

Car Pooling Application for University Students

BY

CHANG LI HAO

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

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

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2022

REPORT STATUS DECLARATION FORM

Title: CAR POOLING APPLICATION FOR UNIVERSITY STUDENTS

Academic Session: JUNE 2022

I CHANG LI HAO

(CAPITAL LETTER)

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

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

Verified by,

Chang Li Hao

(Author's signature)



(Supervisor's signature)

Address:

294 Jalan Serunai 6,

Taman Sri Kulai Baru 2,

81000 Kulai, Johor.

Tan Chiang Kang

Supervisor's name

Date: 5/9/2022

Date: 9/9/2022

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

FACULTY/INSTITUTE* OF Information and Communication Technology

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 8/9/2022

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that **Chang Li Hao** (ID No: **19ACB05664**)
has completed this final year project/ dissertation/ thesis* entitled “ **Car Pooling Application
for University Students** ” under the supervision of **Mr Tan Chiang Kang**
(Supervisor) from the Department of **Computer Science** , Faculty/Institute* of
Information and Communication Technology , and **—** (Co-Supervisor)* from the
Department of _____, Faculty/Institute* of _____.

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

Yours truly,

Chang Li Hao
(Student Name)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled “**Car Pooling Application for University Students**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Chang Li Hao

Signature : _____

Name : _____ Chang Li Hao _____

Date : _____ 8/9/2022 _____

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Professor Tan Chiang Kang, who has given me this bright opportunity to engage in this car-pooling application for university students project. It is my first time creating a proper application, and along the journey I felt helpful with the help and support given by my supervisor. His invaluable advice, generous support, encouragement, patience, and useful comments have helped me to improve my project during its development. A million thanks to you.

Special thanks to my university friends, who have also supported me during my development and provide me with true comments and suggestions whenever I am facing doubt and problems.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

The concept of car-pooling has been around in the market for a while, and it has become a more practical and cost-effective way to travel nowadays due to the advancement of technology used. This project will present a new mobile application solution for the development of car-pooling applications for university students. The project will provide students with an alternative way to commute during their daily routine. We also hope to share the benefits of carpooling through this project with numerous university students. The special features for this project will be that users need to be verified to use the carpool service, and the admin will review the student ID uploaded and decide which user needs to have a verified account. The application will be developed using the React Native framework, as NPM includes many ready-to-use libraries that enable a faster development process. It will be an online application, which always requires an internet connection to receive or send data to the cloud database. The main delivery platform for this project will be the Android platform, as it has more users and allows more flexibility in application selection. In terms of functionality, the application will provide several main features such as user authentication, managing user profiles, posting ride requests, checking requested rides, and so on. The name of this application will be DinoPool, with a cute app icon to attract user attention at the same time as providing more simplicity on the user interface. The user will be further categorised into two different roles, which are: rider or driver. Both roles will have the same interface but slightly different functionality. A rider can request to ride along on the driver's route, while the driver can request to pick up the rider at their origin place. The application will provide three carpool services, which are immediate, scheduled, and regular; each service is different in terms of time request and methods. Although single-occupant travel remains popular in Malaysia, we hope to promote the concept of ride sharing to the millennials, who are university students who are already immersed in technology practically and are more comfortable interacting with technology.

Table of Contents

| | |
|---|------------|
| TITLE PAGE | i |
| REPORT STATUS DECLARATION FORM | ii |
| FYP THESIS SUBMISSION FORM | iii |
| DECLARATION OF ORIGINALITY | iv |
| ACKNOWLEDGEMENTS | v |
| ABSTRACT | vi |
| TABLE OF CONTENTS | vii |
| LIST OF FIGURES | x |
| LIST OF TABLES | xii |
| | |
| CHAPTER 1 | 1 |
| 1.1 Problem Statement and Motivation | 1 |
| 1.1.1 Problem Statement on Geographical Issues..... | 1 |
| 1.1.2 Problem statement on Current Carpool Methods..... | 3 |
| 1.1.3 Motivation..... | 6 |
| 1.2 Project Objectives | 7 |
| 1.3 Project Scope and Direction..... | 10 |
| 1.4 Contributions..... | 12 |
| 1.5 Background Information | 13 |
| 1.5.1 The History of Carpooling | 13 |
| 1.5.2 Types of Carpooling | 14 |
| 1.6 Report Organization..... | 15 |
| | |
| CHAPTER 2 | 16 |
| 2.1 Review on Existing Application | 16 |
| 2.1.1 Uber..... | 16 |
| 2.1.2 LYFT..... | 19 |
| 2.1.3 VIA | 22 |
| 2.1.4 Ryde | 24 |
| 2.1.5 Summary of Existing System..... | 27 |
| 2.2 Review on Carpooling Solutions & Methods | 28 |
| 2.2.1 Dynamic Carpooling | 28 |
| 2.2.2 Informal Carpooling..... | 29 |
| 2.2.3 Flexible Carpooling | 30 |

| | | |
|------------------------|---|-----------|
| 2.3 | Review on the Technologies | 31 |
| 2.3.1 | Hardware Platform | 31 |
| 2.3.2 | Cloud Database | 32 |
| 2.3.3 | Development Framework..... | 33 |
| CHAPTER 3 | | 34 |
| 3.1 | Project Planning Methods | 34 |
| 3.2 | Design Specifications..... | 35 |
| 3.2.1 | Project Description..... | 35 |
| 3.3 | Project Plan | 37 |
| 3.3.1 | Work Breakdown Structure | 37 |
| 3.3.2 | Timeline | 38 |
| CHAPTER 4 | | 39 |
| 4.1 | System Design Diagram | 39 |
| 4.1.1 | System Architecture Diagram..... | 39 |
| 4.1.2 | Use Case Diagram..... | 40 |
| 4.1.3 | Use Case Description | 41 |
| 4.1.3 | Activity Diagram | 48 |
| 4.1.4 | System Flow Chart..... | 59 |
| 4.2 | Technologies Specification | 60 |
| 4.2.1 | Framework | 60 |
| 4.2.2 | Hardware Specification..... | 64 |
| CHAPTER 5 | | 65 |
| 5.1 | Hardware Setup..... | 65 |
| 5.1.1 | Desktop Debugging | 65 |
| 5.1.2 | Physical Android Device Debugging..... | 68 |
| 5.2 | Software Setup | 70 |
| 5.2.1 | React Native environment setup on window machine..... | 70 |
| 5.2.2 | Export release APKs and bundles..... | 70 |
| 5.2.3 | Published to Huawei App Gallery | 72 |
| 5.3 | System Operation (with Screenshot) | 73 |
| 5.3.1 | Introduction..... | 73 |
| 5.3.2 | Module/Features for Admin..... | 74 |
| 5.3.3 | Application Features | 76 |
| 5.3.4 | Module/Features for Users (verified)..... | 82 |
| 5.3.4 | Module/Features for Users (not verified) | 121 |
| 5.4 | Implementation Issues and Challenges | 124 |

| | |
|---|------------|
| CHAPTER 6 | 125 |
| 6.1 Unit Testing | 125 |
| 6.2 Database Testing | 134 |
| 6.3 User Acceptance Testing (UAT) | 143 |
| 6.3.1 UAT Test Scenario | 143 |
| 6.3.2 UAT Test Result | 147 |
| 6.4 Usability Testing | 148 |
| 6.4.1 Survey Form..... | 148 |
| 6.4.2 Survey Result | 149 |
| CHAPTER 7 | 150 |
| 7.1 Conclusion | 150 |
| 7.2 Recommendation for future work..... | 152 |
| REFERENCES | 153 |
| APPENDIX A: FINAL YEAR PROJECT BIWEEKLY REPORT | 156 |
| APPENDIX B: POSTER | 160 |
| PLAGIARISM CHECK RESULT | 161 |

LIST OF FIGURES

| Figure Number | Title | Page |
|----------------------|--|-------------|
| Figure 1.1 | Commute mode in United State (2000-2019) | 5 |
| Figure 2.1 | Uber Pool User Interface Analysis | 18 |
| Figure 2.2 | Lyft Amp | 20 |
| Figure 2.3 | Lyft User Interface | 21 |
| Figure 2.4 | Via User Interface | 23 |
| Figure 2.5 | Ryde User Interface | 26 |
| Figure 3.1 | Agile Development Model | 34 |
| Figure 3.2 | Work Breakdown Structure | 37 |
| Figure 3.3 | Gantt Chart | 38 |
| Figure 4.1 | System Architecture Diagram | 39 |
| Figure 4.2 | Use Case Diagram | 40 |
| Figure 4.3 | Activity Diagram for Users (Login) | 48 |
| Figure 4.4 | Activity Diagram for Users (Sign-Up) | 49 |
| Figure 4.5 | Activity Diagram for Users (Forgot Password) | 50 |
| Figure 4.6 | Activity Diagram for Users (Manage User Profile) | 51 |
| Figure 4.7 | Activity Diagram for Users (Check Rides Requested) | 52 |
| Figure 4.8 | Activity Diagram for Users (Post Ride Request) | 53 |
| Figure 4.9 | Activity Diagram for Admin (Verify User Account) | 54 |
| Figure 4.10 | Activity Diagram for Users (Set Home Address) | 55 |
| Figure 4.11 | Activity Diagram for Users (Set School Address) | 56 |
| Figure 4.12 | Activity Diagram for Rider (Request Ride Along) | 57 |
| Figure 4.13 | Activity Diagram for Driver (Request Pick Up) | 58 |
| Figure 4.14 | System Flow Chart | 59 |
| Figure 5.1 | Android Studio Virtual Device | 65 |
| Figure 5.2 | Window Terminal (NPM runtime) | 66 |
| Figure 5.3 | Window Terminal (run android) | 66 |
| Figure 5.4 | Window Terminal (run android) | 67 |
| Figure 5.5 | Project APK files | 68 |
| Figure 5.6 | Physical Device APK installed | 69 |
| Figure 5.7 | Project Bundle Release | 70 |
| Figure 5.8 | Project Assemble Release (APK files) | 71 |
| Figure 5.9 | Huawei App Gallery Console (Publish App) | 72 |
| Figure 5.10 | Huawei App Gallery (DinoPool) | 72 |
| Figure 5.11 | Firebase Console (users) | 74 |
| Figure 5.12 | Firebase Console (users - imageURL) | 74 |
| Figure 5.13 | Firebase Console (imageURL) | 75 |
| Figure 5.14 | Firebase Console (verified status) | 75 |
| Figure 5.15 | Splash Screen (DinoPool) | 76 |
| Figure 5.16 | Location (DinoPool) | 77 |
| Figure 5.17 | Drawer Menu (DinoPool) | 78 |
| Figure 5.18 | Home Map (DinoPool) | 79 |
| Figure 5.19 | Route Map (DinoPool) | 80 |
| Figure 5.20 | Google Place Autocomplete (DinoPool) | 81 |
| Figure 5.21 | Module for verified users (Login) | 82 |

| | | |
|-------------|--|-----|
| Figure 5.22 | Module for verified users (Login) | 83 |
| Figure 5.23 | Module for verified users (Sign Up) | 84 |
| Figure 5.24 | Module for verified users (Sign Up) | 85 |
| Figure 5.25 | Module for verified users (Sign Up) | 86 |
| Figure 5.26 | Module for verified users (Forgot Password) | 87 |
| Figure 5.27 | Password Reset Link (Forgot Password) | 87 |
| Figure 5.28 | Module for verified users (Post Ride Requests) | 88 |
| Figure 5.29 | Module for verified users (Post Ride Requests) | 89 |
| Figure 5.30 | Module for verified users (Post Ride Requests) | 90 |
| Figure 5.31 | Module for verified users (Post Ride Requests) | 91 |
| Figure 5.32 | Module for verified users (Post Ride Requests) | 92 |
| Figure 5.33 | Module for verified users (Post Ride Requests) | 93 |
| Figure 5.34 | Module for verified users (Post Ride Requests) | 94 |
| Figure 5.35 | Module for verified users (Manage User Profile) | 95 |
| Figure 5.36 | Module for verified users (Manage User Profile) | 96 |
| Figure 5.37 | Module for verified users (Manage User Profile) | 97 |
| Figure 5.38 | Module for verified users (Check Requested Rides) | 98 |
| Figure 5.39 | Module for verified users (Check Requested Rides) | 99 |
| Figure 5.40 | Module for verified users (Check Requested Rides) | 100 |
| Figure 5.41 | Module for verified users (Check Requested Rides) | 101 |
| Figure 5.42 | Module for verified users (Check Requested Rides) | 102 |
| Figure 5.43 | Module for verified users (Search Rides) | 103 |
| Figure 5.44 | Module for verified users (Search Rides) | 104 |
| Figure 5.45 | Module for verified users (Search Rides) | 105 |
| Figure 5.46 | Module for verified users (Request Ride Along) | 106 |
| Figure 5.47 | Module for verified users (Request Ride Along) | 107 |
| Figure 5.48 | Module for verified users (Request Ride Along) | 108 |
| Figure 5.49 | Module for verified users (Request Ride Along) | 109 |
| Figure 5.50 | Module for verified users (Request Pick Up) | 110 |
| Figure 5.51 | Module for verified users (Request Pick Up) | 111 |
| Figure 5.52 | Module for verified users (Request Pick Up) | 112 |
| Figure 5.53 | Module for verified users (Set Home Address) | 113 |
| Figure 5.54 | Module for verified users (Set Home Address) | 114 |
| Figure 5.55 | Module for verified users (Set Home Address) | 115 |
| Figure 5.56 | Module for verified users (Set Home Address) | 116 |
| Figure 5.57 | Module for verified users (Set School Address) | 117 |
| Figure 5.58 | Module for verified users (Set School Address) | 118 |
| Figure 5.59 | Module for verified users (Set School Address) | 119 |
| Figure 5.60 | Module for verified users (Set School Address) | 120 |
| Figure 5.61 | Module for not verified users (Post Ride Requests) | 121 |
| Figure 5.62 | Module for not verified users (Request Ride Along) | 122 |
| Figure 5.63 | Module for not verified users (Request Pick Up) | 123 |
| Figure 6.1 | Firebase Storage Console | 134 |
| Figure 6.2 | Firebase Authentication Console | 135 |
| Figure 6.3 | Firestore Database Console (users) | 136 |
| Figure 6.4 | Firestore Database Console (riderRequest) | 138 |
| Figure 6.5 | Firestore Database Console (driverRequest) | 140 |
| Figure 6.6 | Firestore Database Console (rideAlongRequest) | 141 |
| Figure 6.7 | Firestore Database Console (pickUpRequest) | 142 |

