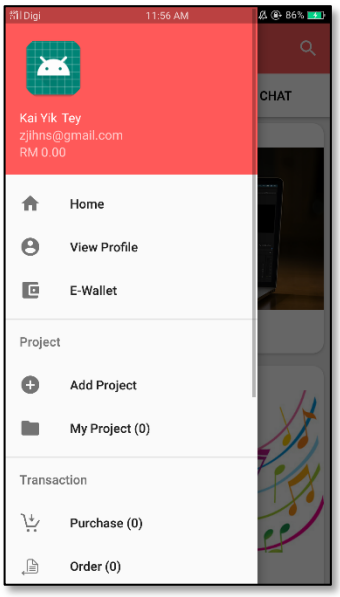


Figure 5.18: Navigation Bar

Figure 5.18 shows the navigation bar of the application. The navigation bar consists of the image of the user as well as information of the user. Besides, user can navigation through the application using this navigation bar.

Figure 5.19: View and Edit Profile Page

Figure 5.19 is the profile page. User can view and edit their own profile information. It shows the name of the users, phone number as well as the address of the users. User can edit their own profile by pressing the pen button at bottom right corner.

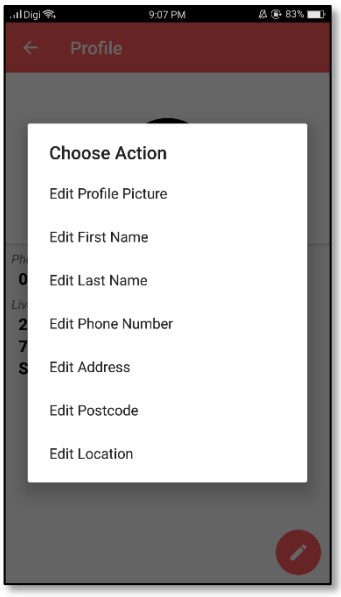


Figure 5.20: Edit Profile Option

When the pen button is pressed, a dialog is popped out as shown in Figure 5.20. User can choose to edit which information, or update their information.

Figure 5.21: Edit Profile Successful

Once successfully edited, the information is update immediately as shown in Figure 5.21.

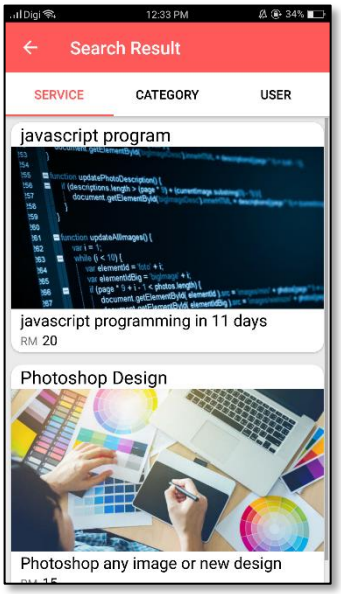


Figure 5.22: Search Result (Service)

Figure 5.22 shows one of the features that included in this application, which is the searching feature. Figure 5.22 shows the result of the search with a list of services.

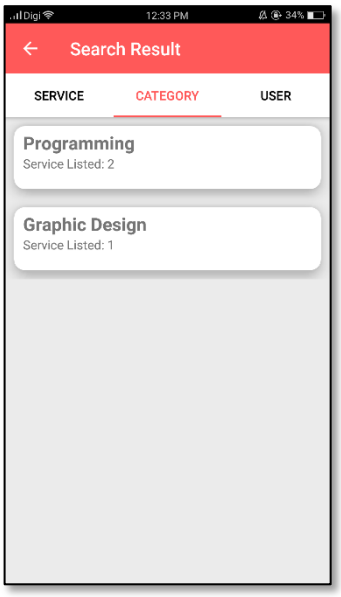


Figure 5.23: Search Result (Category)

Figure 5.23 is the search result for category. When user search specific keyword, the name of the category that consists of the keyword is shown. User can click the category and view list of services that is under the category.

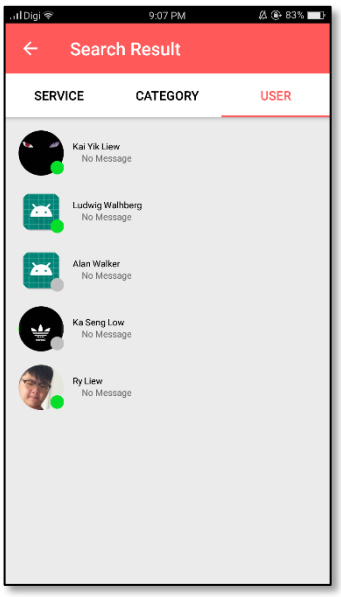


Figure 5.24: Search Result (Users)

Figure 5.24 shows the search result for users. When user search for a keyword, user's first name and last name that consists of the keyword is shown. User can click their profile to view or chat with them.

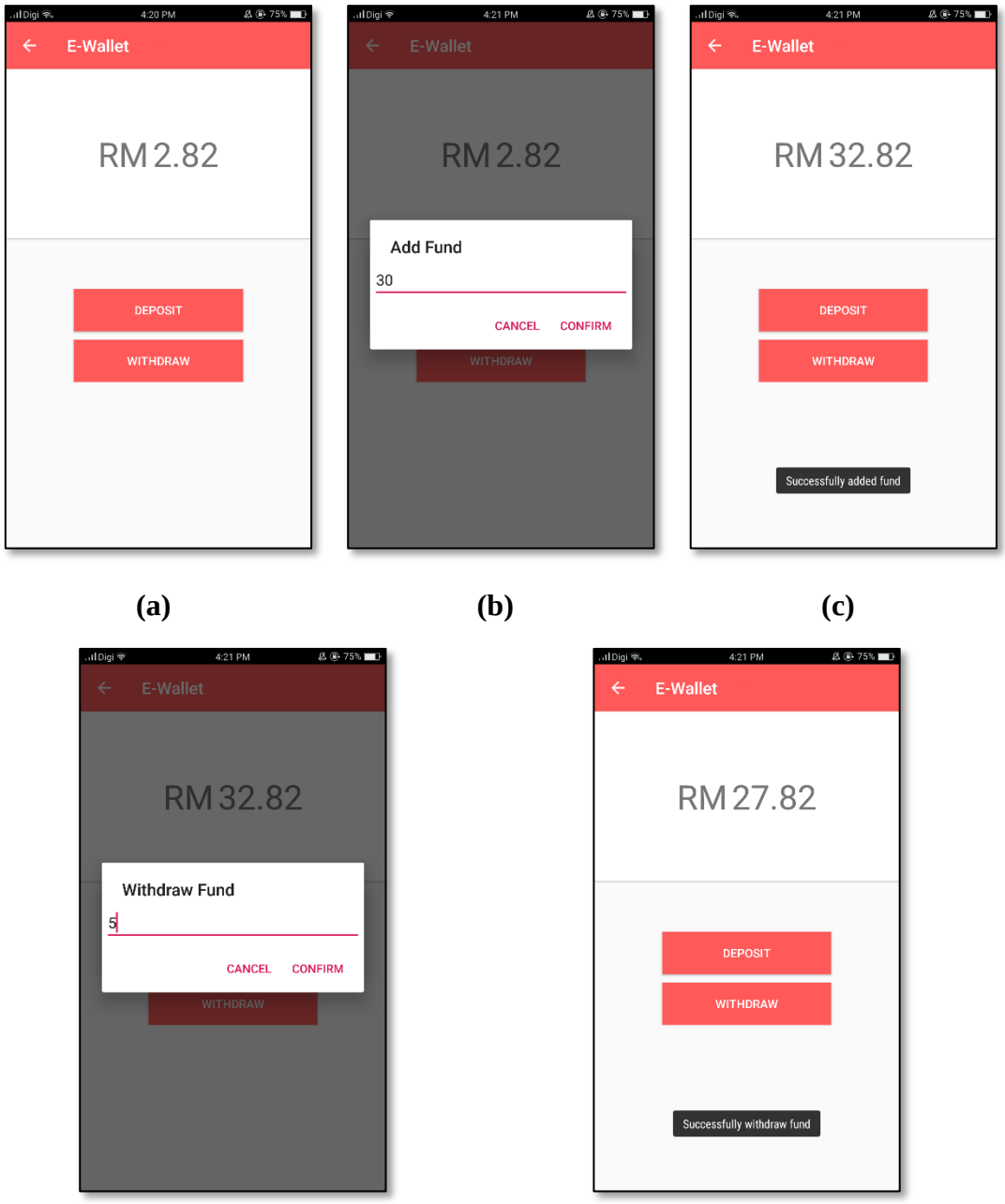
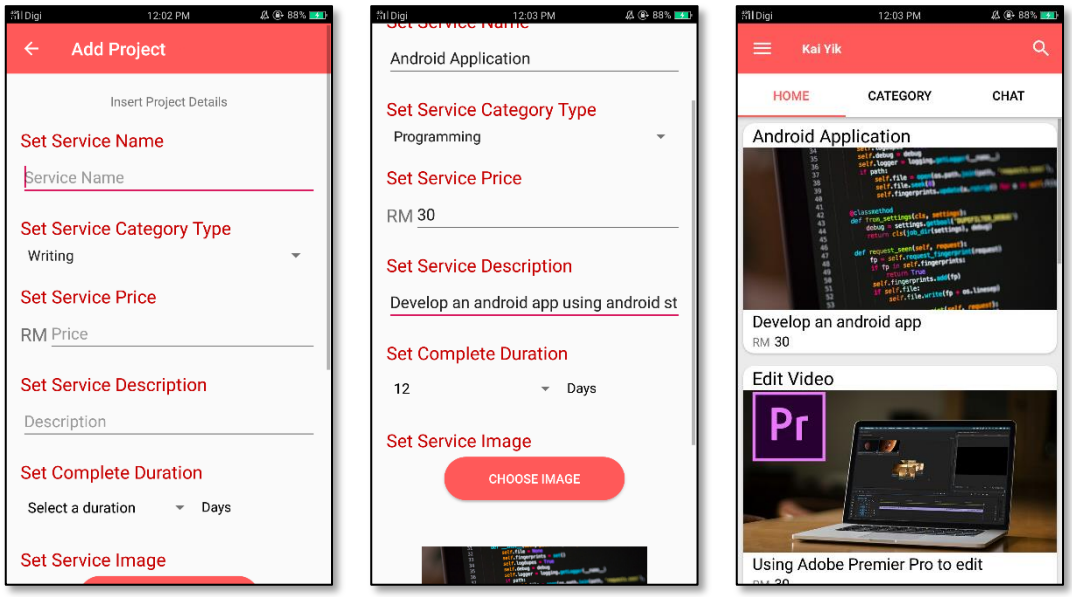