LIST OF TABLES

| Table Number | Title | Page |
|---------------------|---|-------------|
| Table 1.1 | Proposed Objectives corresponding to Problems Statements | 9 |
| Table 1.2 | Scope of Carpooling Solutions and Methods | 11 |
| Table 1.3 | Findings of Types of Carpool Models | 15 |
| Table 2.1 | Summary of Existing System | 28 |
| Table 2.2 | Key Differences between Firebase and AWS | 32 |
| Table 2.3 | Key Differences between React Native and Flutter | 33 |
| Table 3.1 | Term Definitions | 36 |
| Table 4.1 | Login Use Case Description | 41 |
| Table 4.2 | Sign-Up Use Case Description | 42 |
| Table 4.3 | Forgot Password Use Case Description | 42 |
| Table 4.4 | Manage User Profile Use Case Description | 43 |
| Table 4.5 | Check Rides Requested Use Case Description | 43 |
| Table 4.6 | Post Rides Request Use Case Description | 44 |
| Table 4.7 | Verified User Account Use Case Description | 44 |
| Table 4.8 | Set Home Address Use Case Description | 45 |
| Table 4.9 | Set School Address Use Case Description | 45 |
| Table 4.10 | Rate Users Use Case Description | 46 |
| Table 4.11 | Request Ride Along Use Case Description | 46 |
| Table 4.12 | Request Pick Up Use Case Description | 47 |
| Table 5.1 | Modules/Features for user | 73 |
| Table 6.1 | Unit Testing for Rider/Driver (Login) | 125 |
| Table 6.2 | Unit Testing for Rider/Driver (Sign Up) | 126 |
| Table 6.3 | Unit Testing for Rider/Driver (Forgot Password) | 127 |
| Table 6.4 | Unit Testing for Rider/Driver (Manage User Profile) | 127 |
| Table 6.5 | Unit Testing for Rider/Driver (Check Rides Requested) | 129 |
| Table 6.6 | Unit Testing for Rider/Driver (Post Ride Request) | 130 |
| Table 6.7 | Unit Testing for Rider/Driver (Set Home Address) | 130 |
| Table 6.8 | Unit Testing for Rider/Driver (Set School Address) | 131 |
| Table 6.9 | Unit Testing for Rider (Request Ride Along) | 133 |
| Table 6.10 | Unit Testing for Driver (Request Pick Up) | 133 |
| Table 6.11 | Database Testing for Firebase Storage (sign up) | 134 |
| Table 6.12 | Database Testing for Firebase Authentication (sign up) | 135 |
| Table 6.12 | Database Testing for Firestore Database (sign up) | 136 |
| Table 6.13 | Database Testing for Firestore Database (post rider request) | 137 |
| Table 6.14 | Database Testing for Firestore Database (post driver request) | 139 |
| Table 6.15 | Database Testing for Firestore Database (request ride along) | 141 |
| Table 6.16 | Database Testing for Firestore Database (request pick up) | 142 |
| Table 6.17 | UAT Test scenario (Sign Up, Login and Logout) | 143 |
| Table 6.18 | UAT Test scenario (Post Ride Request and Check Rides Requested) | 144 |
| Table 6.19 | UAT Test scenario (Request Ride Along) | 144 |
| Table 6.20 | UAT Test scenario (Request Pick Up) | 145 |
| Table 6.21 | UAT Test scenario (Forgot Password) | 145 |

| | | |
|------------|---|-----|
| Table 6.22 | UAT Test scenario (Manage User Profile) | 146 |
| Table 6.23 | UAT Test scenario (Set Home and School address) | 146 |
| Table 6.24 | UAT Test Result | 147 |
| Table 6.25 | Usability Test (Survey Form) | 148 |
| Table 6.25 | Usability Test (Survey Result) | 149 |

Chapter 1

Introduction

In this chapter, we present the background and motivation of our project, which consist of problem statement and motivation, project objective, project scope and direction, contributions to the field, background information and report organization.

1.1 Problem Statement and Motivation

1.1.1 Problem Statement on Geographical Issues

The Rise of Private Own Car in Malaysia

In many cities today, the private car has become an important mode of transportation. The growing dominance of the private car is due to the advantages it offers users in terms of privacy and comfort. The freedom of owning a private car is an important reason why many people want to own a car instead of taking public transportation because they need to share the service with strangers. In addition, private cars can sometimes be considered as a symbol of social wealth and success. As the most developed city in Malaysia, Kuala Lumpur has a large number of newly registered cars every year. Although during the pandemic, the percentage of the total number of newly registered motor vehicles continues to rise each year. Thus, during the recovery from the COVID-19 pandemic, Malaysia's economy is gradually recovering, and the percentage of motor vehicle registrations is expected to increase. According to The Star's report, Southeast Asian markets may have the lowest car ownership rates in the world, except for Malaysia, which ranks third in the world with 93% of car ownership, and where purchase intentions are the highest among our neighbor countries with 71%. Malaysia is also expected to drive most of the world's car demand in the coming year.

Single-Driver Commuting are popular in Malaysia

According to a survey conducted by Centre for Governance and Political Studies (Cent-GPS), most of the Malaysians are preferring to drive alone to workplace instead of taking public transport or schedule a carpool. The survey was conducted during the morning rush hour on people who drive from Ampang, Seri Kembangan, Sungai Buloh, and Petaling Jaya going to work, with a total of 4689 drivers participate in this survey from 23 October to 25 October in year 2019. The survey found 87.3% of the drivers travel alone to work, only 11.6% drive with one passenger on board and only 0.9% travel with two passengers on board. The primary reason behind this statistic is believed to be the affordable cost of fuel and toll in Malaysia may not be an attractive enough incentive to urge people to take public transport or schedule carpool. The survey also found that drivers prefer to drive alone to work are because of the comfort and privacy, most of them would rather spend their morning driving alone to listen to their favourite music and gather their thought. Other reasons would be drivers prefer spending parking fee to park their cars at work instead of parking at LRT or MRT as they are cost the same. Sitting through traffic is just minor inconvenience for the drivers when compared to the inconvenience of carpool or public transport. Cent-GPS also found that Malaysia has a long way to go compared to other countries that have started to see a decline in the number of solo drivers, with many opting for public transport and carpooling.

Traffic Congestion in Malaysia

Today, traffic congestion has become a global problem in many developing countries around the world, and it negatively affects people and the country itself. In addition, traffic congestion also contributes significantly to air pollution due to the large amount of carbon monoxide emitted by every vehicle in the city. However, in Malaysia, traffic congestion is common, particularly in the bigger cities like Kuala Lumpur. According to the report from Traffic Index by City 2020 Mid-Year by Numbeo, Malaysia is ranked fourth with 169.14 points followed by Singapore at 148.61. The report also shows that Kuala Lumpur has an average Time Index of 41.99 minutes compared to our neighbor countries, which means that people in Kuala Lumpur average spend 41.99 minutes in the traffic congestions every day. There are many reasons why traffic congestion occurs in Malaysia, such as an increase in the number of vehicles each year, overpopulation in developed city, poor public transportation management and poor road conditions.

1.1.2 Problem statement on Current Carpool Methods

Despite all the benefits that show that carpooling is a reliable transportation option, there are also some problems that might be concerning.

a) Effectiveness Issue

The ineffectiveness of the transportation system is considered to be one of the most detrimental aspects of carpooling. In many countries, local governments' decisions to allocate preferential lanes for carpooling, although beneficial to users of this alternative mode of transportation (Manzini and Pareschi, 2012) [1], It also limits the space available for other vehicles. This has led to problems such as increased traffic congestion, increased fuel use and rising air pollution. (Calvo et al., 2004) [2].

b) Flexibility Issue

Although carpooling has been advertised from the beginning as a rather flexible mode of transportation, big data analysis has revealed a considerable degree of unhappiness as people have always been unable to adapt the service to their specific requirements and specifications. Indeed, regardless of the technology used to assist in describing the operation of the service, carpooling requires flexibility on the part of the user, as it encourages them to reach an agreement with an unknown person to determine the time, route, and cost (Li et al., 2007) [3]. This limitation is not confirmed in "flexible carpooling", so called because it does not require a pre-determined start or arrival time, but only a route for each vehicle (Dorinson et al., 2009) [4].

c) Personal Privacy

Carpooling, like many other multiuser services, requires users to register on the online platform, which allows them to be identified, have their information digitally traced, and offer feedback after using the alternative mode of transportation. However, while allowing for a minimum amount of information about the actors involved in the service, this inevitable activity raises privacy concerns (Kladedtiras and Antoniou, 2015) [5]. As with any kind of resource sharing, people's private spheres conflict with those of other users, who may, for example, find themselves listening to a phone conversation received or conducted by another passenger or driver. (Avodji et al., 2015 [6]; Friginal et al., 2014 [7]).

d) Personal Safety

Other issues that have been highly discussed with carpooling are personal safety issues, which are strongly tied to privacy as well. In fact, 767 tweets have made remarks indicating that people are worried about their personal safety while using a service shared with strangers. However, such statistics are not considered mature and must be further studied because, at least so far, crimes associated with the carpooling issue have been uncommon. In any case, crimes committed are no more common than those committed in more traditional modes of transportation such as taxis, buses, trains, and so on (Minett et al., 2018) [8].

e) Trust Issue

One of the top five reasons individuals claim they do not use carpooling is a lack of trust. As noted in 582 comments posted by Twitter users, all people who intend to use the platform must register in advance to use the app, and all transactions made through the platform are automatically tracked. But this is different from the case of shared services, where the same carpool of unknown people voluntarily chooses to spend different amounts of time, depending on the distance of the route (Ciasullo et al., 2018) [9].

f) Limited Usage of Carpool

Based on the yearly percentages of commute mode in the United States (United States et al., 2019 [10]) presented in Figure 1-1, it appears that carpooling usage has been consistent and restricted throughout the years.

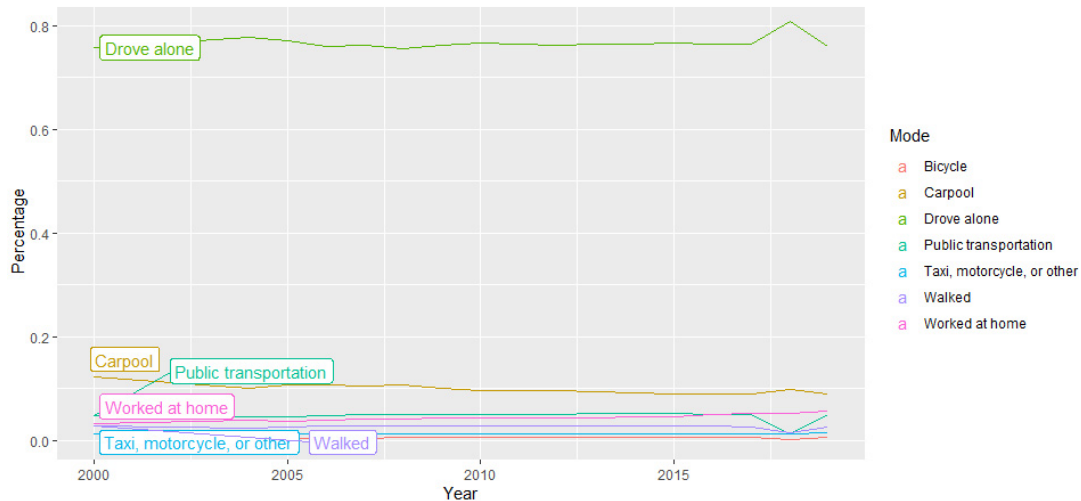


Figure 1.1 Commute mode in United State (2000-2019)

Low carpooling usage has also been shown in European countries, despite various measures to encourage carpooling. For example, European Union-funded initiatives such as increase in Car Occupancy (ICARO in 1997) and City-VITALity-Sustainability (CIVITAS in 2002 and 2005) attempted to reduce the number of automobiles on the road while improving car occupancy. Despite the European Union's investing heavily in these projects, their results have been limited, and their overall impact on carpooling in Europe has been minimal (Delhomme, P.; Gheorghiu, A., 2016) [11].

1.1.3 Motivation

Therefore, the motivation of this project is creating a carpooling application to improve traffic congestions and promote the benefits of carpooling. This is because carpooling allows transporting more people at one time resulting in fewer trips and cost-effective. Systems like these allow people to travel more effectively while creating less traffic and less pollutions. In a boarder aspect, carpooling also allow people to traveling with like-minded people, which can help them to expand their interpersonal relationships network. Furthermore, in today's Malaysia market, there is limited application that provide carpooling services. This could be due to the less awareness of carpool in Malaysia, like our above statement mentions, public transportation and carpool are not so popular in Malaysia as most of the people are more preferring to drive alone. Another factor can be due to the personal safety issues as most of the Malaysian considered Malaysia as a low security country, this is mostly because of the high crime rate in Malaysia and some people might worries or felt uncomfortable when commuting with strangers. However, this is only part of the people's thoughts, there is also people who are excited and willing to try different ways of commuting that can bring many benefits to them and to the community. We choose to set out target audience to be university student is because we believe that younger generation are much willing to try out new things and much familiar with the use of network and technology. Furthermore, most families have at least one car in Malaysia, and some even have a situation where each family member has their own car. It is mostly because, according to the average salary of Malaysians, cars are affordable in Malaysia, and national cars in Malaysia are popular in the markets as they are always the first choices of most people. This has also happened in Malaysia, where an average of university students borrows cars from their parents or other family members to get around during their studies. This has the advantage of encouraging university students to carpool with other students, which could be within the same university. Carpooling can be a great option for students who don't have a vehicle to commute in during their university years in some cities where public transportation remains difficult in terms of availability and accessibility. We hope to share the benefits of carpooling through this project with numerous Malaysia university students.

1.2 Project Objectives

The aim of this project is to develop a carpooling application for university students to provide an alternative way of transport and commute, yet also to provide a safer and more reliable environment for university students while enjoying the carpooling services.

Sub-Objectives

a) Provide authentication for user who sign up an account

Users need to have an account to use the carpooling services, therefore it is necessary for user to sign up an account. Users need to upload their student ID or related certify documents that can prove their identity as a university student to the system for the system to verify. User accounts that have not been verified will not be able to post any ride request or request any rides. This procedure is also to make sure we could better track on every user and to better ensure the safety and reliability of the carpooling services.

b) Provide options for users to post ride requests with driver profile or rider profile

Users are allowing to post request with driver profile or rider profile in the application. User who owns a car and wish to be a driver can upload their vehicle details such as the car model, license number and others details to the application to start fetching riders. On the other hand, user who do not own a car or did not feel to drive can also choose to use the service as a rider. In fact, both profiles will have the similar user interface in terms of functionality.

c) Provide user with the function to set their school address and home address

Users are encouraged to set their school address and home address whether the user is a driver or rider, as user can reuse the address to post their future ride request. This will make the match between driver and rider more predictable and more efficient. Drivers who have set their school address and home address in their account can reuse it for schedule their carpool services. Same as the rider, rider can also reuse the addresses set in their account to schedule for a ride.

d) Provide notification to alert users before the schedule ride time or when the driver has reached

Notifications provide convenient for both the driver and ride to remind them the schedule time or driver arrived. Therefore, driver won't have to wait for long time to have the rider noticed he has arrived. At the meantime, rider will also be alerted before the schedule time to be at the pick-up location so that they won't missed the ride.

e) Provide user with the function to give their feedback after every rides

Users are also encouraged to give their feedback after every ride, no matter driver or rider. User feedbacks are important for us to build relationship with them and to better understand their needs. As the carpooling service or application is not always perfect, user's feedbacks are crucial for us to constantly make improvement on the application. Besides, user feedback can also help to identify issues such as bad attitude or habit from a particular user for us to make further observation or actions.

f) Provide three different kinds of carpooling services with different carpool method

As mentions in the project scope, this application will provide three different carpooling services to cater different kind of needs from user. The three different services will use different carpool method to ensure the efficiency of the services. Such as the schedule rides service will use an informal carpool method as the ride are pre-arranged or schedule with user desire date and time. Second service which is extended from schedule rides is regular rides service, the service will allow users to post their regular ride request with recursive date and time and same route. Third service which is immediate rides service will allow users to post their ride request with real-time system date and time.

Chapter 1 Introduction

The table below shows how the objectives of this project are corresponding to provide solutions to the problems statement listed in Chapter 1.1.2. However, **problem statement (f)** we cannot find a solution yet as it involved of general environmental and countries' policy factors.

List of Problem Statement in Chapter 1.1.2:

- (a) Effectiveness Issue
- (b) Flexibility Issue
- (c) Personal Privacy Issue
- (d) Personal Safety Issue
- (e) Trust Issue
- (f) Limited Usage of Carpool

| Objectives | Problem Statements |
|--|---|
| Provide authentication for user who sign up an account | Problem Statement (d) |
| Provide options for users switch between a driver profile or rider profile | Problem Statement (b) |
| Provide user with the function to set their usual go to school time and go home time | Problem Statement (a) |
| Provide notification to alert users before the schedule ride time or when the driver has arrived | Problem Statement (a) |
| Provide user with the function to give their feedback after every ride | Problem Statement (c), Problem Statement (e) |
| Provide three different kinds of carpooling services with different carpool method used | Problem Statement (b) |

Table 1.1 Proposed Objectives corresponding to Problems Statements

1.3 Project Scope and Direction

This proposed project will focus on developing a carpooling application specifically for university students. The deliverable of this project is to help university students by providing them with an alternative way to commute during their daily routine or study time. We also aim to provide a safe and secure environment for university students while enjoying carpooling services by identifying all user's identification before they can request a service. In addition, we hope to bring the benefits of carpooling to all university students, such as providing an affordable ride based on shared costs, reducing one's carbon footprint, and providing a carpooling platform that allows users to connect with others during the ride, which also contributes to positive mental health.

By analyzing all the strengths and functionality of the existing application in the previous report, we have found some basic and necessary functionality that needs to be included in this project. First is user authentication. The user will need to sign up as a member of the application in order to use the carpooling service. We also need to verify the user's identity to determine whether the user is a registered university student or not. Next, the user can post their ride request with different roles such as rider or driver, which means that user can use the app with a driver profile or a rider profile. Both profiles will provide a similar but different user interface in terms of functionality. Next is to provide users with the ability to set their school address and home address. Furthermore, it provides notification to alert users before reaching their booked time for the ride. Next, which is a crucial function, is to allow users to provide feedback for all the rides. User feedback is important for us to keep track of the user quality. Finally, the application will provide three kinds of carpooling services for the user, which can be separated into immediate rides and schedule rides and regular rides.

To come up with a carpooling application with a suitable carpool model, we have analyzed all the types of carpool models and methods that are available on the market. We have found some of the carpool solutions or methods that can be included in our carpooling services. The scope of carpooling solutions and services provided are refined as below:

| Carpooling Services | Carpool Methods & Solutions |
|---------------------|---|
| Schedule Rides | This service will use an Informal Carpool method. Which means that most of the ride are pre-arranged and scheduled. because most of time university students are going to commute between school and hostel, this service is suitable for university students who in these situations where the destination and depart time might already be decided before they want to schedule a carpool. |
| Regular Rides | This service is extended from schedule rides. User can post their ride request with recursive date and time on the same route. This will be suitable for students who are going to commute to school every week at the same time, or go out to have breakfast, lunch, and dinner every day. |
| Immediate Rides | This service will use a Dynamic Carpool method. Which means that most of the ride are requested by the users on demand. The service will be much similar as the service provided in the reviewed carpooling application VIA and Ryde. It will make the pick-up more efficient and easier for the driver or even rider to search for the ride. But users might not find the matched rider or driver in short time or wait in vain. |

Table 1.2 Scope of Carpooling Solutions and Methods

1.4 Contributions

Our project will bring benefits to university students and the community. Although carpooling might not be as comfortable as using your own car and may be more time-consuming, the widespread use of ridesharing has been used by other countries to prove the feasibility of carpooling. From a social perspective, less fuel consumption, fewer CO2 emissions, less traffic congestion, and more social interaction are the benefits of ridesharing. By reducing air pollution and reducing traffic congestion, carpooling is considered an improvement in sustainable public transportation.

1.5 Background Information

1.5.1 The History of Carpooling

Carpooling is an old technology that has been promoted by many government programs for more than half a century (Bresciani et al., 2018) [12]. During World War II, carpooling became popular as a cost-cutting measure. It was pushed by the American government as a sustainable fuel source. Carpooling has evolved from a need to save rubber and fuel for wartime activities in North America to becoming an important aspect of regional infrastructure planning and reducing the "inevitable" cost of congestion. The government's determination to educate and motivate each citizen with clear, forceful messages about the essential need for everyone to take action, including carpooling, created a large conservation campaign in the United States in the 1940s (Cadyjacob, 2010) [13]. In the 1970s, the carpooling idea reappeared and grew in popularity due to a similar shortage of resources. During this period, the first "vanpools" were built for Chrysler and 3M employees. However, when gas prices declined in the 1990s, carpooling became less popular, although it remained popular among a specific population in the 1990s: college students. With limited parking on college campuses, students began carpooling to save themselves the hassle of finding a parking spot. When the 20th century arrived, only a few years after the start of the new millennium, technology began to transform the carpooling game. Carpooling has grown in popularity in recent years as a result of the development of many web platforms that enable those people who are seeking and/or offering rides to connect and help organize their travel plans.

1.5.2 Types of Carpooling

According to Anthopoulos (L.G.) and Tzimos (D.N.) [14], The types of carpooling on the market can be divided into two main general types: regular trips (e.g., daily commuting to work) and ad hoc trips, where users request a trip from one place to another. In the United States, the term "ad hoc carpool" is commonly used, and it refers to user-run carpooling in the form of three or more commuters per vehicle.

Besides that, there are also another 3 types of carpool models that can be further analyzed, which were organized by (JJ O'Brien, 2020) [15]. The types of carpool models are summarized as below:

| Carpool Models | Findings |
|----------------------------------|---|
| Open vs Exclusive Networks | <p>The most common approach is to build on an open network where any user in the system can be paired with another user for a ride. However, this can create security issues for businesses and organizations that want to provide secure carpooling services to a large group of employees or students. Therefore, it is necessary to create a trusted network for each user.</p> <p>The open network approach may be less efficient because users may have different destinations. Carpoolers on private networks are usually traveling to and from the same workplace, school or other location.</p> |
| Fixed vs Dynamic Ride-Matching | <p>Another typical ride-matching technique is based on pre-defined groups, routes, or zip codes, which is not efficient. After that, it is up to the matched users to organize their travel and vacation plans. In contrast, dynamic ride matching adapts to real-time requests and intelligently selects the best route.</p> |
| Cost-Sharing vs No-Cost to users | <p>Some carpooling platforms use a cost-sharing mechanism. They charge a fee to share the cost with users. For business purposes, this configuration is the least efficient</p> |

| | |
|--|---|
| | <p>because it creates another barrier to user adoption.</p> <p>Establishing carpooling is difficult enough without adding additional friction. When commuters must pay and submit their payment information to the system, they are less likely to use carpooling services.</p> <p>On the other hand, some companies have pioneered a no-cost method for consumers to further incentivize participation by rewarding regular carpooling. The rewards are paid for by a portion of the customer's membership fee. This is a small but effective investment to keep carpoolers motivated and interested, while addressing the problems caused by single-occupancy car use on campus or in the area.</p> |
|--|---|

Table 1.3 Findings of Types of Carpool Models

1.6 Report Organization

The details of this project are shown in the following chapters. In Chapter 2 literature reviews, some related review on the existing carpooling application is done and some carpooling method that have been used in the market. Then, proposed method and approach are introduced in Chapter 3. And then, Chapter 4 describe the preliminary work of the project. Furthermore, Chapter 5 include the conclusion of the chapters written.

Chapter 2

Literature Review

This chapter is comprised of three sections. The first sections will be reviews of several related carpooling applications that are available on the market. The application's review will be more focused on analyzing its strengths and functionalities, and further reviewing its overall user interface that helps with user selection and navigation. The second sections will be about the carpooling solutions and methods that are used in most of the carpooling applications. We will be further analyzing the types of carpooling methods and their impacts on the carpooling sector. The last section will be discussing technologies used in mobile application development, we do comparison hardware platform, cloud database and development framework.

2.1 Review on Existing Application

2.1.1 Uber

Uber is a well-known app that is headquartered in San Francisco, which is among some of the world's largest digital companies. They have increased their footprints in over 630 cities across the globe and have an estimated 110 million worldwide users. They are the ridesharing companies that have been at the forefront of fundamentally altering the transportation sector. However, Uber has stopped its ride-hailing service in Malaysia since the year 2018, following its South-East Asian operations' merger with Grab.

Uber's prices are competitive and given the number of drivers on the platform in other countries, consumers should have no problem getting normal fares for travel on most days. At the same time, Uber continued its trend of innovation by offering Uber Pool, which allows users in some cities to get cheaper rides by sharing their rides with other passengers and letting the system algorithm match riders taking a similar route. Users are allowed to split the fare among passengers all the time if they are riding with a larger group, which is a great convenience for students who regularly go out with friends. This not only proved popular as a cost-saving technique, but it also allowed passengers to reduce their carbon footprint. In

addition, Uber was developed in a way to cater to the convenience of users in terms of payment methods; it offers a variety of payment methods such as credit cards, Android Pay, PayPal and cash in selected cities.

Uber has also offered some driver-oriented features, such as a passenger-rating system to identify tough riders, and the system will remove them from the platform if necessary. In addition, Uber also offers some bonuses for the driver in certain cities based on the number of rides given, and special considerations will also be given for hearing-impaired drivers. Drivers are also given the choice to opt-out of Uber Pool for drivers who aren't trying to get a bonus by giving more rides. In rider-oriented features, passengers may schedule their rides in advance. They can also have the ability to split fares between friends sharing a ride and also set up multiple drop-off points for groups of riders.

For the user interface analysis on Uber Pool, which is shown in Figure 2-1 (Varun Nambiar, 2019) [16], after the users have entered their destination address, they can view the estimated trip route and the trip cost of the ride before they decide to make a request. If users are satisfied with the trip cost, they can press the button to confirm the carpool request. The system will then match the riders with the driver and show the vehicle details of the driver and their location on the map. Users can check on the further ride details, which show the driver's ETA (Estimated Time of Arrival) and the destination ETA. Co-rider details will also be displayed to let the users know the person's name they are riding with. The user will be alerted when the driver is nearby the user's location and ready to pick them up. At the same time, users can also see the real-time location of the driver and the vehicle's license plate to make sure they are not in the wrong vehicle.

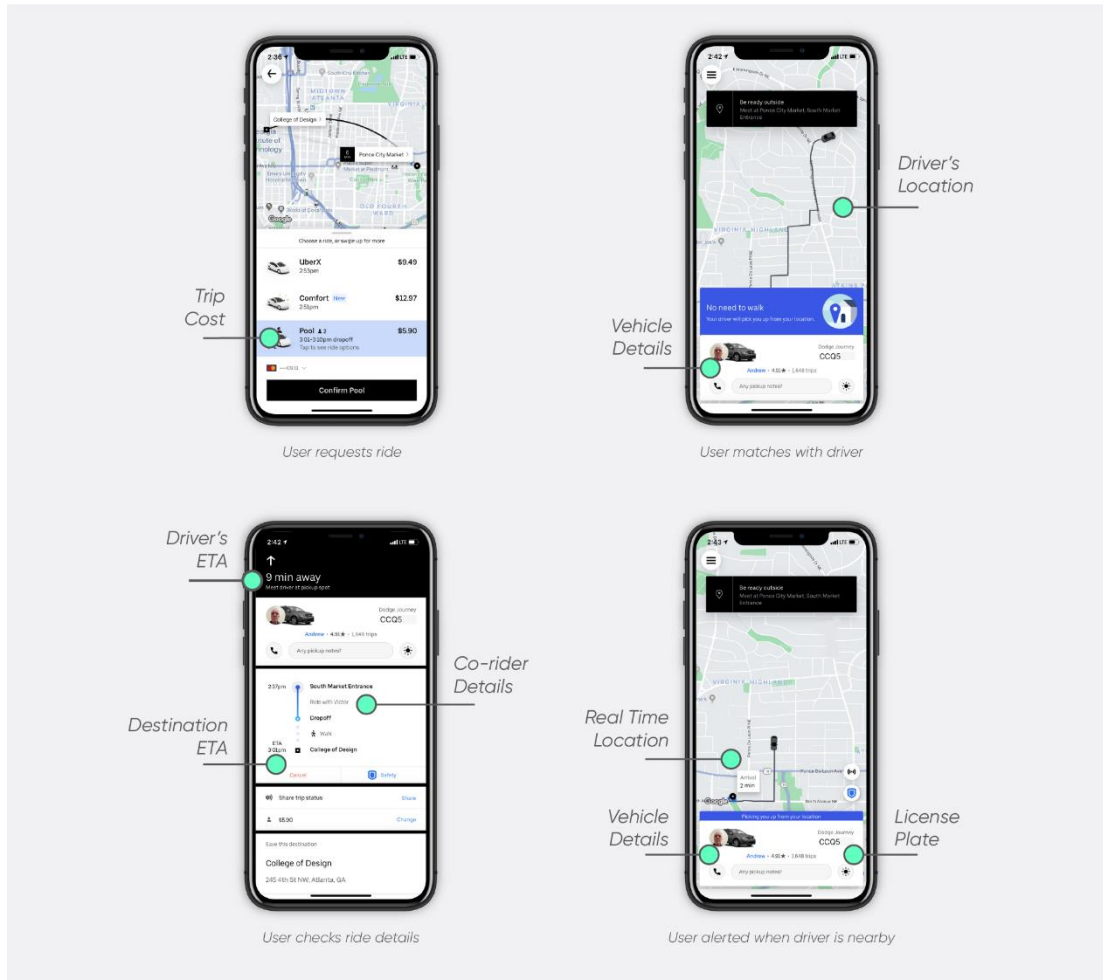


Figure 2.1 Uber Pool User Interface Analysis

2.1.2 LYFT

Lyft, founded in San Francisco, is another popular app. Lyft is now the second largest carpooling service in the industry based on ridership. Lyft operates in more than 600 cities across the U.S., including New York City, San Francisco and Los Angeles. As part of its global expansion, Lyft's car-sharing service is available in 12 locations across Canada. They have also embraced the growing trend of using scooter and bike-sharing services.

The business promises to offer an affordable, memorable, and welcoming rider experience. Uber's brand identity is distinct from Lyft's. Lyft refers to itself as "your friend with a car," whereas Uber refers to itself as "your personal driver." While the differences may appear slight, they result in a unique experience for both riders and drivers. In other words, it cultivated a unique "culture" among its users. Lyft passengers and drivers socialize more, and it was normal for Lyft passengers to sit in the front car before the epidemic. Uber riders preferred to sit in the rear seat and didn't engage in much conversation. Its vehicles range in size from mid-sized Lyft cars to bigger Lyft Plus vehicles. Lyft offers demand-based pricing during peak hours to make commuters' lives simpler. This ride-sharing service also allows users to choose an autonomous vehicle. The Lyft app is functionally identical to the Uber app. Once the app is launched, the next step is to request a ride by entering your location data into the app. You'll be matched with a driver, and you can track their real-time whereabouts or contact them through the app to get updates. For safety reasons, Lyft conducts thorough background checks, including DMV (Department of Motor Vehicles) checks, on all Lyft drivers. Only the drivers with the highest ratings are permitted to drive after receiving their driver's license.