Figure 5.25 (a-e): E-Wallet

Figure 5.25(a-e) shows the flow of deposit and withdrawing fund from the e-wallet. User can view their balance in this page and they can add fund to their e-wallet or withdraw the fund from the e-wallet.



(a) **(b)** **(c)**

Figure 5.26 (a-c): Creating Service

Figure 5.26 shows the flow of creating service. User have to fill in all the details of service such as name, category, price, description, duration to complete and image of service. After user complete all the field and upload the service, it is listed in the homepage for others to view.

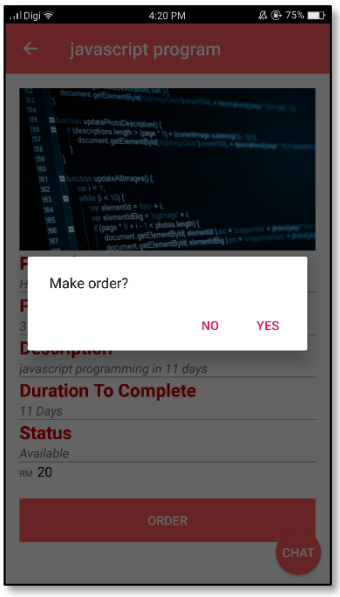


Figure 5.27: Make Order

Figure 5.27 shows a dialog pop up to confirm that user wish to make the order. When user pressed “NO”, the dialog disappears, but if the user pressed “Yes”, the application places the order and lead the user to another page.

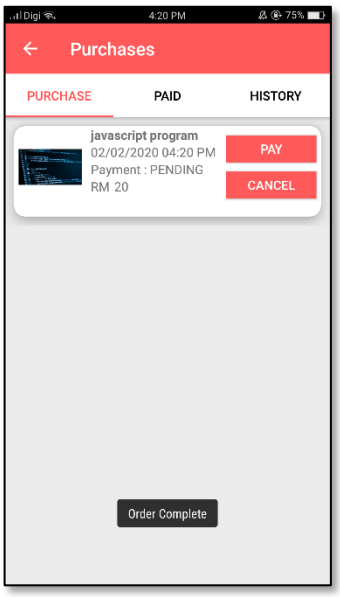


Figure 5.28: Make Order Complete

Figure 5.28 shows the page where the application leads the user to after an order is made. Here, use can pay for the order, the fund is directly deducted from the user’s e-wallet, or user can cancel the order.

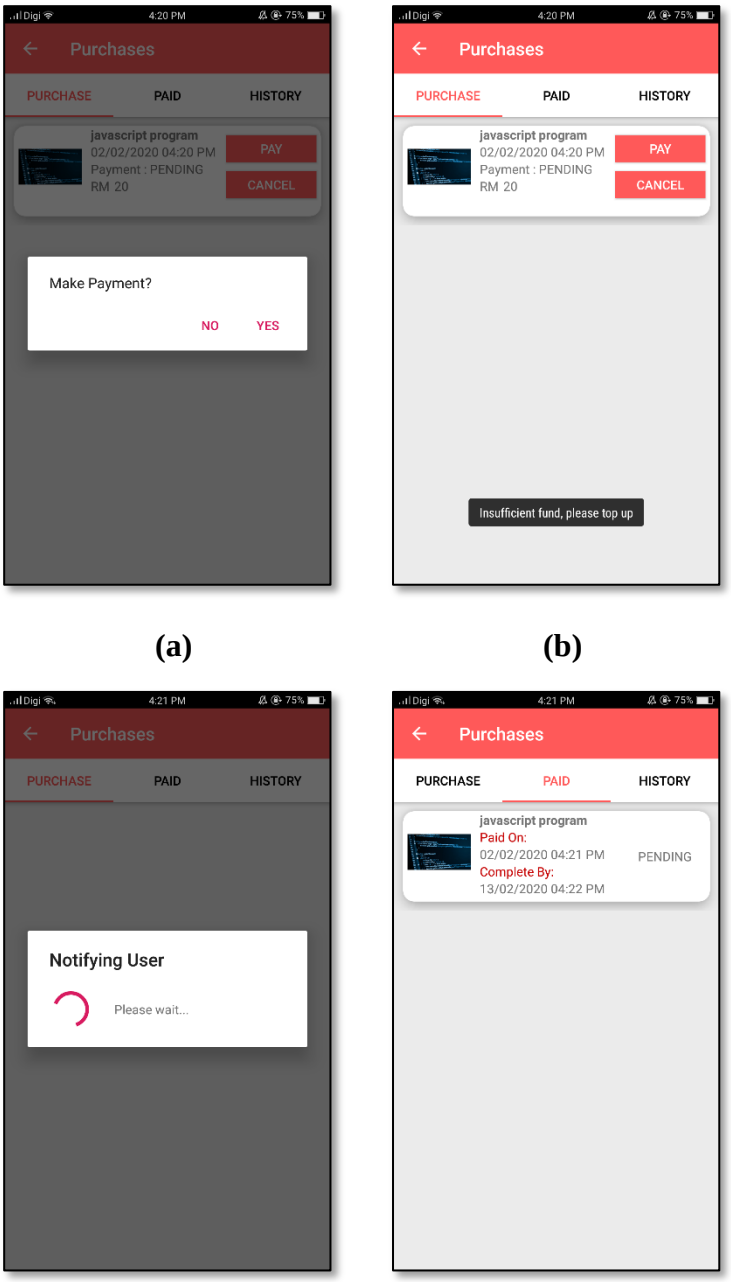
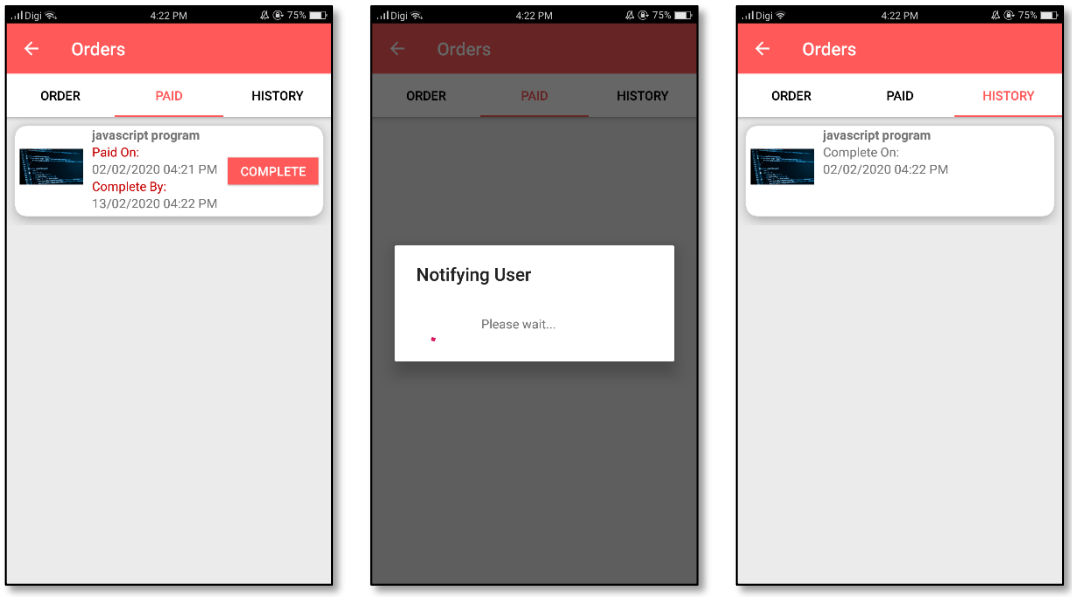