Lyft offered some driver-oriented features, such as a separate app for passengers and drivers that included driver-specific features. Drivers are given the ability to indicate the current ride as their last ride, so they don't feel forced to accept an unexpected request. Lyft drivers are also given the opportunity to earn tips from riders. Although Uber does this as well, Lyft is considered the pioneer of it. For rider-oriented features, riders can choose a "shared-saver" ride, which means allowing the driver to pick up or drop off other rides that are headed in the same direction, and the rider will pay less for the shared ride. Riders may always use the "one tap to ride" option for quick requests and pickups. For rider safety, drivers will have to pass a background check to provide services. For rider convenience, some drivers will be given a "Lyft Amp," as shown in Figure 2-2, which is a glowing emblem to make pick-up easier for both drivers and riders.



Figure 2.2 Lyft Amp

For user interface analysis on Lyft, as shown in Figure 2-3 (Darrell Etherington, 2017) [17], This appears to imply that the functionality is the same as Uber. A user can choose to enter their destination address or choose a location that has been pre-set on their account. After that, the user can choose different kinds of ride services that they wish to request. Different services will have different trip fares. From the same screen, users can also choose their payment method for the current trip. Once the user has decided on the service selection, they can press the Go button to make the ride request. The system will then match the driver and display the driver's details, such as the driver's name, car model, and license plate. The user will be informed when the driver has arrived. The user can also choose to call the driver to make contact or cancel the current ride if they don't wish to continue.

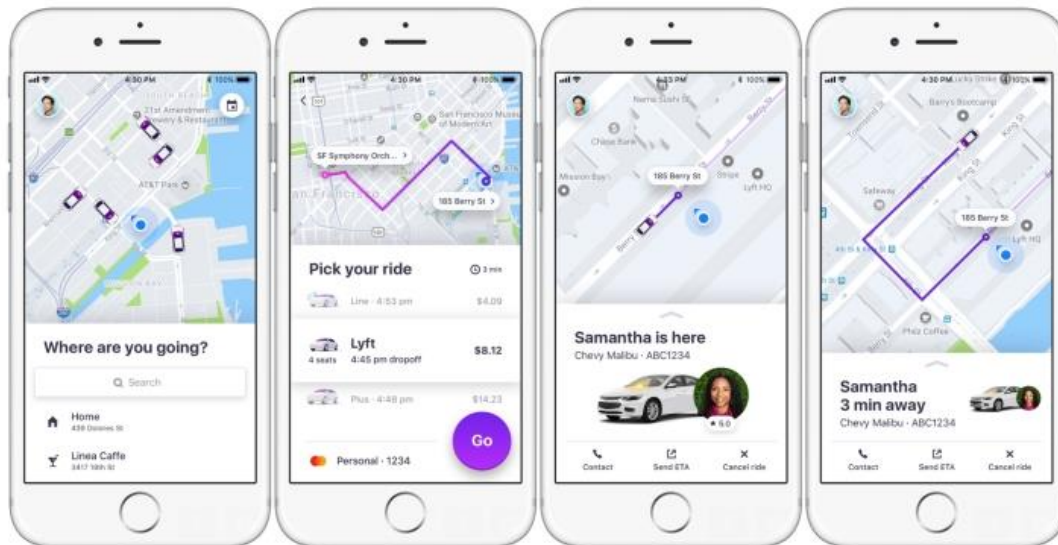


Figure 2.3 Lyft User Interface

2.1.3 VIA

VIA is an application that is popular for providing dedicated ridesharing services. VIA is headquartered in New York and has spread its branches across 20 countries worldwide. It runs under its own brand in only six countries, and in the rest of these operating countries, it runs in partnership with local companies. Daniel Ramot, a Ph.D. in neuroscience at Stanford University, co-founded the company and added a "logistics engine" to the app designed to fill all the seats in cars headed to popular locations. Unlike Uber and Lyft, this ride-sharing service doesn't reach every part of the city and the route is fixed. Once the customer simply selects his destination, the system tries to drop the passenger off at a close location along the route. Because the route is fixed, users often have to walk a block or two to get to the pick-up point, although the system allows passengers to bring one or more companions. The system will automatically find cars with enough seats for extra members, and each extra member in the passenger party rides at half price. To compensate for the route model, VIA will always operate, and it can be used in some of the busiest cities such as Chicago, New York, Los Angeles, Washington, D.C. In 2017, VIA announced a partnership with Curb to offer carpooling in yellow cabs. This move has greatly helped more riders get around, reducing their carbon footprint while saving money.

VIA has offered some driver-oriented features, such as a sign-up bonus of up to \$400 for new drivers. People who wish to join as drivers but can't afford a car can use a system called "rent a car". It's available for them to rent a car and start to earn. VIA also provides flexibility to the driver whereby the driver can choose to drive whenever it's convenient. Besides, VIA will only take a 10% commission from the drivers, compared to Uber's 25% commission fee for each ride. For rider-oriented features, the ride fares will be more affordable compared to other applications. Riders can also refer friends to get free ride credits on the app to encourage riders to share rides with their friends. Because the service will only pick up multiple passengers that are heading in the same direction, the pick-up will be more efficient and faster, but the passenger might have to walk a distance to reach the designated pick-up spots.

Chapter 2 Literature Review

For user interface analysis on Via, as shown in Figure 2-4 (Brett Helling, 2018) [18]. The functionality of the application will be much simpler compared with Uber and Lyft. For user to book a ride, the blue marker on the application will indicated the closest pickup location to the user, and the orange marker is for user to select their drop off location. User will be able to view the pickup time, the price of the ride, and the details of the trip. Once the user has confirmed their pickup location and destination, user can press on the “Book This Ride” button to confirm the ride. And user will have to head to their pickup location on time and look for the right car to ride.

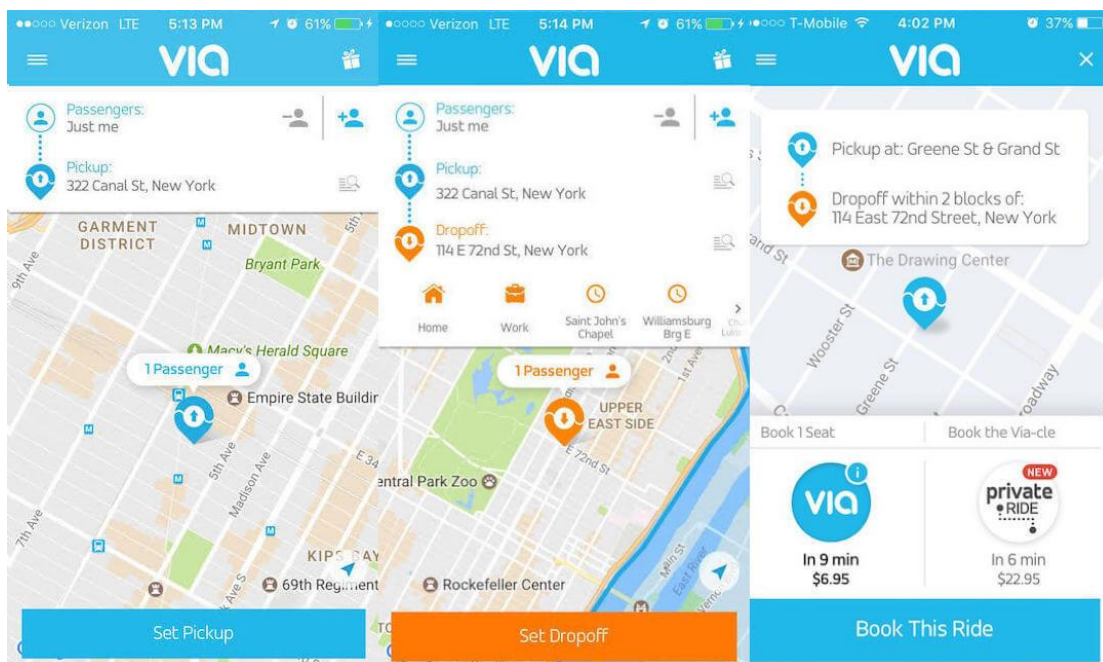


Figure 2.4 Via User Interface

2.1.4 Ryde

Ryde is a carpool application that was founded by its CEO, Terence Zou, and is headquartered in Singapore. Their services are currently available in Singapore and will be launched in Malaysia in 2019. They claim that it's Singapore's first and most popular carpooling app, providing users with a sustainable alternative to the daily commute and addressing the city's congestion and pollution problems. Furthermore, this ride-sharing software uses GPS technology to connect drivers with passengers who are heading in the same direction. The feature is called RydePOOL, and it means that drivers and passengers are matched if they are travelling in the same direction. This contrasts with the popular e-hailing service, Grab, where drivers deliver passengers to their preferred destination regardless of whether it is convenient or not. Furthermore, the app's algorithm will calculate the fare price depending on the distance travelled, allowing the user to know the actual amount before they confirm the request. Passengers can pay cash directly to the drivers at the end of the trip according to the division cost of the trip. To provide extra safety features to the users, Ryde Singapore has around \$1,000,000 in insurance coverage for sexual offence claims and third-party liabilities. They did so in response to multiple cases of sexual assaults that happened while users were using the carpool service. They also provide insurance for accidental death and permanent disability for passengers while travelling in the carpool. To be eligible for the insurance, the user must be a qualified Ryde car-pooler. To become a member, users must register using their application, and they will also provide evaluations and comments from both sides. Driver and passenger can give feedback to one another. Any party whose rating is lower than 4 stars will be suspended from the service for one month. This is to ensure a good user experience for both parties.

For rider-oriented features, RydePOOL allows passengers to book a trip for up to four passengers, and the passenger can manually input the number of passengers while booking the ride. The driver will then know how many seats in their car are booked and how many more passengers they can take on for the trip. Besides, RydePOOL allows passengers to schedule their ride starting from 3 minutes before the ride or up to 7 days in advance. This has made it convenient for passengers who just happen to remember to have somewhere to be at the last minute. Users can also set their work and home locations for ease of use, especially for users who plan on using the service daily. Furthermore, it also allows users to set their usual leave home time and usual leave work time to make finding a commute more efficient

and easier. One of the features that was not offered by any other similar carpooling application is that it provides passengers with an option to choose a preferred driver. If the passengers had a particularly memorable carpool experience with a certain driver on Ryde, they could choose to put in a private request to hitch a ride with them again. Lastly, Ryde has also offered a pet-friendly carpooling service, which allows passengers to bring along their common household pets, such as dogs, cats, rabbits, etc.

As for user interface analysis on Ryde, as shown in Figure 2-5 (APKPure, 2021) [19], the user interface looks much more complicated compared to other applications. To book a ride, users can search for an available pick-up point and destination on the app. After that, the user can choose which service to go with. An estimated trip fare is then presented to the user. On the same screen, users can change the number of people they wish to ride with and choose an available payment option for the trip. One of the features that was not commonly found on the other apps was that users could increase the fare to have a higher chance of getting matched with a driver. After the user has confirmed the selection, they can press the book button to confirm the request. Once the user has successfully matched with a driver, the app will display the driver's information such as the driver's name, car model, and license number. The app will track the driver's real-time location as they approach.

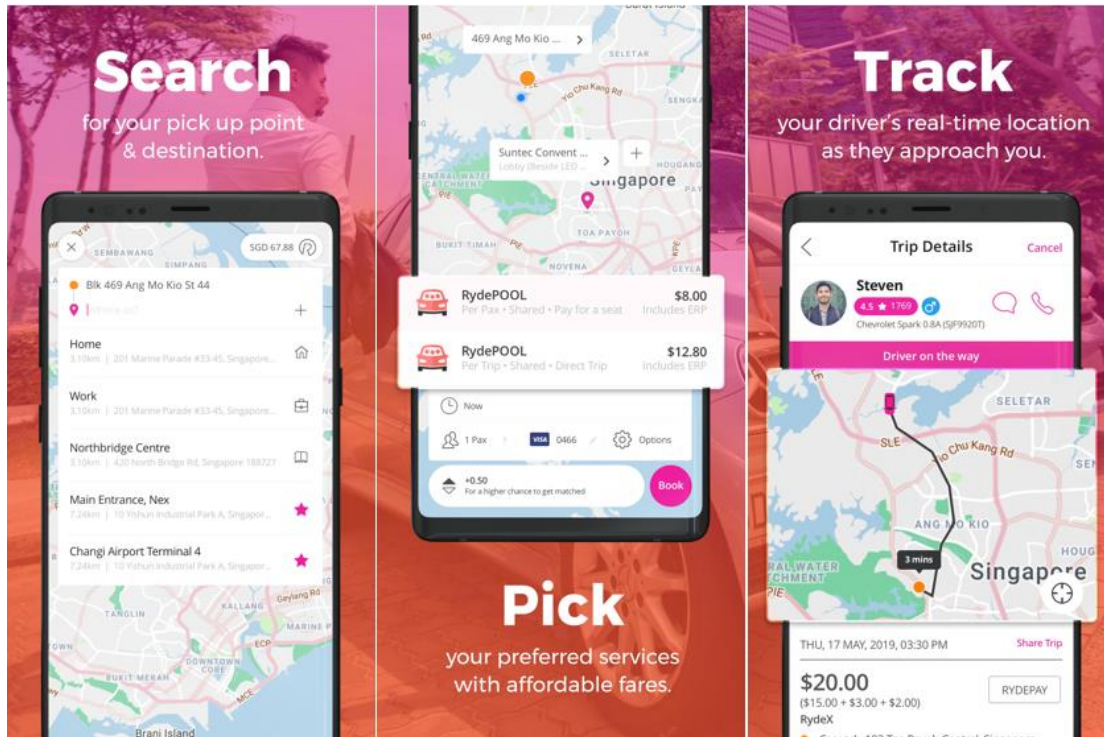


Figure 2.5 Ryde User Interface

2.1.5 Summary of Existing System

| Uber | Features Provided |
|----------------------------------|---|
| Driver-Oriented Features | <p>Passenger Rating system.</p> <p>Offer bonuses based on number of rides.</p> <p>Special Consideration given for hearing-impaired drivers.</p> |
| Rider-Oriented Features | <p>Schedule rides in advance.</p> <p>Split fares while sharing a ride.</p> <p>Multiple drop-off points.</p> |
| Lyft | Features Provided |
| Driver-Oriented Features | <p>Separate App which included with driver-specific features.</p> <p>Indicate last ride to force to accept request.</p> <p>Allow to earn tips from riders.</p> |
| Rider-Oriented Features | <p>“Shared-saver” ride allow pick up or drop-off headed in the same direction.</p> <p>“One tap to ride” allow quick requests and pickups.</p> <p>Background check on driver for ride safety.</p> <p>“Lyft Amp” used for easier pick up.</p> |
| VIA | Features Provided |
| Driver-Oriented Features | <p>Sign-up bonus up to \$400.</p> <p>“Rent a car” feature for driver who cannot afford a car.</p> <p>Provide flexibility choose when to drive.</p> <p>Lower commission charge.</p> |
| Rider-Oriented Features | <p>More affordable ride fare.</p> <p>Refer friends to get free ride credits.</p> <p>Efficiency on pick up passenger heading to the same direction.</p> <p>Walk to pick-up spot.</p> |
| Ryde | Features Provided |
| Driver & Rider-Oriented Features | <p>Connect driver – passenger heading same direction</p> <p>Insurance coverage for sexual offence</p> <p>Feedback system between both parties</p> <p>Schedule ride from minimum 3 minutes</p> <p>Set usual leave home & leave work time</p> |

| | |
|--|---|
| | Choose preferred driver feature Pet-friendly |
|--|---|

Table 2.1 Summary of Existing System

2.2 Review on Carpooling Solutions & Methods

2.2.1 Dynamic Carpooling

Dynamic carpooling is a type of trip sharing designed to accommodate any new drivers or passengers joining or exiting the "pool" of carpoolers in real time, (Arnould et al., 2011) [20] said. This requires knowledge of any events that could affect the trip, such as traffic congestion, incidents, accidents, construction, etc. Dynamic carpooling has been around since the early 1990s, but only recently has it shown itself to be profitable. According to (Friginal et al., 2014) [7], recent research in China (Xin et al., 2009) [21] has shown a growing interest in dynamic carpooling solutions, with the main benefits being cost savings and traffic reduction. It allows users to use mobile devices with GPS and social media connections to create personalized trips on the go in exchange for a certain amount of money. (Mallus et al., 2017).