Figure 5.29 (a-d): Paying Order

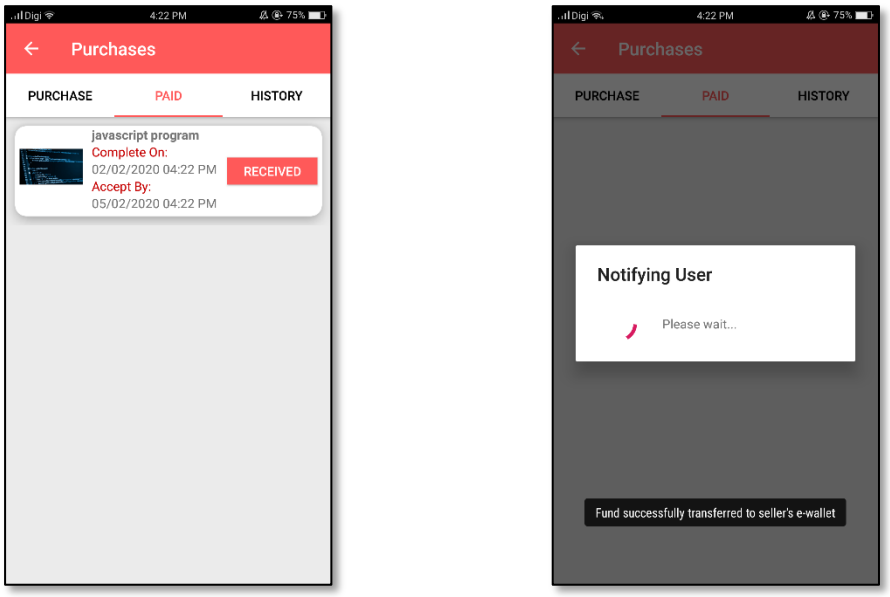
Figure 5.29 (a-b) shows the error message is shown when the user presses the pay button but the e-wallet fund is insufficient. Figure 5.29 (c-d) shows when the user presses the pay button and the fund from the e-wallet is sufficient. An e-mail notification is sent to the seller's e-mail to notify them that an order has been made and paid.



(a) **(b)** **(c)**

Figure 5.30 (a-c): Completing Order (Seller Side)

Figure 5.30 (a-c) shows the flow of the activity when user from the seller side finishes the project and press the "COMPLETE" button. After the "COMPLETE" button is pressed, the application sends an e-mail notification to the buyer's e-mail to notify them to confirm the order so that the fund can be transferred to the seller's e-wallet.



(a) **(b)**

Figure 5.31: (a-b): Confirming Order (Buyer Side)

Figure 5.31 (a-b) shows the buyer side of confirming the order. When the user presses the “RECEIVED” button, an e-mail notification is sent to the seller’s e-mail notify them that the fund has been transferred to their e-wallet.

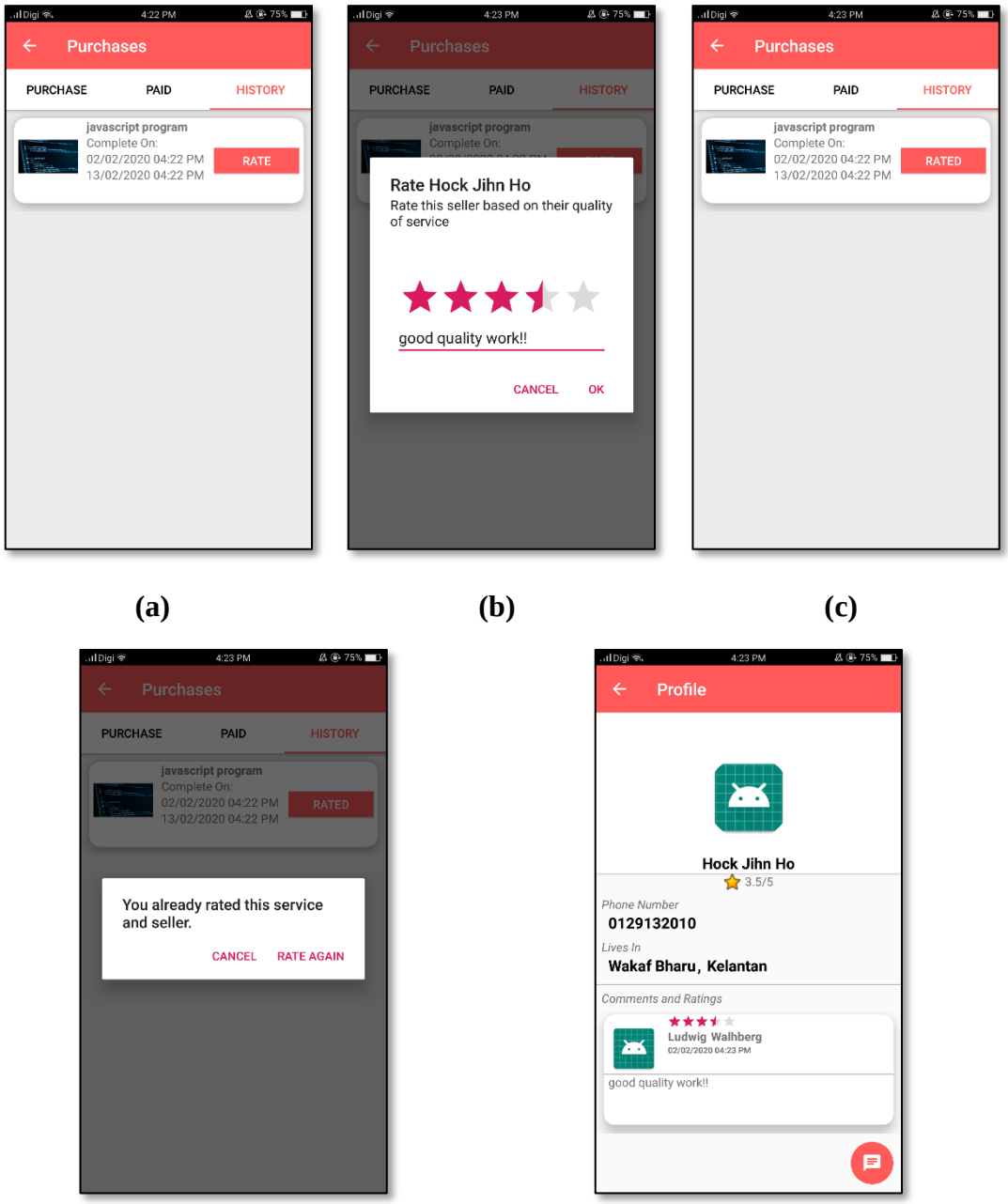
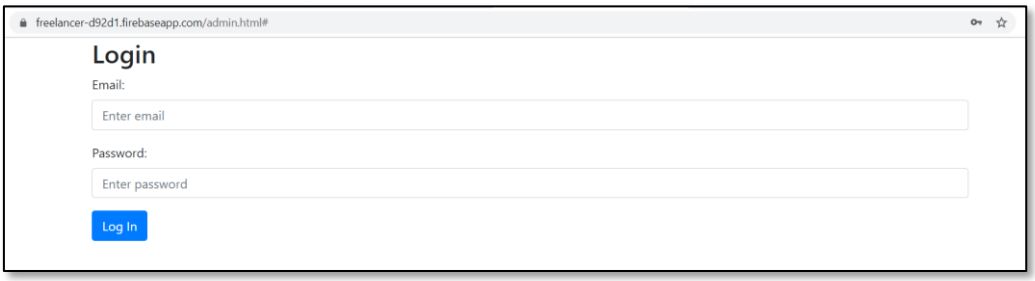


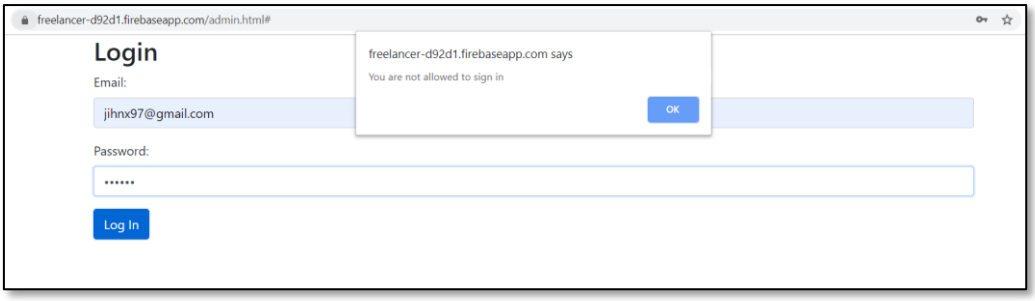
Figure 5.32 (a-e): Rating Seller (Buyer Side)

Figure 5.32 (a-e) shows the flow of activity to rate the seller based on the quality of their service. The rating is shown on the user's profile. Figure 5.32 (a-c) shows when the user presses the "RATE" button. A dialog is shown to allow user to rate the seller using star, and write comment. Figure 5.32 (d) is when the user already rated the seller and presses the "RATED" button again. A dialog will confirm that the user wishes to rate the seller again by modifying previous rating. Figure 5.32 (e) shows the rating and comment shown on the user's profile.

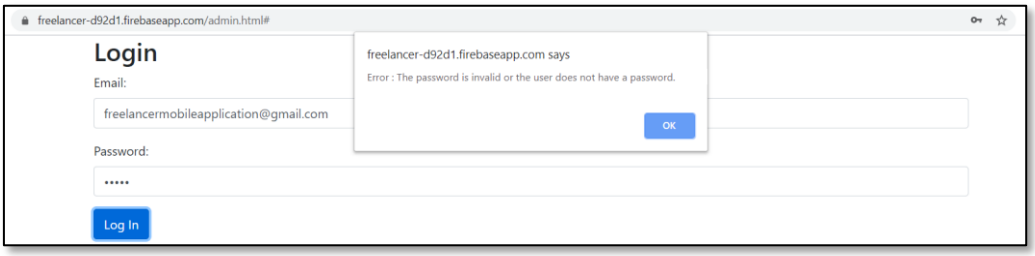
5.2.2 Web Browser Interface (Admin)



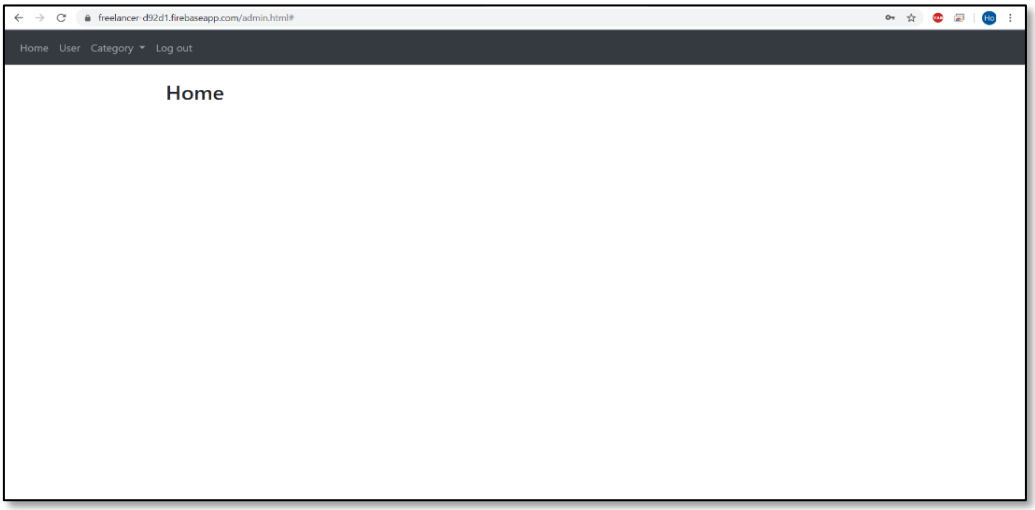
(a)



(b)



(c)



(d)

Figure 5.33 (a-d): Admin Login

Figure 5.33 shows the flow in login page of admin. Figure 5.33 (b) shows that when a user that is not admin tries to login, a message is prompted to let the user know that they are not allow to sign in.

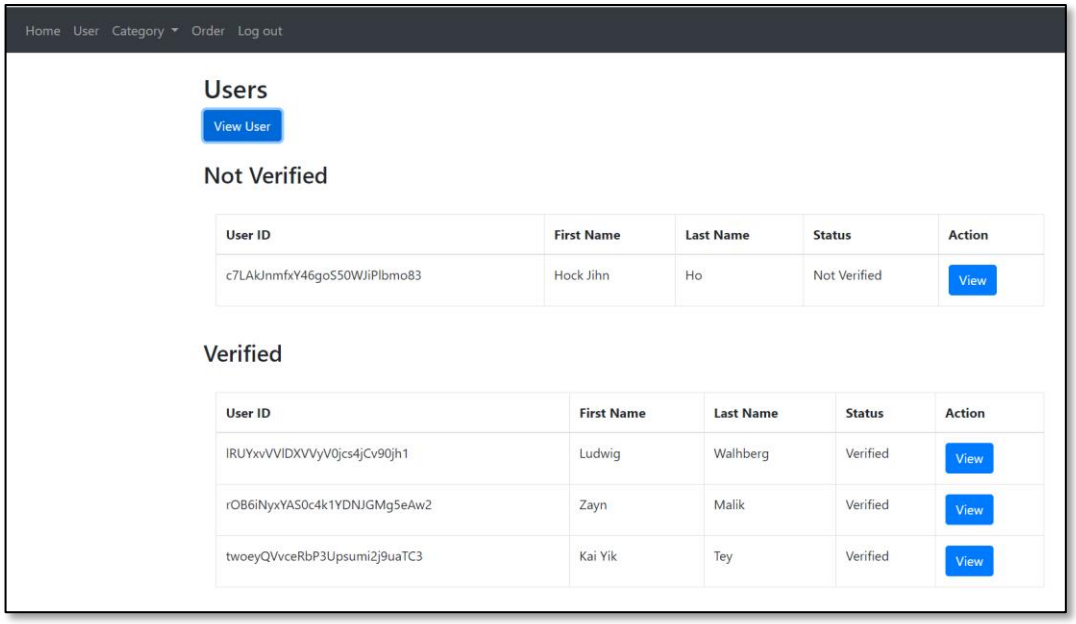
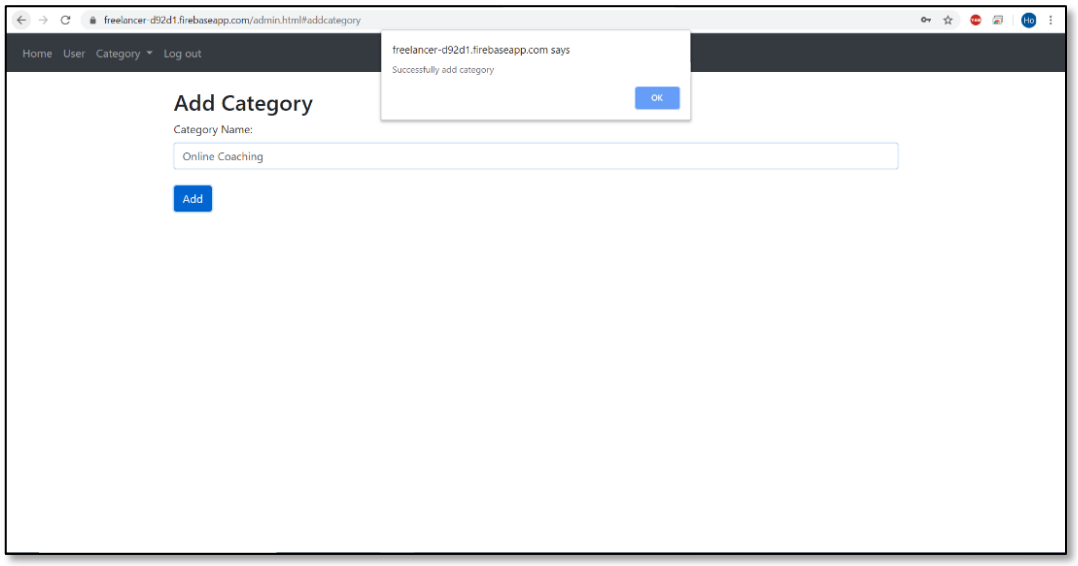
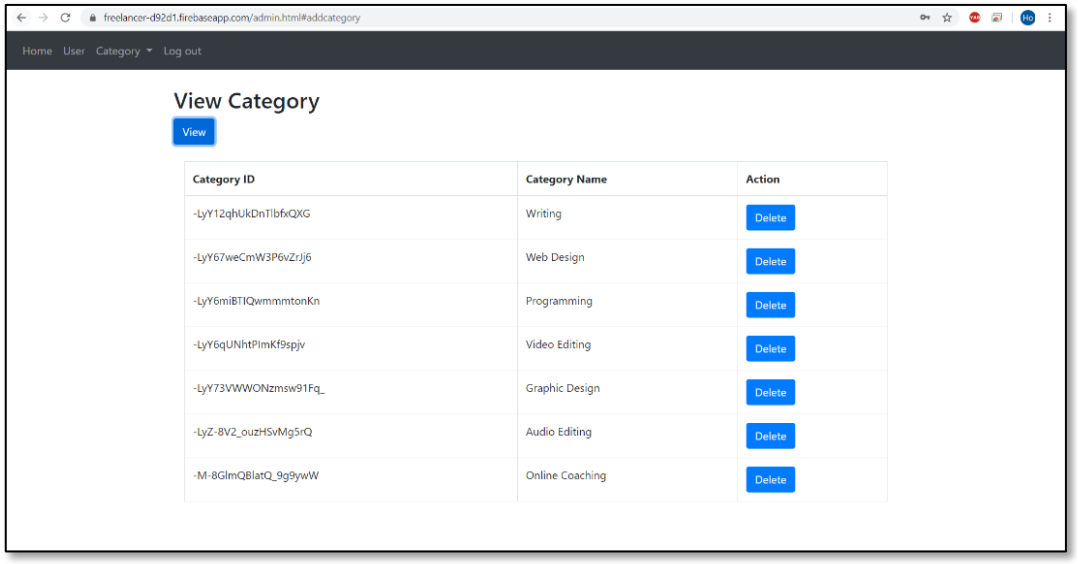