Dynamic carpooling is also known as real-time carpooling, on-demand carpooling, instant carpooling, and temporary carpooling. (Amey et al., 2011) [22]. It is a revolutionary social incentive service that allows users to share vehicles effortlessly (Friginal et al., 2014) [7]. It increases the most in areas where traditional modes of transportation (buses, trains, trams, cabs, and eventually railroads) are unable to meet demand. The tendency to utilize carpooling only in emergency situations distinguishes this form of carpooling from other forms of carpooling where it is cheaper to use other modes of transportation to meet scheduled or regular requirements (Créno, 2016) [23]. According to (Massaro et al., 2009) [24], Dynamic carpooling overcomes the limitations of traditional carpooling by dynamically matching large numbers of passengers and drivers in real time, allowing for on-the-spot transportation arrangements. Recent dynamic carpooling systems also allow customers to take multiple vehicles on a single route. Although comfortable in theory, however, this form of transportation has failed to live up to expectations (Grgureviet al., 2015) [25].

2.2.2 Informal Carpooling

Informal carpooling began in the mid-1970s to accommodate the needs of passengers and drivers (Chan and Shaheen, 2012) [26]. In fact, the former can benefit from the ability to travel to specific locations for free or at a lower cost than traditional travel systems; on the other hand, drivers can use dedicated lanes reserved by some local governments (especially in the United States and Canada), reducing road congestion and helping to reduce pollution levels (Masoud and Jayakrishnan, 2017) [27].

On the other hand, carpooling was and still is an alternative mode of transportation that is entirely privately organized and controlled. The potential to save time rather than make money was the primary consideration when people considered offering informal carpooling services. This feature, combined with the fact that they are used mainly in the morning, is what sets them apart from other types of carpooling. (Badger, 2011) [28]. (Masoud and Jayakrishnan, 2017) [27] feels that carpooling is often organized and happens between people who are interested, rather than based on the time and place of their trip. In this regard, (Mote and Whitestone, 2011) [29] noted that carpooling has become increasingly popular because of the ability to meet lane requirements on an informal basis without having to coordinate and schedule the number of passengers needed daily. Therefore, it is easy to start carpooling because there is no need to register or make plans in advance, and most people hear about carpooling from friends and colleagues (by means of word of mouth).

2.2.3 Flexible Carpooling

Flexible carpooling is a form of carpooling that rose to prominence in the early 1980s (Chan and Shaheen, 2012) [26]. It is a cross between dynamic carpooling and informal carpooling. In fact, it requires the creation of vehicle routes, but does not specify departure or arrival times (Beroldo, 1990) [30]. In other words, users can use this alternative mode of transportation by going to a pre-determined meeting location, offering, or hitchhiking. Thus, the main advantage of flexible carpooling is the flexibility to travel along a certain route without having to arrange it in advance (Dorinson et al., 2009) [4].

(Minett, 2009) [31] says the flexible carpooling approach benefits from the fact that trips don't have to be arranged one by one in advance. On a first-come, first-served basis, people gather at a meeting place and load up in the order of their arrival. In the informal system, there is no pre-registration process for offering and receiving transportation, and no money is exchanged. Therefore, this type of carpooling is well suited for situations where last-minute transportation needs arise. (Minett et al., 2008) [8] said people benefit from flexibility in three ways: first, they can enter another user's car or allow another participant to enter theirs; second, they can participate as a driver and passenger, switching freely between the two sides each day; and third, they can use the system at different times of the day. However, with the enhanced flexibility, passengers may have to wait a long time at the meeting place to find an available car or wait in vain for a car to arrive (Shaheen et al., 2016 [32]; Kelly, 2007). Therefore, flexible carpooling may be seen as an emergency means of transportation, as it is not possible to rely on it 100% of the time (Minett et al., 2008) [8].

2.3 Review on the Technologies

2.3.1 Hardware Platform

Comparison between Android and IOS platform

I. Ease of use

Apple products are become popular nowadays, mostly due to the ease of use of IOS interface has gain many satisfactions from people. However, Android interface will allow more control for the users compared to IOS interface, lots of customization on the screens can be done on Android that IOS interface does not supported. And Android also have more application available for users compared to IOS, as IOS only supported App Store to be the only ways to acquire application while Android provide much flexibility for users to choose from.

II. Closed and open systems

IOS is well known to be a closed operating system, which users will not have the flexibility on selection of application. iPhone users will always to be locked into Apple software ecosystem because of the closed software. While Android being an open-source software, it able to accept more alternative applications, which provide more flexibility for the users to install any APK or alternative application.

III. Security

When it comes to security, IOS has more advantage compared with Android, it is not Android have problems, it is due to Google has more flexibility than Apple on what application are allows to be publish into its app store. For Android users to keep away from malware is to download apps from Google play store. Even so, Google report still indicate that 0.16% of its apps contain malware. In general, IOS is considered more secure, but that is build on the user's trust on how Apple treat users' privacy.

2.3.2 Cloud Database

Comparison between **Firestore** and **AWS Amplify**

I. Similarities

- Free tier available
- Both are app development platforms
- Scalable architecture
- Designed for mobile and web apps
- Both deliver Realtime database updates
- Provide authentication

II. Key Differences

| | AWS Amplify | Google Firestore |
|-------------------|---------------------------------------|---------------------------------------|
| Parent Company | Amazon | Google |
| Platform | Open-source Framework | Proprietary Technology |
| Cloud | AWS | GCP |
| API | REST and GraphQL APIs | REST APIs only |
| Database | Works with Dynamo DB | NoSQL database |
| Support Platforms | Native IOS, Android or via frameworks | Native IOS, Android or via frameworks |
| Storage | Directly on S3 | Google Cloud Storage |

Table 2.2 Key Differences between Firestore and AWS

To conclude, AWS Amplify is an open-source framework that fully integrated with AWS, works with GraphQL and other database, and it support mobile or web development. Firestore is a BaaS (Backend-as-a-Service) service from Google, work with NoSQL databases and fully integrated with REST API, it is easy to use, fully managed and scalable.

2.3.3 Development Framework

Comparison between React Native and Flutter

I. Similarities

- Free and open source
- Hot reload
- Great native performance
- Cross Platform capacity

II. Key Differences

| | React Native | Flutter |
|----------------------|---------------------------------|---------------------|
| Parent Company | Facebook | Google |
| Framework | Building native app using React | Portable UI toolkit |
| Programming Language | JavaScript | Dart |
| Packages & Libraries | NPM (Node Package Manager) | Pub-Dev |

Table 2.3 Key Differences between React Native and Flutter

To conclude, React Native is a much mature framework and has been longer time in the market, it uses much popular programming language which is JavaScript, it builds native apps or web apps using React framework. React Native uses native components which the app's components will be upgraded along with any OS UI updated. Flutter is a much newer framework that launched by Google, it uses Dart as main programming language, it builds apps web apps using portable UI toolkit and using single codebase to render on its own canvas to deliver the apps. React Native's ecosystem is huge and much bigger than Flutter as NPM is a repository that has been around for ages and consist of lots of packages and libraries that is ready to be used, however there are also many low-quality libraries that is hardly usable. Flutter using Dart's pub-dev repository that is dynamically growing in time, but it is still young and without as many ready used solutions.

Chapter 3

System Methodology/Approach

In this chapter we present the project planning methods used, design specifications and project plan in this project.

3.1 Project Planning Methods

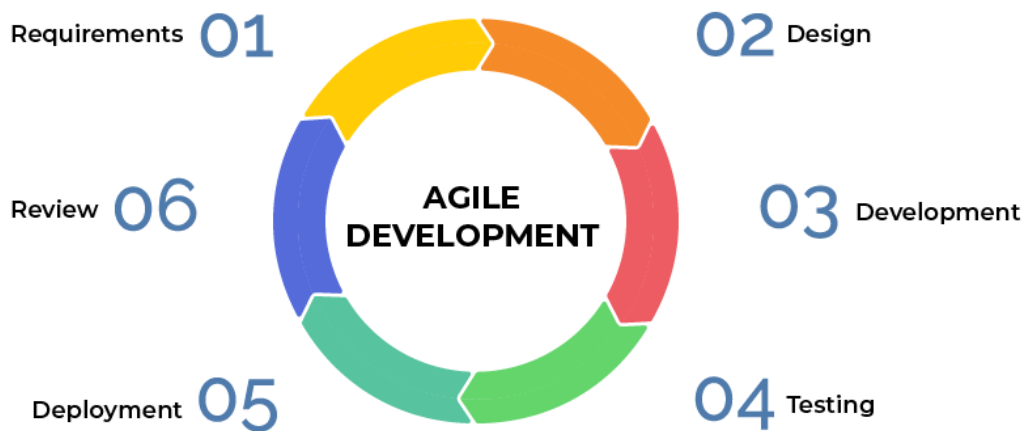


Figure 3.1 Agile Development Model

Agile methodology will be used for this project as it is suitable for this project compared to waterfall method. In this methodology, an iterative approach has been chosen for this project. This is because the iterative approach allows us to develop the product in a cyclical way, building small parts in an evolving way, instead of starting with a complete understanding of the requirements. In this case, the developers only needed the functional part of the requirements. In addition, there are high-risk features and requirements that may need to be changed regularly in this project, such as integration tools, carpooling models, and mapping APIs, for which the developers may not be familiar with the relevant technology. Therefore, there is an iterative process for each feature, such as analyzing, designing, coding, testing, and going back to the analysis process. In addition, an iterative approach allows for more focus on user value. These cycles will continue until the testing process is successful, at which point development will continue to the next feature of the application.

3.2 Design Specifications

3.2.1 Project Description

This section will contain the general definitions that will be used throughout the project. The application will be a system that connects drivers and riders. Users need to have a verified account in order to use the services. Users can post their ride request with different roles, i.e., driver or rider. Riders can request a ride at any time when they need to travel to another location. Pre-scheduling will also be implemented in the future for riders who need to travel to the same location at a specific time each day. On the other hand, riders will be able to insert the pick-up points along with the driver's departure location and destination of the trip each time they want to be picked up. The app will offer users three types of carpooling services, which can be divided into immediate rides, schedule rides and regular rides. Immediate rides will use a dynamic carpooling method, which means most of the rides are requested by the users on demand. Scheduled rides will use an informal carpooling method, which means that most rides are pre-arranged and have been scheduled. Regular rides is extended from schedule rides which means most of the rides are pre-schedule with recursive date and time.

Users can set their home and school address in the application that can be reused in their next ride request posting.

a) Users

The users of the application can be students or commuters that need to travel to another places or users that are driving and want to pick up riders. The users can post ride requests with **drivers** or **riders** while using the application.

- The **driver** is a user who owns a car, wants to travel elsewhere, and wants to pick up riders by entering his origin and destination to share ride with riders.
- The **rider** is a user who does not own a car and wishes to request a ride that matches the driver's route.

b) Definitions

In order not to confuse all the terms used and to facilitate good habits in this report, the following table shows the frequently used terms and definitions of use.

| Term | Definitions |
|-----------------------|--|
| Driver | User who owns a car, wants to travel elsewhere, and wants to pick up riders by inserting his origin and destination to share ride with riders. Driver can request to pick up rider in rider request posts. |
| Rider | User who does not own a car and wishes to request a ride that matches the driver's route. Rider can request to ride along with driver's route in driver request posts. |
| Immediate Ride | An immediate service will allow user to post their ride request with their departure and destination location and real-time system date and time. |
| Schedule Ride | A schedule service will allow users to post their ride request with pre-arranged departure and destination location and time. |
| Regular Ride | A regular ride service will allow users to post their ride request with departure and destination location and recursive date and time. |

Table 3.1 Term Definitions

c) Payment Methods

Since this a community kind of application, the application will not implement any payment methods function to both drivers and riders. The application platform also will not charge any fees from the users.

3.3 Project Plan

In this section we present the Work Breakdown Structure and timeline of the project.

3.3.1 Work Breakdown Structure

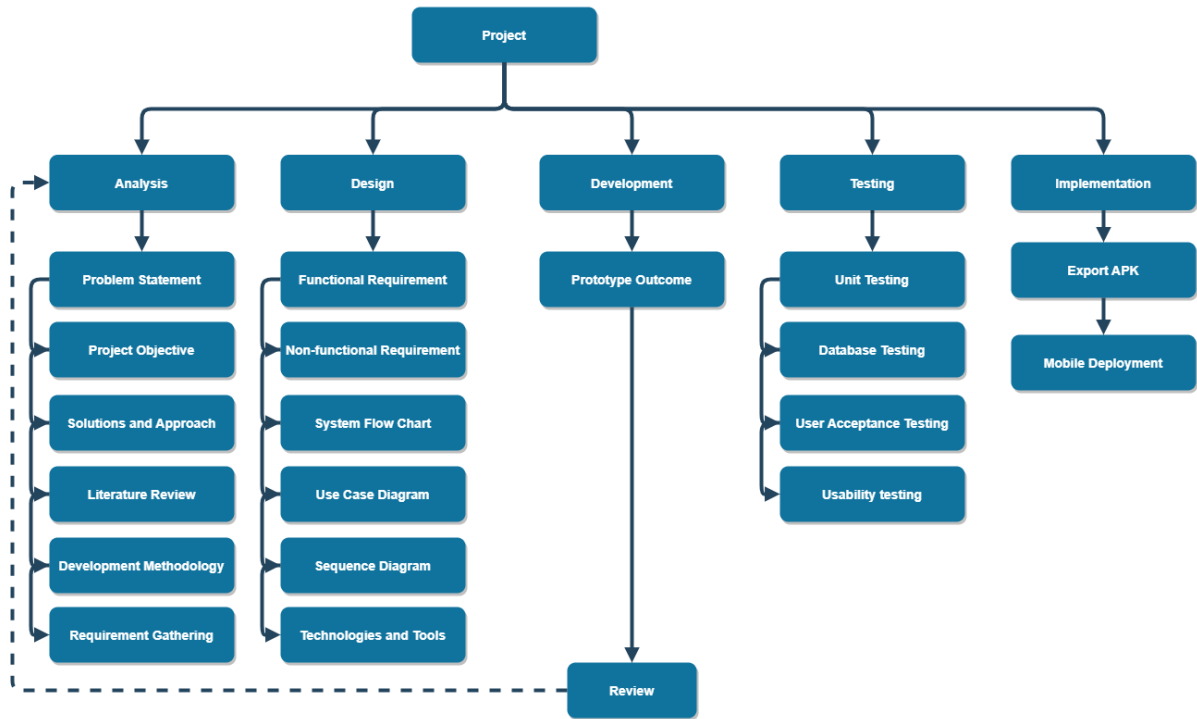


Figure 3.2 Work Breakdown Structure

3.3.2 Timeline

This section will present the estimated timeline for deliverables and milestones.