Figure 5.34: User Page

Figure 5.34 shows the user page viewed by admin. Here, admin can view the list of details of the users.



(a)



(b)

Figure 5.35 (a-b): Category Page

Figure 5.35 shows the category page in the admin side. In Figure 5.35 (a), admin can add category by typing in the name of the category in the text field. After adding a category, it goes to Figure 5.35 (b). Here, admin can view a list of categories. Admin also can delete category.

5.3 Summary or Overview Diagram of the Solution

Table 5.1: Summary of Features in System

Name of the System	Freelancer Mobile Application
Platform	User: Android Admin: Website
Key Features (User)	1. Create Service 2. Update Service 3. Search Service 4. View Service 5. Search Category 6. View Category 7. Search User 8. Chat with User 9. Make Order 10. Cancel Order 11. View Profile 12. Update Profile 13. Write Rating 14. View Rating 15. Add Duration of Service 16. Edit Service Availability 17. Email Notification
Key Features (Admin)	1. View Users 2. Verify Users 3. Add Category 4. View Category

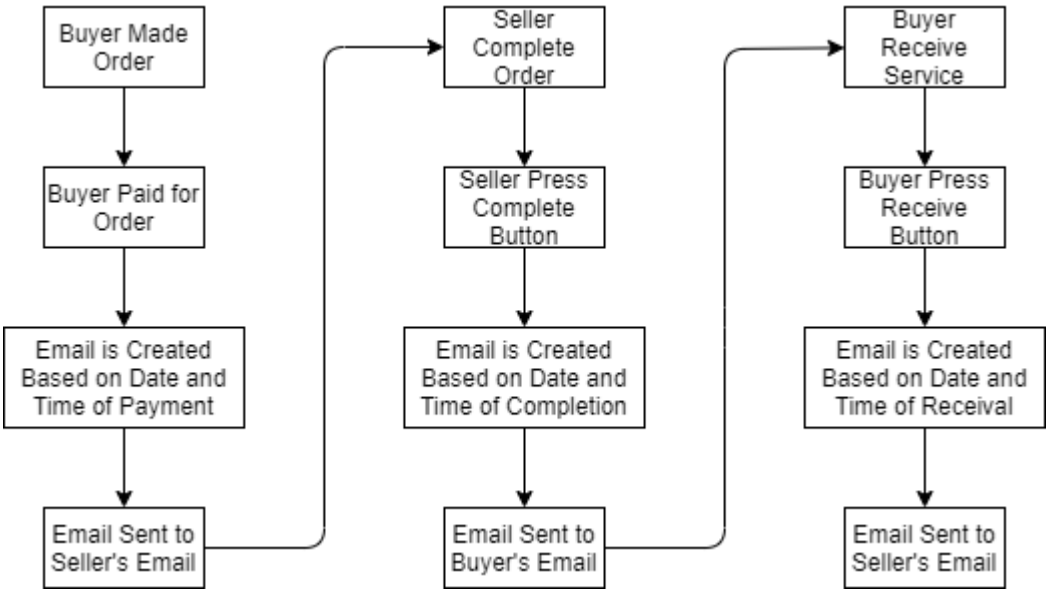


Figure 5.36: Flow of E-Mail Notification

Figure 5.36 shows the flow of e-mail notification start from when the buyer made the order. After the buyer has paid for the order, an e-mail is created and sent to the seller's e-mail. Next, after seller is notified, seller can complete the order by pressing the "COMPLETE" button and an email is created and sent to the buyer. Then, once buyer is satisfy with the service, buyer then press the "RECEIVE" button and an email is created and sent to seller's email to notify them.

CHAPTER 6

THE IMPLEMENTATION PROCESS / RESULTS

6.1 Comparison between Original Goal and Current System

Table 6.1: Objective Fulfilment

No	Objective	Status
1	To design a mobile application that enable freelancer to sell their skill by providing double layers protection on the transaction between the freelancer and the buyer.	Meet
2	To create a digital platform that enables the communication between freelancer and buyer before a service is delivered.	Meet
3	Prepare an admin to monitor the identity of seller to prevent fraud.	Meet

Table 6.2: Comparison between Original Goal and Current System

No	Original Plan	Current System	Reason
1.	Buyer and seller are two different users. One account can either to buy or to sell but not both.	User can either buy or sell in the application using only one account.	It is more convenient to the user because they do not have to create two different account for the same application to do both tasks.
2.	Admin table in database design is included in this project.	Admin table in database design is remove from this project.	Because the responsibility of admin is only to view details of service or user and verify the user using the web. Because of this, admin does not require to store their details in a database.
3.	E-mail notification is not included in the system.	New feature e-mail notification is added to the system.	To allow buyer or seller to be notified through their e-mail.
4.	Availability of service is not included in the system.	New feature availability of service is added to the system.	To allow seller to choose whether to sell the service to other users without deleting the service.
5.	Duration of the service will be completed is not included in the system.	New feature duration of the service will be completed is added to the system.	To allow the buyer to know approximately how long the service will be completed.

6.2 New Features Added in the System

6.2.1 Availability of Service

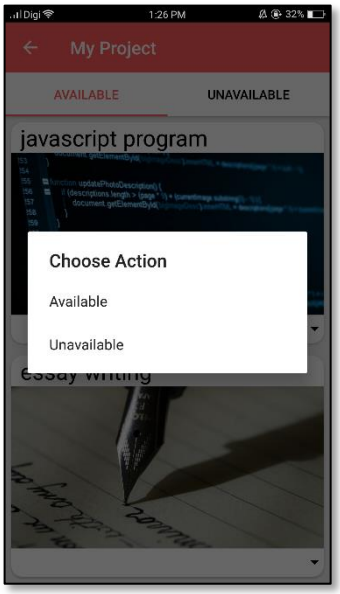


Figure 6.1: Service Availability

Figure 6.2 shows user can set the availability of the service. Initially when the service is added, the availability is set to available. User can set the service to available to order or unavailable to order. If the service is available, other user can view the service and make order. If the user set the service availability to unavailable, the service is shown in the homepage or search result and user can also view the details of the service but unable to make the order.

6.2.2 Email Notification

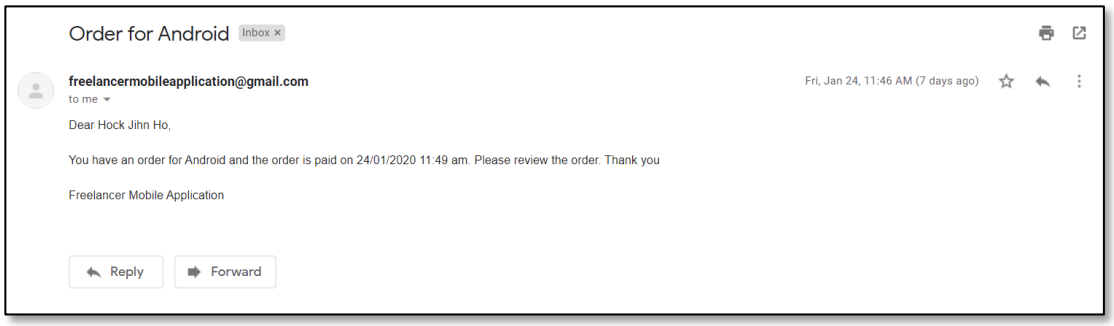
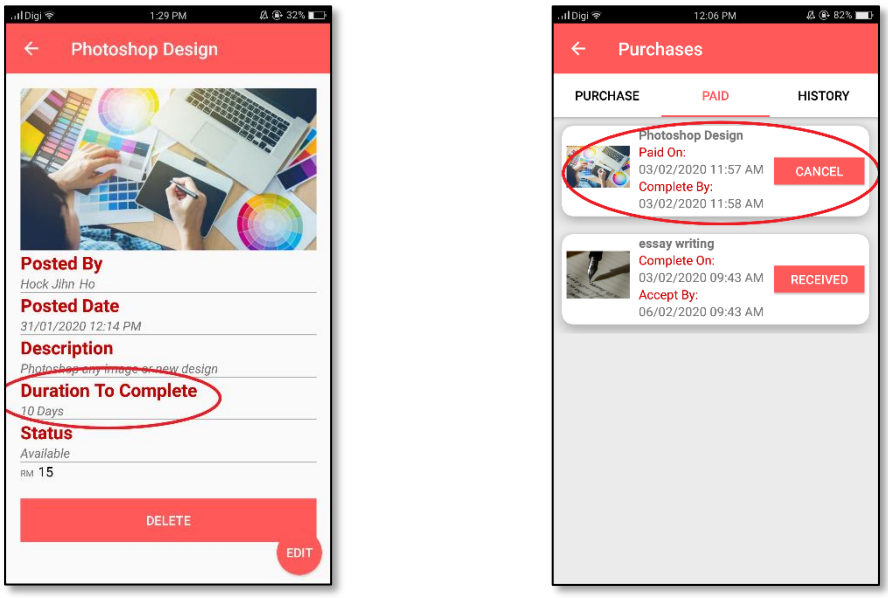


Figure 6.2: Email Notification (Order Paid)

Email notification is a new feature added to the system while developing. The e-mail is sent to the seller's e-mail when a buyer has paid the order. Besides, when the seller has completed the order, once the "COMPLETE" button is pressed, an e-mail is sent to the buyer's e-mail. This e-mail is to notify the buyer that their order is completed and notify them to review and confirm the order. Once the buyer has pressed the "RECEIVED" button, an e-mail is sent to the seller to notify them that the fund has been transferred to their e-wallet. Figure 6.1 shows the sample e-mail content when the buyer has paid for the order.

6.2.3 Duration of Service



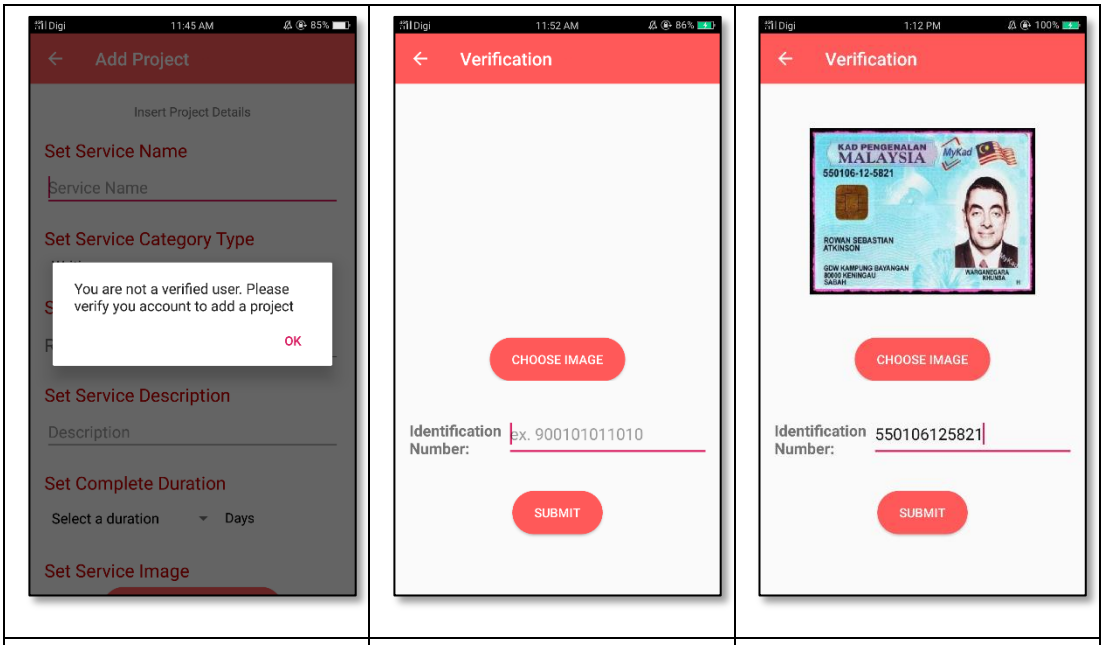
(a) **(b)**

Figure 6.3 (a-b): Service Duration

Figure 6.3 (a) shows the duration of service set by the seller. It is a new feature added to this system. When the user is uploading a project, they can choose how long the project will be in from 1 days to 30 days maximum. This feature is added to this system to allow the buyer to know approximately how long the service will be completed. But if the duration exists the duration set by the seller, buyer has the right to cancel the order. Figure 6.3 (b) shows the "CANCEL" button is shown when the time is due. Although buyer has the right to cancel, seller can communicate with buyer to let them complete their project to refrain them from cancelling the order.

6.3 Extra Features

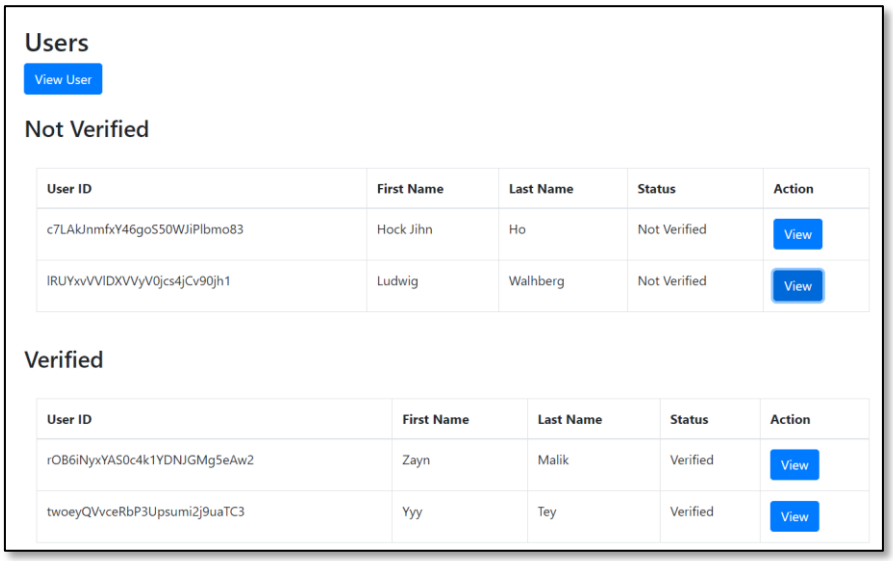
6.3.1 User Verification



(a)	(b)	(c)