The estimated timeline given for project 2 will be from 13th June 2022 until 21st August 2022.

Which is from week 1 to week 10 of the June 2022 semester.

Below table are the functions that will be implement in this project label with number in sequence:

| Sequence | Function Name | Days Given |
|------------|--|------------|
| Function 1 | Login & Register | 9 |
| Function 2 | Manage User Profile | 11 |
| Function 3 | Set Regular Trip | 5 |
| Function 4 | Notification | 15 |
| Function 5 | Rate Driver/Rider | 13 |
| Function 6 | Carpooling Service (Flash & Schedule Ride) | 16 |

Table 3-2 Timeline Table

Gantt Chart

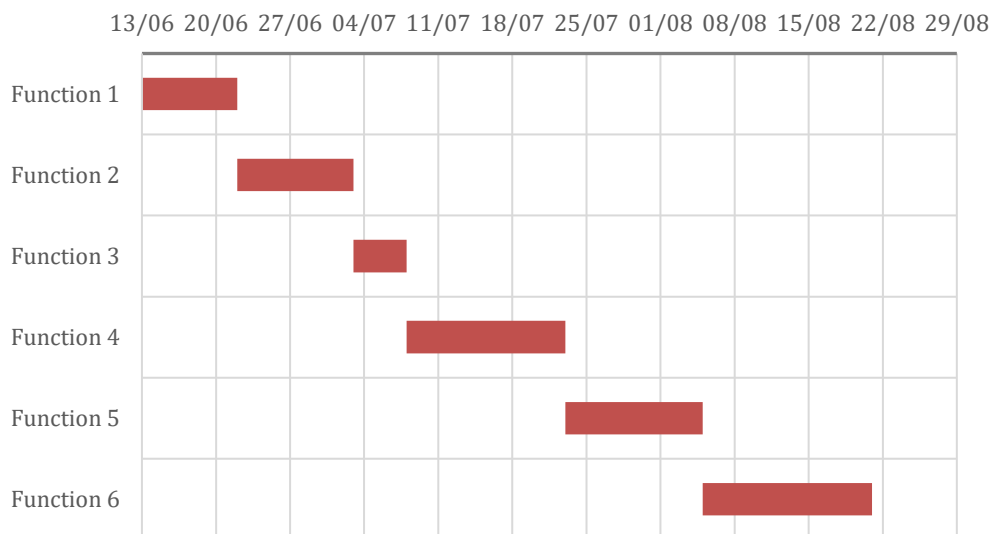


Figure 3.3 Gantt Chart

Chapter 4

System Design

In this chapter we present the system design diagram and technologies specification in this project.

4.1 System Design Diagram

4.1.1 System Architecture Diagram

The architecture used in this project will be 3-tier (layer) architecture, which is commonly used as any client-server architecture application. The architecture used will be able to increase the efficiency and accuracy of data retrieving process. The increase scalability of the architecture also benefits in future planning.

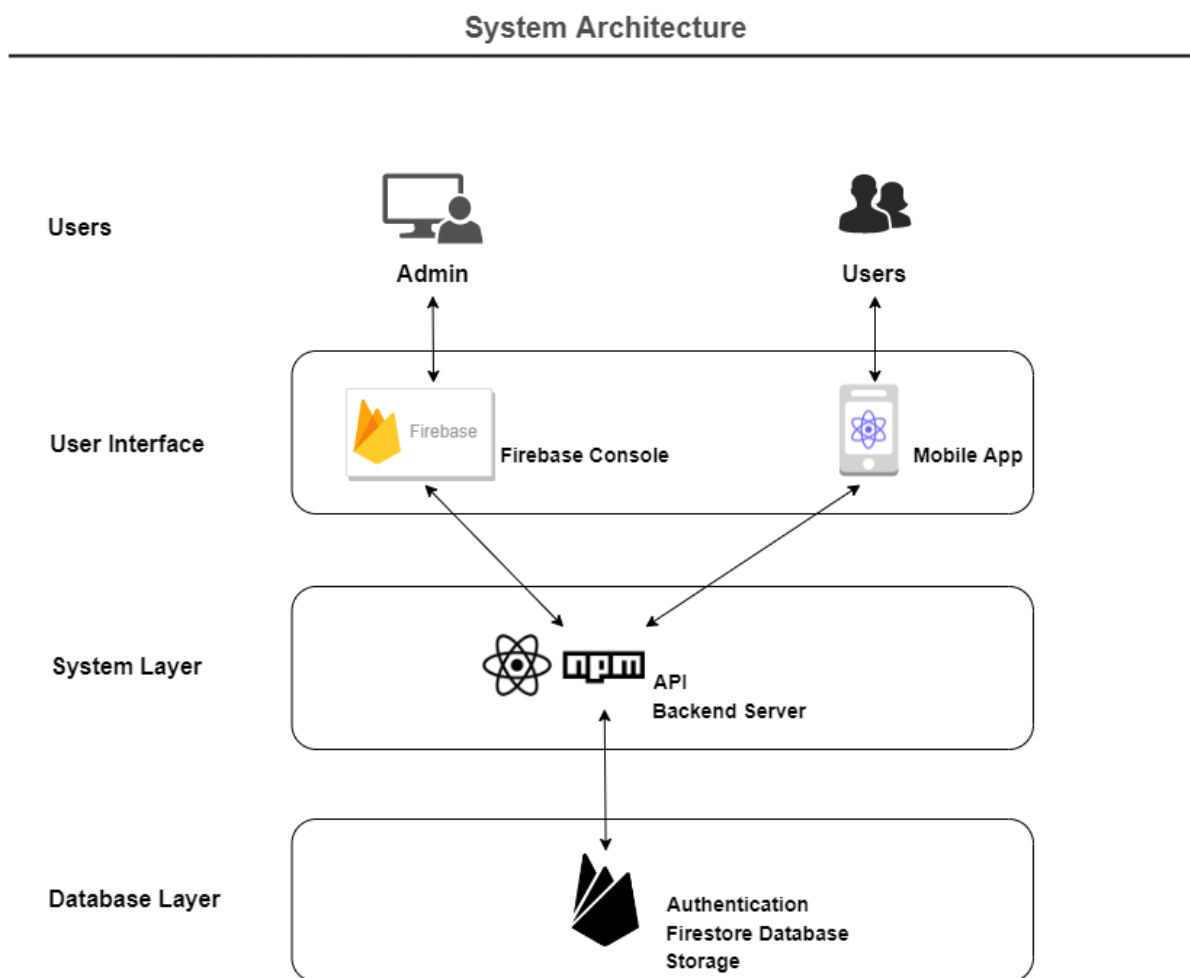


Figure 4.1 System Architecture Diagram

4.1.2 Use Case Diagram

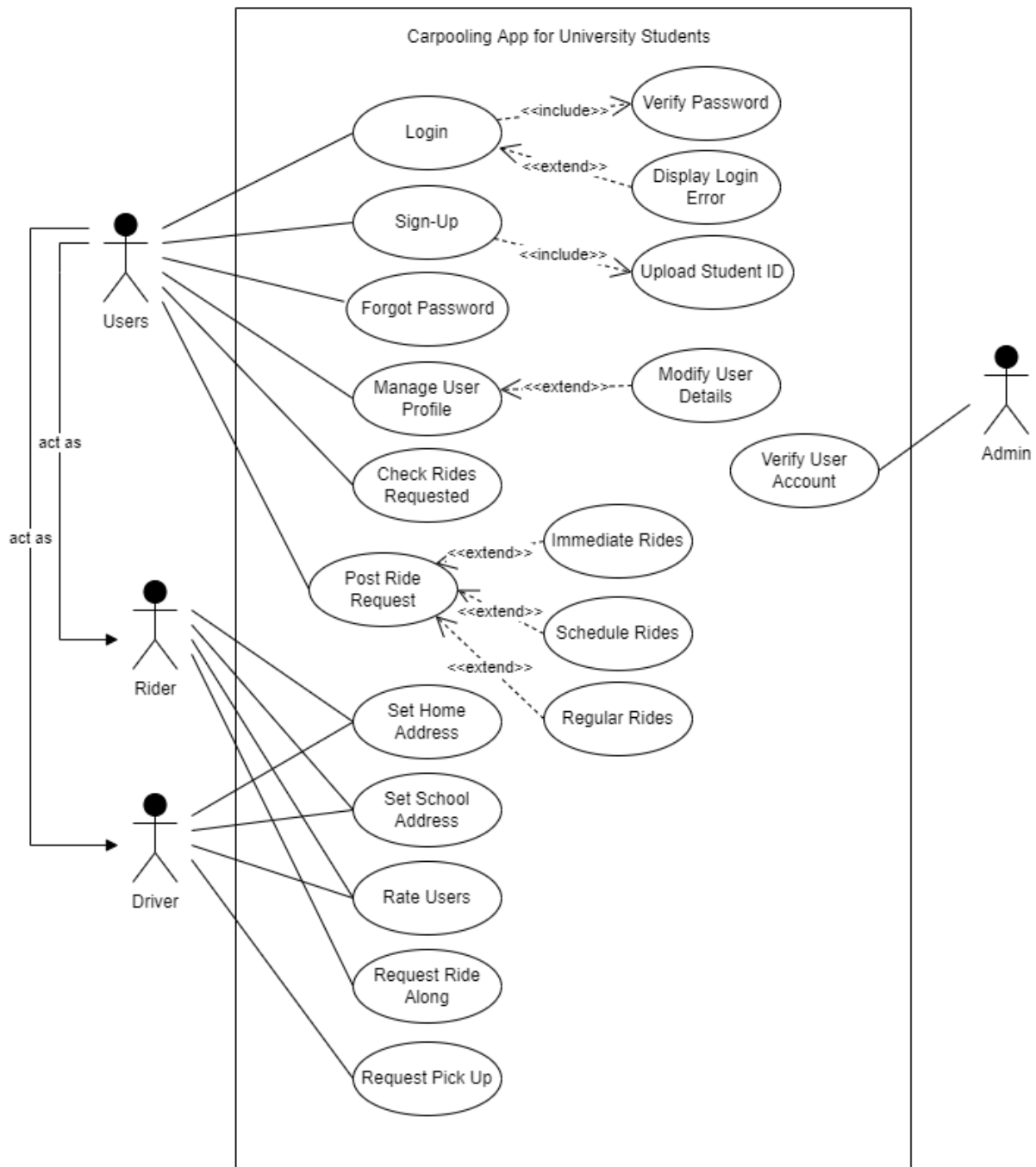


Figure 4.2 Use Case Diagram

4.1.3 Use Case Description

| | |
|---|-------------------------------|
| Use Case ID | 1 |
| Use Case Name | Login |
| Actors | Users |
| Description | Users login in their account. |
| <p>Flow of events:</p> <ol style="list-style-type: none"> 1. Users enter their email and password to login into the application. 2. Application will display message indicate successful login activities. | |
| <p>Alternative flow of events:</p> <ol style="list-style-type: none"> 1.1 Users did not enter any value in email or password input. 1.2 An alert message will prompt to remind users an email or password input is required. 2.1 Users login with wrong email or password. 2.2 An alert message will prompt display login failed and remind users their email or password entered is incorrect. | |

Table 4.1 Login Use Case Description

| | |
|---|--------------------------------|
| Use Case ID | 2 |
| Use Case Name | Sign-Up |
| Actors | Users |
| Description | Users sign up for new account. |
| <p>Flow of events:</p> <ol style="list-style-type: none"> 1. Users enter their personal details that include username, email address, password, repeat password, phone number, and upload their student id image from their phone’s photo gallery to register a new account. 2. Application will display message indicate successful register activities. | |
| <p>Alternative flow of events:</p> <ol style="list-style-type: none"> 1.1 Users did not enter any value in username, email address, password, repeat password, phone number. 1.2 Users did not upload their student id image from their phone’s photo gallery. 1.3 An alert message will prompt to remind users that a username or email address or password or phone number or student id image upload is required. | |

| |
|---|
| 2.1 Users entered password does not match with repeat password entered. |
| 2.2 An alert message will prompt to remind users that password is not matched with repeat password. |
| 3.1 Users click on sign in button to navigate to the sign in page if the user has owned an account. |

Table 4.2 Sign-Up Use Case Description

| | |
|--|--|
| Use Case ID | 3 |
| Use Case Name | Forgot Password |
| Actors | Users |
| Description | Users request to reset their account password. |
| Flow of events: | |
| 1. Users enter their email address to request for reset password. | |
| 2. Application will display message indicate successful send reset password link activities. | |
| Alternative flow of events: | |
| 1.1 Users entered email address does not exist in database. | |
| 1.2 An alert message will prompt to remind users email address not available. | |

Table 4.3 Forgot Password Use Case Description

| | |
|---|-------------------------------------|
| Use Case ID | 4 |
| Use Case Name | Manage User Profile |
| Actors | Users |
| Description | Users manage their account details. |
| Flow of events: | |
| <ol style="list-style-type: none"> 1. Users has successfully login into their account. 2. Users navigate to profile page to edit their personal details. 3. Application will display message indicate the personal details update successful activities. | |
| Alternative flow of events: | |

Table 4.4 Manage User Profile Use Case Description

| | |
|---|-------------------------------------|
| Use Case ID | 5 |
| Use Case Name | Check Rides Requested |
| Actors | Users |
| Description | Users manage their account details. |
| Flow of events: | |
| <ol style="list-style-type: none"> 1. Users has successfully login into their account. 2. Users navigate to my rides page to check on their ride request posts. 3. Users click on requested rider/driver's icon to contact the requested user. | |
| Alternative flow of events: | |
| <ol style="list-style-type: none"> 1.1 No other users have requested on user's ride request post. | |

Table 4.5 Check Rides Requested Use Case Description

| | |
|--|--|
| Use Case ID | 6 |
| Use Case Name | Post Ride Request |
| Actors | Users |
| Description | Users post their ride request to the server. |
| <p>Flow of events:</p> <ol style="list-style-type: none"> 1. Users has successfully login into their account. 2. Users navigate to post ride request. 3. Users enter their origin and destination location and select for available location searched. 4. Users given selection to post ride request as a rider or driver profile. 5. Users given selection to choose between immediate, schedule, and regular ride services. 6. Users enter date and time to submit ride request. | |
| <p>Alternative flow of events:</p> <ol style="list-style-type: none"> 1.1 Users did not select any rider or driver profile. 1.2 An alert message will prompt to remind users a profile selection is required. 2.1 Users did not enter any date and time. 2.2 An alert message will prompt to remind users a date and time input is required. | |