Figure 6.4 (a-c): Uploading Identification Details

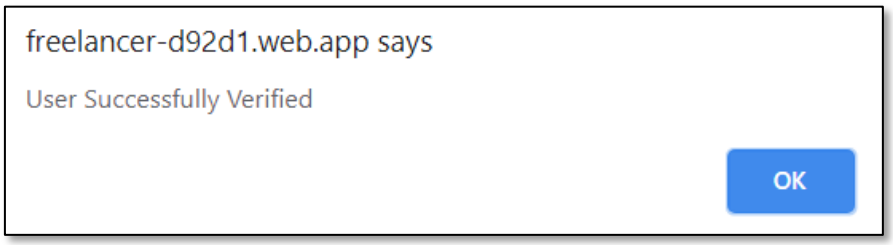
Identity of a person is easily stolen. Anyone can steal other person's identity without their knowledge. To avoid the usage of fake identity in this project, admin have to verify every user registered to this application. Figure 6.4 (a-c) shows that when an unverified user tries to add a service, the application prevents the user to add the project by prompting the user to add their identification card image and number for verification purposes. Only user who intend to sell service are require to verify their account.


Users
View User
Not Verified
User ID
First Name
Last Name
Status
Action
c7LAkJnmfxY46goS50WJiPlbmo83
Hock Jihn
Ho
Not Verified
View
IRUYxvVVIDXVVyV0jcs4jCv90jh1
Ludwig
Walhberg
Not Verified
View
Verified
User ID
First Name
Last Name
Status
Action
rOB6iNyxYAS0c4k1YDNJGMg5eAw2
Zayn
Malik
Verified
View
twoeyQVvceRbP3Upsumi2j9uaTC3
Yyy
Tey
Verified
View

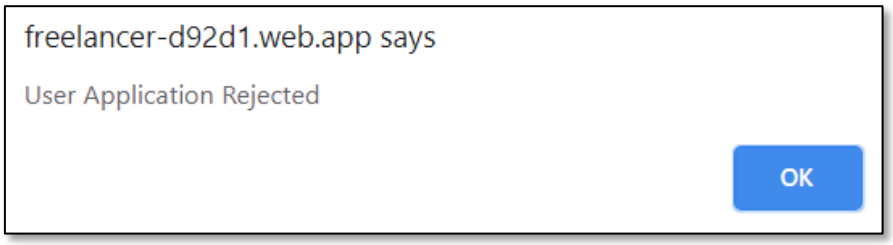
(a)


Ludwig
Walhberg
Phone Number: 0112580369
Address: pout
City: Jitra
State: Kedah
Verify
Reject
IC Image
IC Number
550106125821

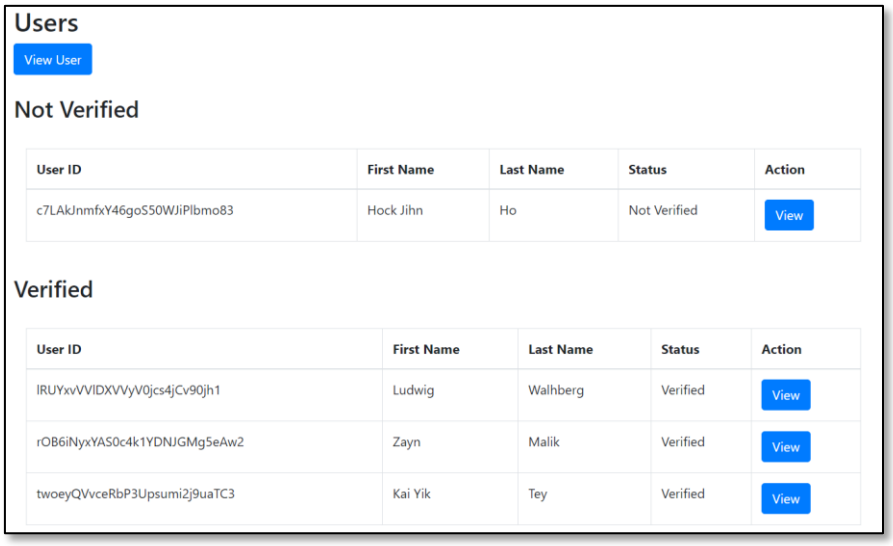
(b)


freelancer-d92d1.web.app says
User Successfully Verified
OK

(c)



(d)



(e)

Figure 6.5 (a-e): Admin Verifying User

Figure 6.5 shows that admin verifying the user. In Figure 6.5 (a) Admin view a list of verified and unverified users. When admin click the "View" button, the information of user is listed out as shown in Figure 6.5 (b). Here, admin can verify whether the Identification Number provide by the user are match with the Identification Card image. If match, admin can click the "Verify" button. After verify, an alert is shown as in Figure 6.5 (c) to alert the admin. If Admin reject the application, an alert is shown to Admin as shown in Figure 6.5 (d). Figure 6.5 (e) shows the list of tables after refreshing the table.

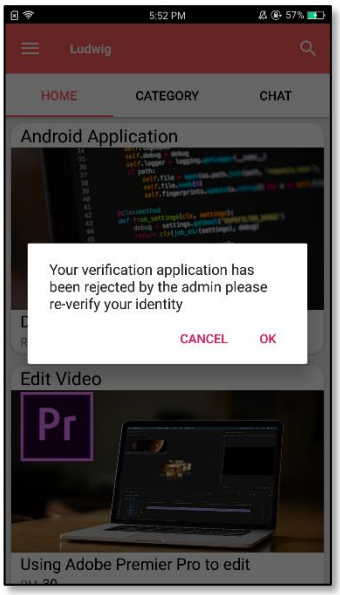


Figure 6.6: Verification Application Rejected

Figure 6.6 shows after the admin reject the application of the user, when user access the mobile application, a notification is shown to notify user that their application is rejected and prompt the user to resubmit their verification details or they can dismiss the alert.

CHAPTER 7

TESTING / EVALUATION OF FINDINGS

7.1 Print Screen of Errors and Solved Interface from the System

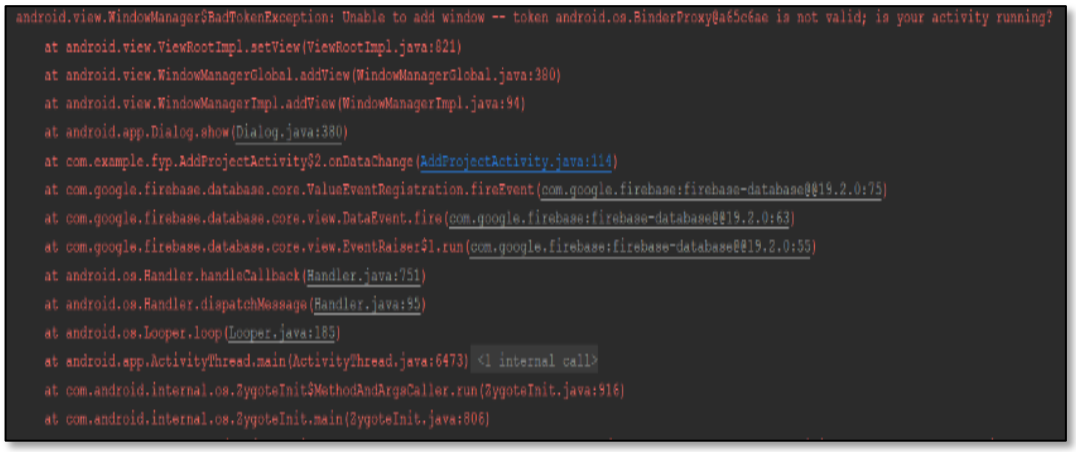


Figure 7.1: Error Log 1

Figure 7.1 shows the error caused by not checking whether the activity is running. To solve this problem, before running a code, check whether the activity is running by using *if(!((Activity) context).isFinishing())*. Write the code inside the if statement, only the error does not occur.