Table 4.6 Post Rides Request Use Case Description

| | |
|--|---|
| Use Case ID | 7 |
| Use Case Name | Verify User Account |
| Actors | Admin |
| Description | Admin manage and verified users' account. |
| <p>Flow of events:</p> <ol style="list-style-type: none"> 1. Admin has successfully login into firebase console. 2. Admin review on new users' personal details. 3. Admin update new users' account verified status to be true. | |
| <p>Alternative flow of events:</p> <ol style="list-style-type: none"> 1.1 Admin found unusual or fake student id uploaded be user. 1.2 Admin update new users' account verified status to be false. | |

Table 4.7 Verified User Account Use Case Description

| | |
|---|---|
| Use Case ID | 8 |
| Use Case Name | Set Home Address |
| Actors | Rider & Driver |
| Description | Rider or Driver can set their home address in the settings. |
| <p>Flow of events:</p> <ol style="list-style-type: none"> 1. Users has successfully login into their account. 2. Users navigate to settings. 3. Users enter their new home address and select for available location. 5. Users done selection and update addresses. 6. Application will display message indicate successful update address activities. | |
| <p>Alternative flow of events:</p> <ol style="list-style-type: none"> 1.1 Users did not enter any address in home address input. 1.2 An alert message will prompt to remind users to enter address. | |

Table 4.8 Set Home Address Use Case Description

| | |
|---|---|
| Use Case ID | 9 |
| Use Case Name | Set School Address |
| Actors | Rider & Driver |
| Description | Rider or Driver can set their school address in the settings. |
| <p>Flow of events:</p> <ol style="list-style-type: none"> 1. Users has successfully login into their account. 2. Users navigate to settings. 3. Users enter their new school address and select for available location. 5. Users done selection and update addresses. 6. Application will display message indicate successful update address activities. | |
| <p>Alternative flow of events:</p> <ol style="list-style-type: none"> 1.1 Users did not enter any address in school address input. 1.2 An alert message will prompt to remind users to enter address. | |

Table 4.9 Set School Address Use Case Description

| | |
|---|--|
| Use Case ID | 10 |
| Use Case Name | Rate Users |
| Actors | Rider & Driver |
| Description | Rider or Driver can rate the user after every rides. |
| <p>Flow of events:</p> <ol style="list-style-type: none"> 1. Users has successfully login into their account. 2. Users rate on ride requested posted by another user. 3. Application will display message indicate successful rate activities. | |
| <p>Alternative flow of events:</p> <ol style="list-style-type: none"> 1.1 Users had rated on the ride request post. 1.2 An alert message will prompt to remind users had rated on the ride request. | |

Table 4.10 Rate Users Use Case Description

| | |
|---|---|
| Use Case ID | 11 |
| Use Case Name | Request Ride Along |
| Actors | Rider |
| Description | Rider request to ride along on driver's request post. |
| <p>Flow of events:</p> <ol style="list-style-type: none"> 1. Riders has successfully login into their account. 2. Riders navigate to search for available driver's ride request post. 3. Riders click on one of the driver's requests posted and check on ride details. 4. Riders click on request to ride along. 5. Riders long press and drag location pin to select location to be pick up on driver's route. 6. Riders submit request. 7. Application will display message indicate successful request activities. | |
| <p>Alternative flow of events:</p> <ol style="list-style-type: none"> 1.1 Riders have requested on the ride request post. 1.2 Application will indicate that rider has requested on the driver's request post. | |

Table 4.11 Request Ride Along Use Case Description

| | |
|---|---|
| Use Case ID | 12 |
| Use Case Name | Request Pick Up |
| Actors | Driver |
| Description | Rider request to pick up on rider's request post. |
| <p>Flow of events:</p> <ol style="list-style-type: none"> 1. Drivers has successfully login into their account. 2. Drivers navigate to search for available rider's ride request post. 3. Drivers click on one of the rider's requests posted and check on ride details. 4. Drivers click on request to pick up. 5. Drivers submit request. 6. Application will display message indicate successful request activities. | |
| <p>Alternative flow of events:</p> <ol style="list-style-type: none"> 1.1 Drivers have requested on the ride request post. 1.2 Application will indicate that driver has requested on the rider's request post. | |

Table 4.12 Request Pick Up Use Case Description

4.1.3 Activity Diagram

Login

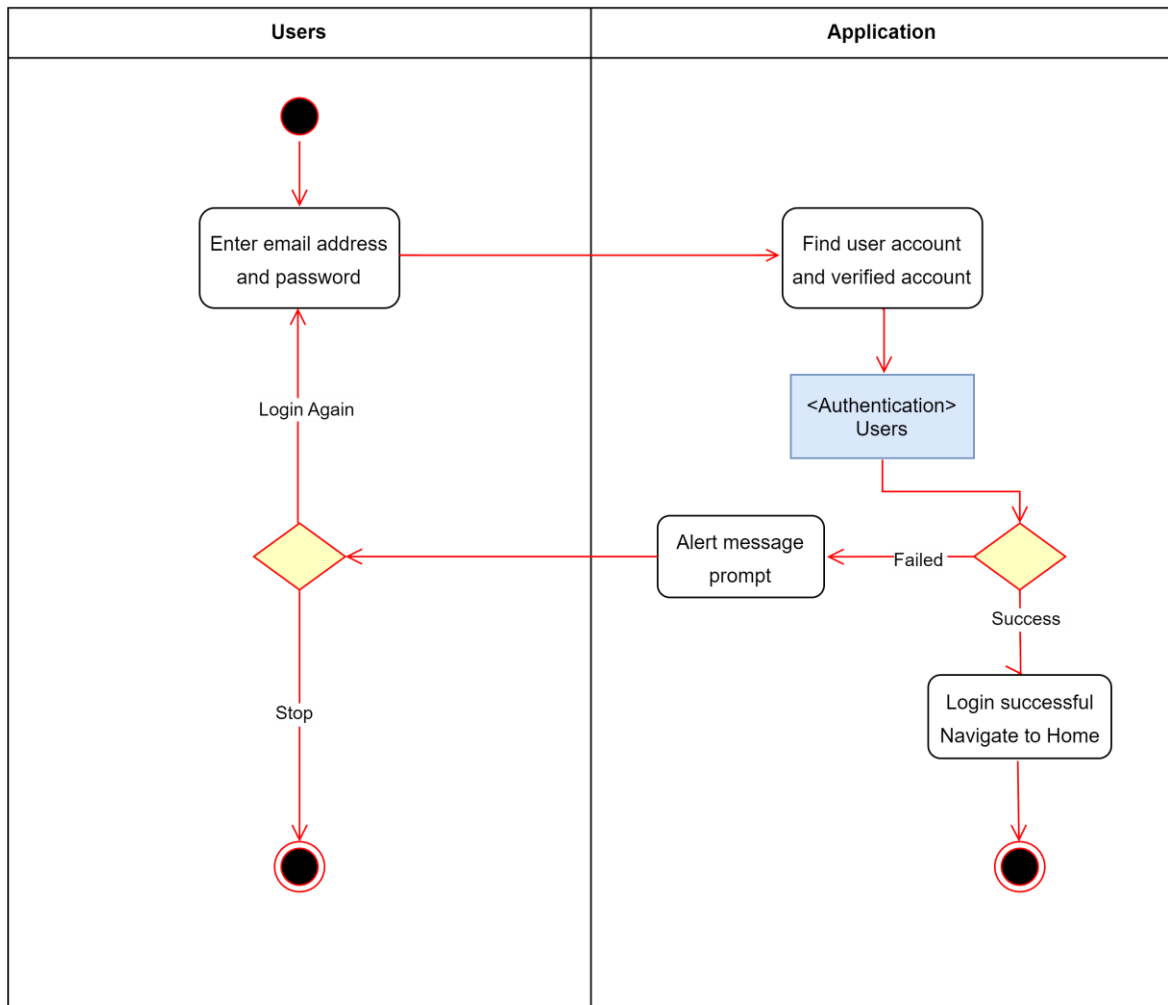


Figure 4.3 Activity Diagram for Users (Login)

Sign-Up

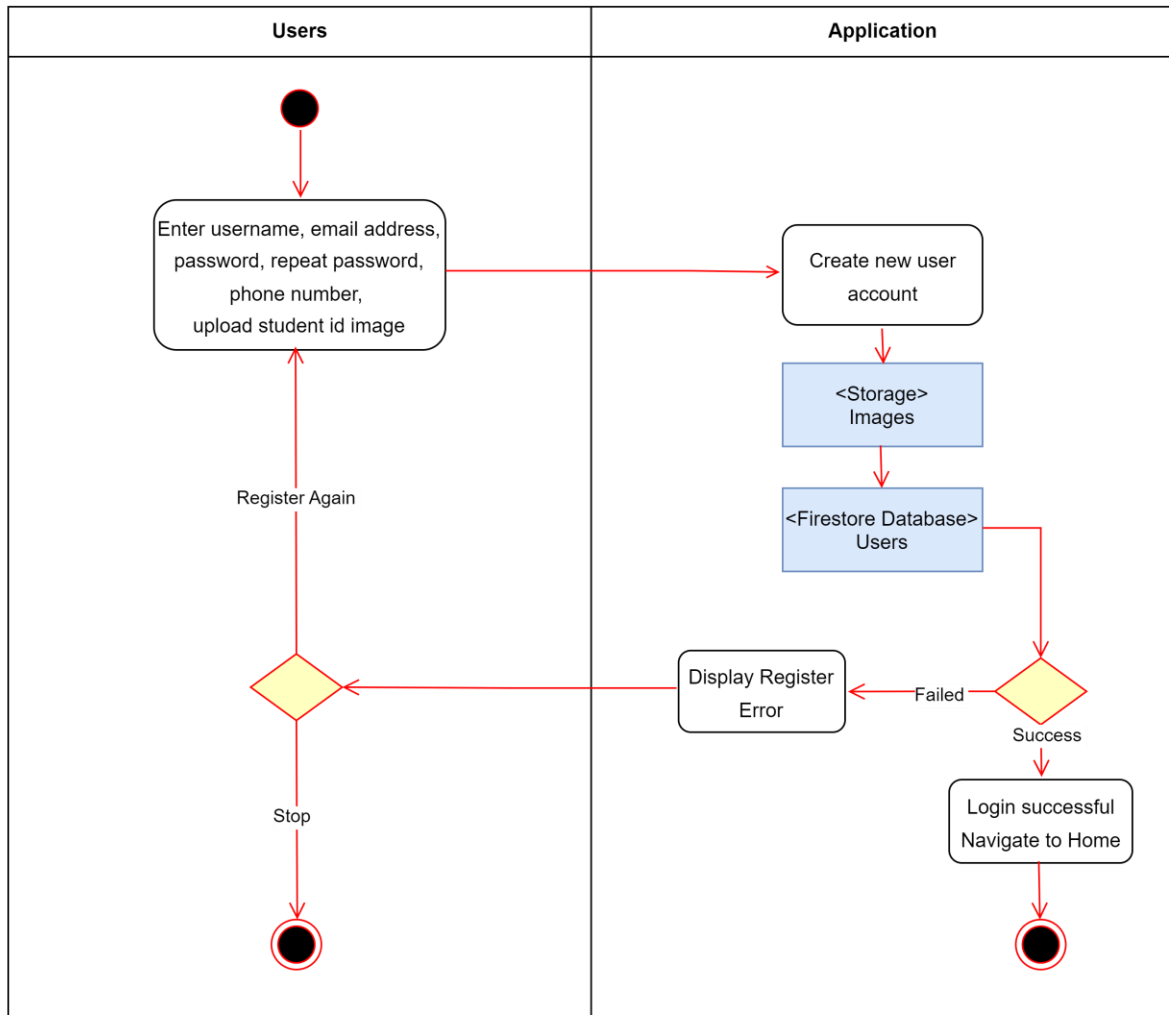


Figure 4.4 Activity Diagram for Users (Sign-Up)

Forgot Password

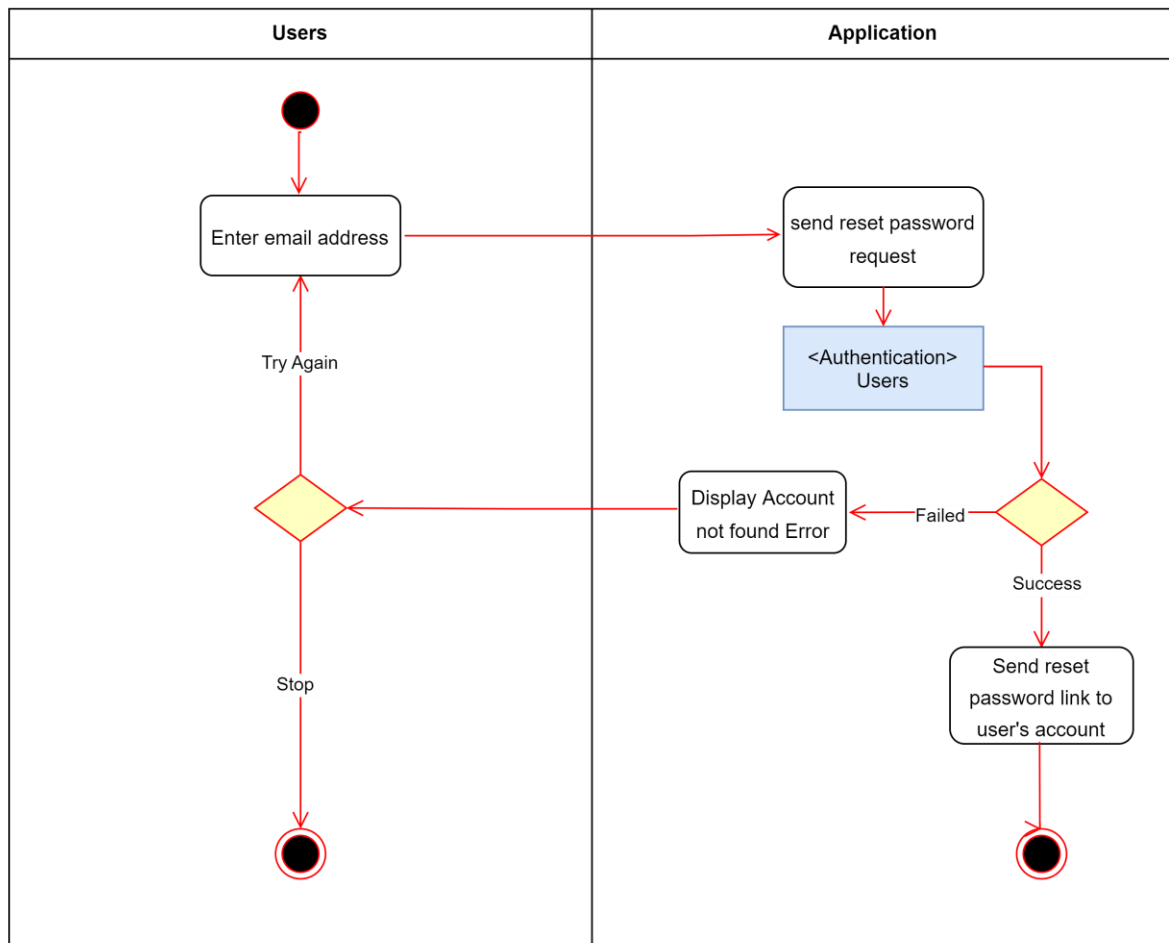


Figure 4.5 Activity Diagram for Users (Forgot Password)