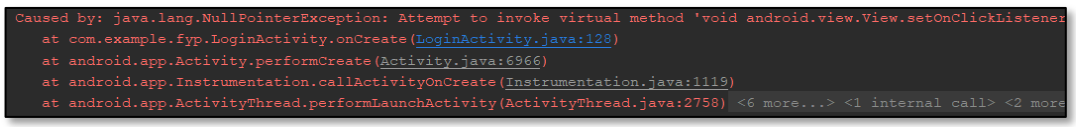


Figure 7.2: Error Log 2

Figure 7.2 shows an error caused by not properly initializing and defining the variable. To solve this, define the variable in the suitable function so that it does not return a null value.

7.2 Test Script

Table 7.1: Test Conducted for All the Function

Test ID	Test Case	Expected Result	Actual Result	Status
1	Login: Fill in an invalid username and password to login	User should be unable to login into the system.	User is unable to login into the system.	**Pass**
2	Register: Fill in e-mail address and password to register as user.	User should be able to register.	User is able to register.	**Pass**
3.	Insert Details: Fill in personal information to store for later user.	User should be able to fill in their personal information after registration.	User is able to fill in their personal information after registration.	**Pass**
4.	View Profile: View the account information	User should be able to view their own profile and other user's profile	User is able to view their own profile and other user's profile	**Pass**
5.	Edit Profile: Change personal information of account.	User should be able to edit their own profile.	User is able to edit their own profile.	**Pass**
6.	Session Access: Cannot direct access application without login.	User should be able to use the application after login.	User is able to use the application after login.	**Pass**
7.	Manage E-Wallet: Deposit and withdraw fund from e-wallet	User should be able to deposit and withdraw fund from e-wallet.	User is able to deposit and withdraw fund from e-wallet.	**Pass**

8.	Add Services: Add service into the system.	User should be able to add services into the system.	User is able to add services into the system.	**Pass**
9	Manage Services: Manage own services in the system.	User should be able to edit, delete and set availability of their services.	User is able to edit, delete and set availability of their services.	**Pass**
10	View Category: View added category in the system.	User should be able to view various category and services for specific category.	User is able to view various category and services for specific category.	**Pass**
11	Chat: Chat with another user.	User should be able to chat with another user via instant messaging.	User is able to chat with another user via instant messaging.	**Pass**
12	Search: Search for service, category and user.	User should be able to search for services, categories and users in the system.	User is able to search for services, categories and users in the system.	**Pass**
13	Make Order: Make order for services.	User should be able to place order for services.	User is able to place order for services.	**Pass**
14	Make Payment: Pay for ordered services.	User should be able to pay the total amount for ordered services.	User is able to pay the total amount for ordered services.	**Pass**
15	Cancel Order: Cancel services before payment is made or after payment is made but	User should be able to cancel the services before the payment is made or after payment is made but	User is able to cancel the services before the payment is made or after payment is made but seller does	**Pass**

	completion date is due.	seller does not complete the service in time.	not complete the service in time.	
16	<u>Give Review:</u> Give rating and comment for services done by the seller.	User(buyer) should be able to give rating and write comment after seller has complete the service.	User(buyer) is able to give rating and write comment after seller has complete the service.	**Pass**

CHAPTER 8

CONCLUSION

As a conclusion, it is no doubt that freelancing has a flexible working hour, freelancer able to choose whether to accept or reject a project. They can be their own boss and work whenever they want. However, without a proper platform for them to use, this would be almost an impossible to think of. This project not only able to help freelancer to have a platform to advertise themselves and earn some side income, they can use this application to enhance their skills that may be useful to them in the future. Besides that, this project can allow buyer and seller to communicate through digital platform to have a better understanding on the project they working on. Apart from that, this project also can bring benefit to buyer as they have a reliable platform to look for expert. The function that included in this project able to help both the buyer and the seller to utilize this application.

To create a better freelancer application, many existing applications are evaluated and research is carried out to find and list out potential flaws in current freelancer related application. Through this research, data acquired has produced a valuable information to refrain from making same the flaws as present application and help to decide whether new implementation design satisfy user requirements for a user-friendly freelancer mobile application interface.

In a nut shell, after completing this project, there is no certainty that this project would be a perfect mobile application. Developing and perfecting a project is not an easy task because it required a large amount of time to achieve it. It is a great pleasure that someday this application able to distribute to people through Google PlayStore to help people throughout the world.

REFERENCES

Andor (2018). *History of Freelancing* Retrieved 29 July, 2019 from https://www.rent-acoder.com/blog/history-of-freelancing/29

Android Studio (2014). *Android Studio logo* 1 August, 2019 from https://dwglogo.com/android-studio/

Elaine Profeldt (2015). *Elance-oDesk Becomes 'Upwork' In Push To Build $10B In Freelancer Revenues* Retrieved 1 August, 2019 from https://www.forbes.com/sites/elainepofeldt/2015/05/05/elance-odesk-becomes-upwork-today-odesk-brand-gets-phased-out/#9ebfa8251f5e

Feehour – Android Apps on Google Play. (Feehour). Retrieved 18 July, 2019 from https://play.google.com/store/apps/details?id=com.Feehour.Feehour&hl=en

Feehour (2017). Feehour. Retrieved 18 July, 2019 from https://www.appbrain.com/app/feehour-freelance-services/com.Feehour.Feehour

Feehour (n.d.). About Us Retrieved 1 August, 2019 from https://www.linkedin.com/company/feehour

Fiverr (2014). Fiverr. Retrieved 18 July, 2019 from https://www.appbrain.com/app/fiverr-freelance-services/com.fiverr.fiverr

Fiverr – Android Apps on Google Play. (Fiverr). Retrieved 18 July, 2019 from https://play.google.com/store/apps/details?id=com.fiverr.fiverr&hl=en

Fiverr (2010). Fiverr's Term of Service Retrieved 1 August, 2019 from https://www.fiverr.com/terms_of_service

Freelancer.com – Android Apps on Google Play. (Freelancer.com). Retrieved 18 July, 2019 from https://play.google.com/store/apps/details?id=com.freelancer.android.messenger&hl=en

Freelancer.com (2014). Freelancer.com. Retrieved 18 July, 2019 from https://www.appbrain.com/app/freelancer%3A-experts-from-programming-to-photoshop/com.freelancer.android.messenger

Freelancer.com (2009). Overview Retrieved 1 August, 2019 from https://www.freelancer.com/about

GIMP (n.d.). Gimp Retrieved 4 August, 2019 from https://www.gimp.org

Justinmind (2015). Justinmind Retrieved 1 September 2019 from https://www.justinmind.com/support/start-prototyping-web-and-mobile-apps/

Margaret Rouse (n.d.). *Java* Retrieved 4 August, 2019 from https://www.theserverside.com/definition/Java

Merriam-Webster (n.d.) The Surprising History of 'Freelance' Retrieved 29 July, 2019 from https://www.merriam-webster.com/words-at-play/freelance-origin-meaning

PeoplePerHour – Android Apps on Google Play. (PeoplePerHour). Retrieved 19 July, 2019 from https://play.google.com/store/apps/details?id=com.pph&hl=en

PeoplePerHour (2013). PeoplePerHour. Retrieved 18 July, 2019 from https://www.appbrain.com/app/peopleperhour/com.pph

PeoplePerHour (2007). About Retrieved 1 August, 2019 from https://www.peopleperhour.com/about

PeoplePerHour (2007). Team Retrieved 1 August, 2019 from https://www.peopleperhour.com/team/

Rouse (n.d.) *Java* Retrieved 1 August, 2019 from https://www.theserverside.com/definition/Java

Smartsheet (n.d.). Retrieved 3 August, 2019 from https://www.smartsheet.com/system-development-life-cycle-guide

Tomas Laurinavicius (2016). *The History of the Freelance Evolution* Retrieved 29 July, 2019 from https://www.vandelaydesign.com/freelance-defintion/

Technopedia (n.d.). Retrieved 3 August, 2019 from https://www.techopedia.com/definition/22193/software-development-life-cycle-sdlc

Upwork – Android Apps on Google Play. (Upwork Global Inc.). Retrieved 18 July, 2019 from https://play.google.com/store/apps/details?id=com.upwork.android.apps.main&hl=en

Upwork Global Inc. (2015). Upwork. Retrieved 18 July, 2019 from https://www.appbrain.com/app/upwork-for-freelancers/com.upwork.android.apps.main

W3schools (n.d.). Java Introduction Retrieved 4 August, 2019 from https://www.w3schools.com/java/java_intro.asp

APPENDICES

Appendix A: Checklist for FYP Final Report Submission

Appendix B: CD

Appendix C: CD