Manage User Profile

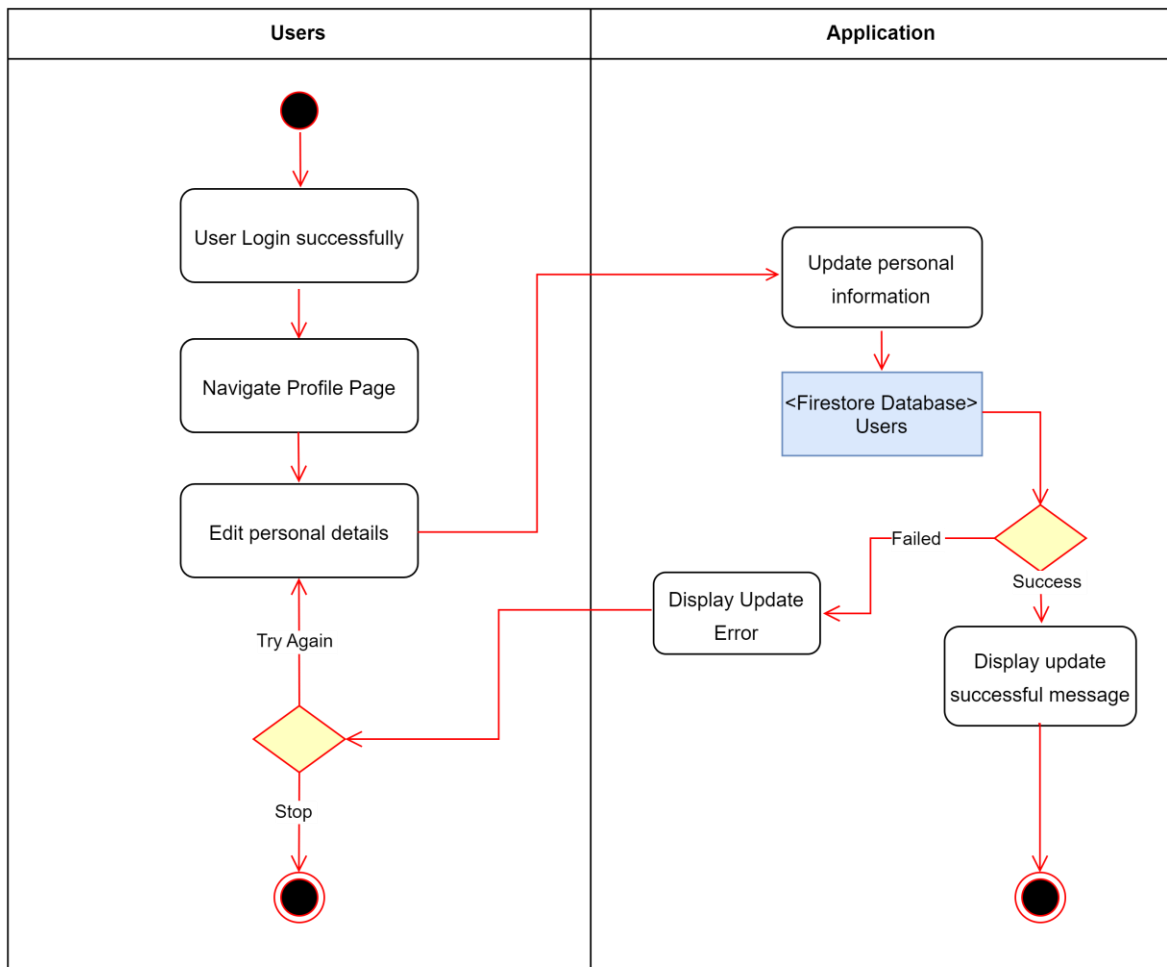


Figure 4.6 Activity Diagram for Users (Manage User Profile)

Check Rides Requested

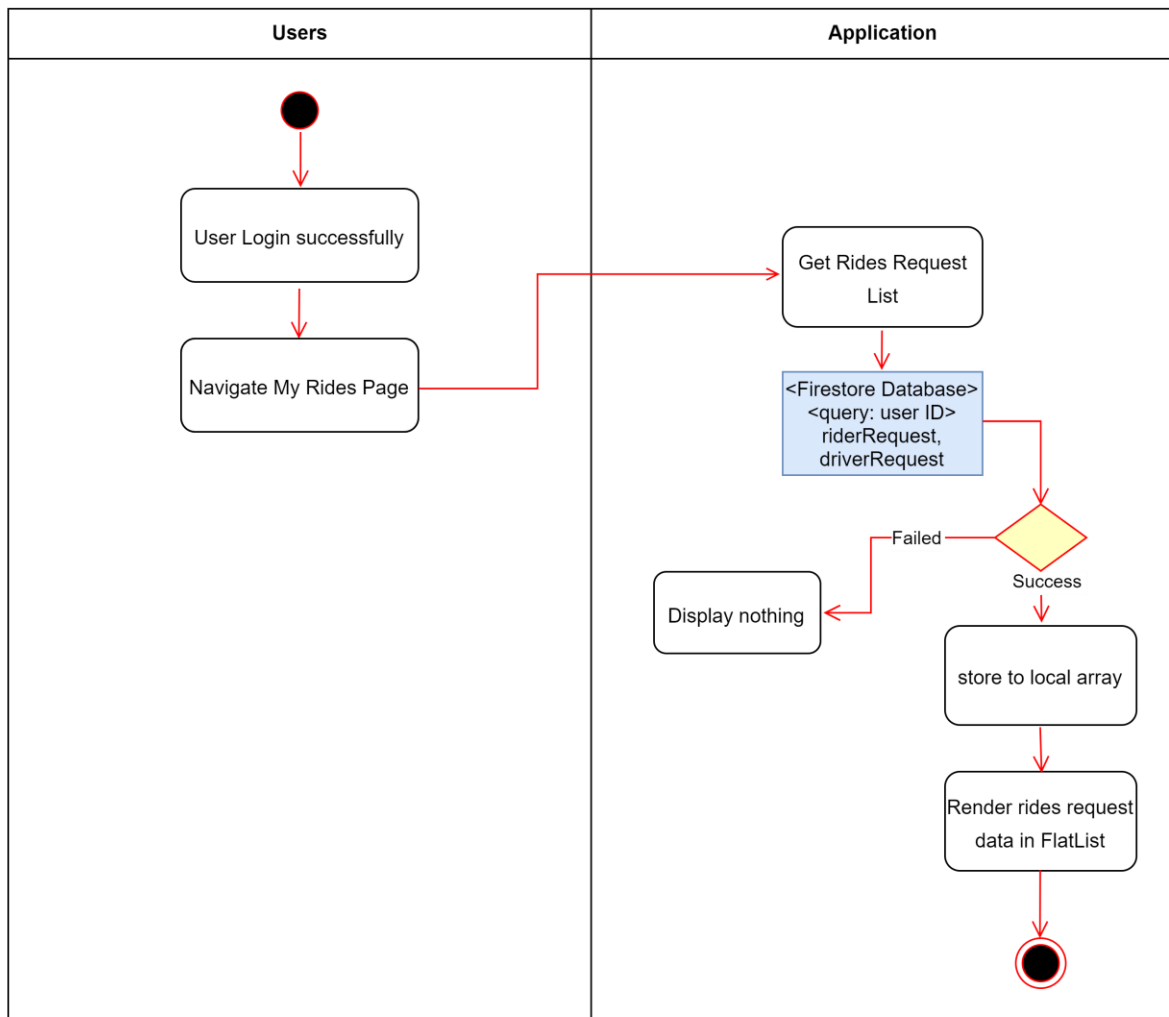


Figure 4.7 Activity Diagram for Users (Check Rides Requested)

Post Ride Request

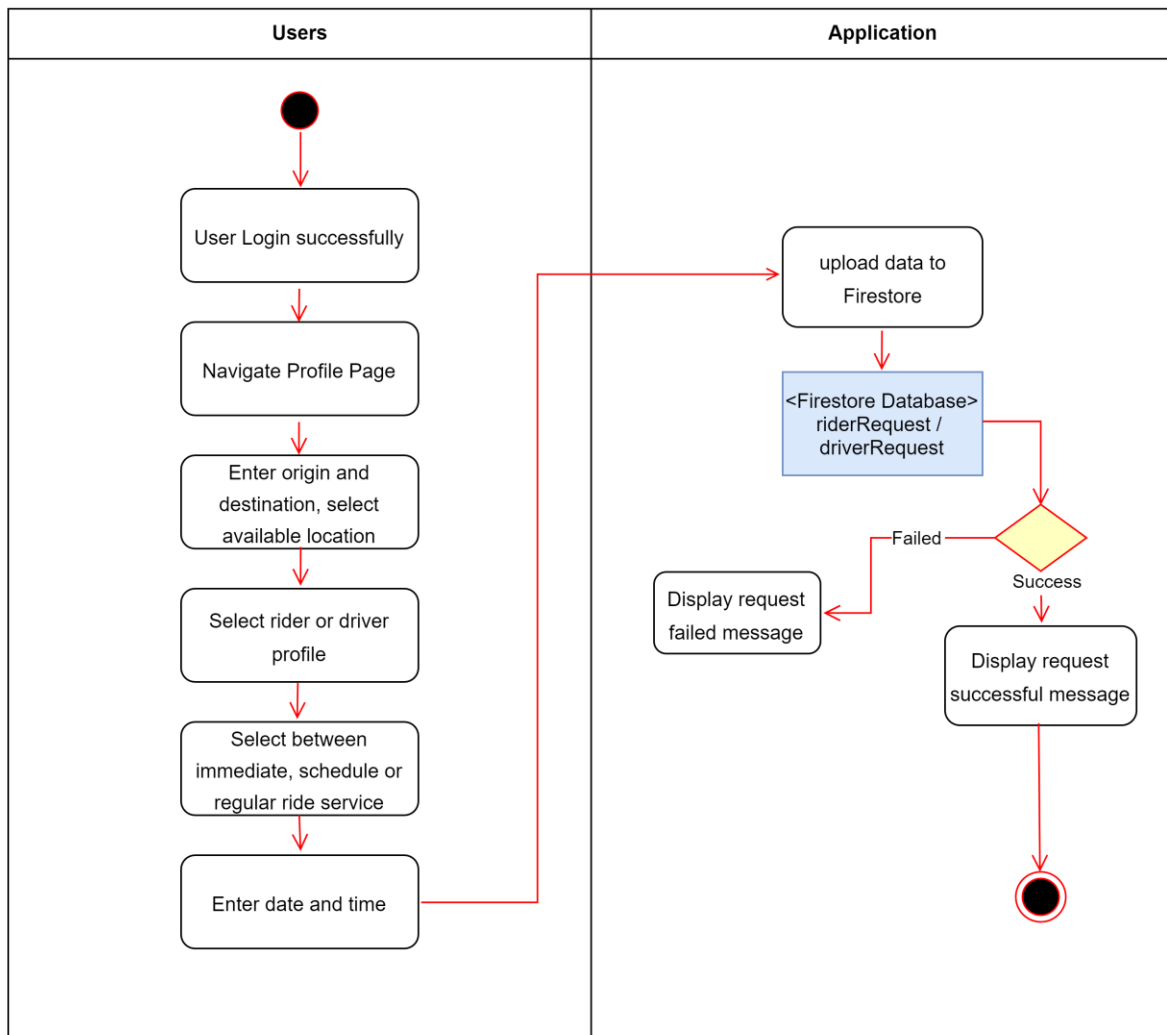


Figure 4.8 Activity Diagram for Users (Post Ride Request)

Verify User Account

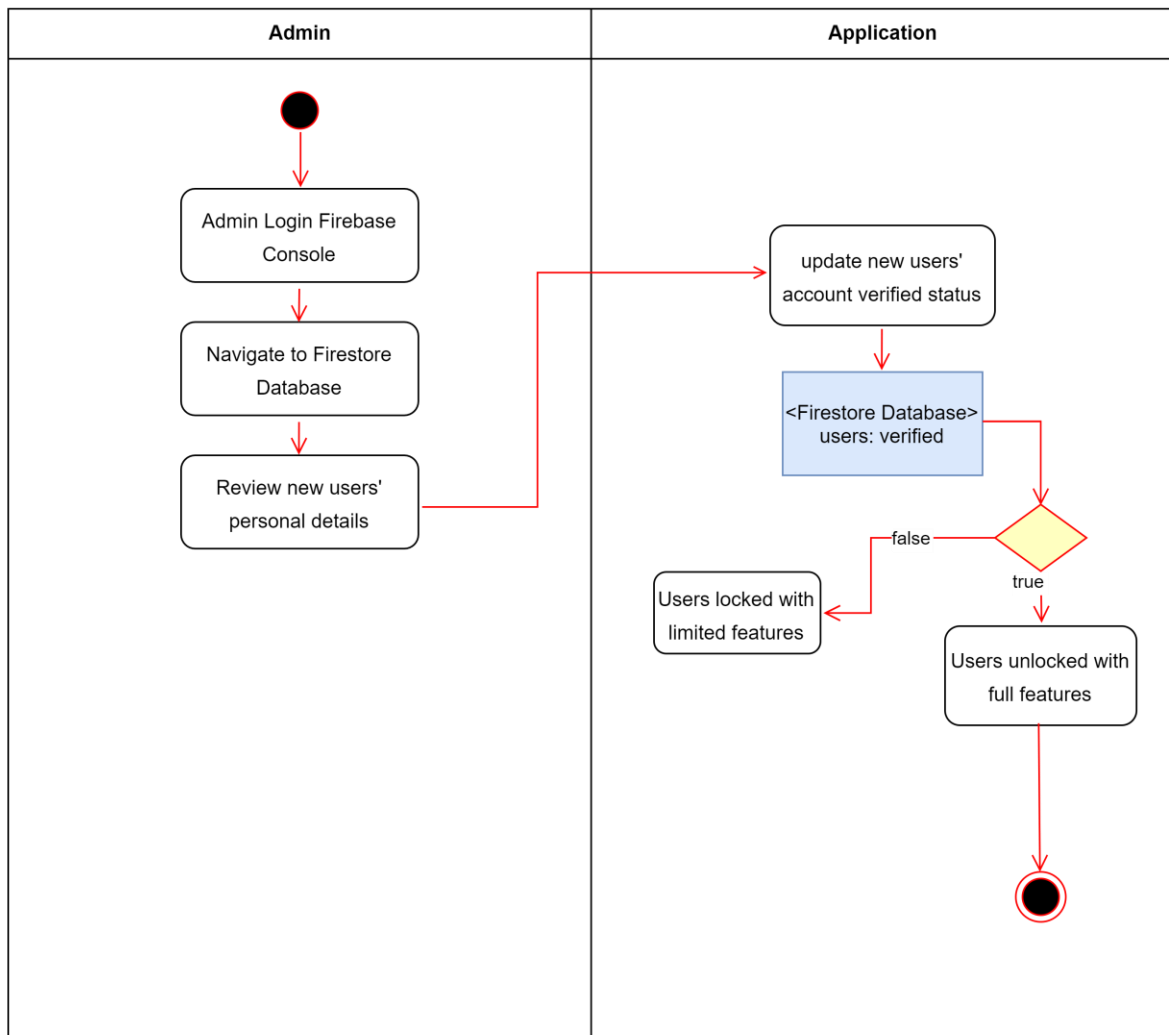


Figure 4.9 Activity Diagram for Admin (Verify User Account)

Set Home Address

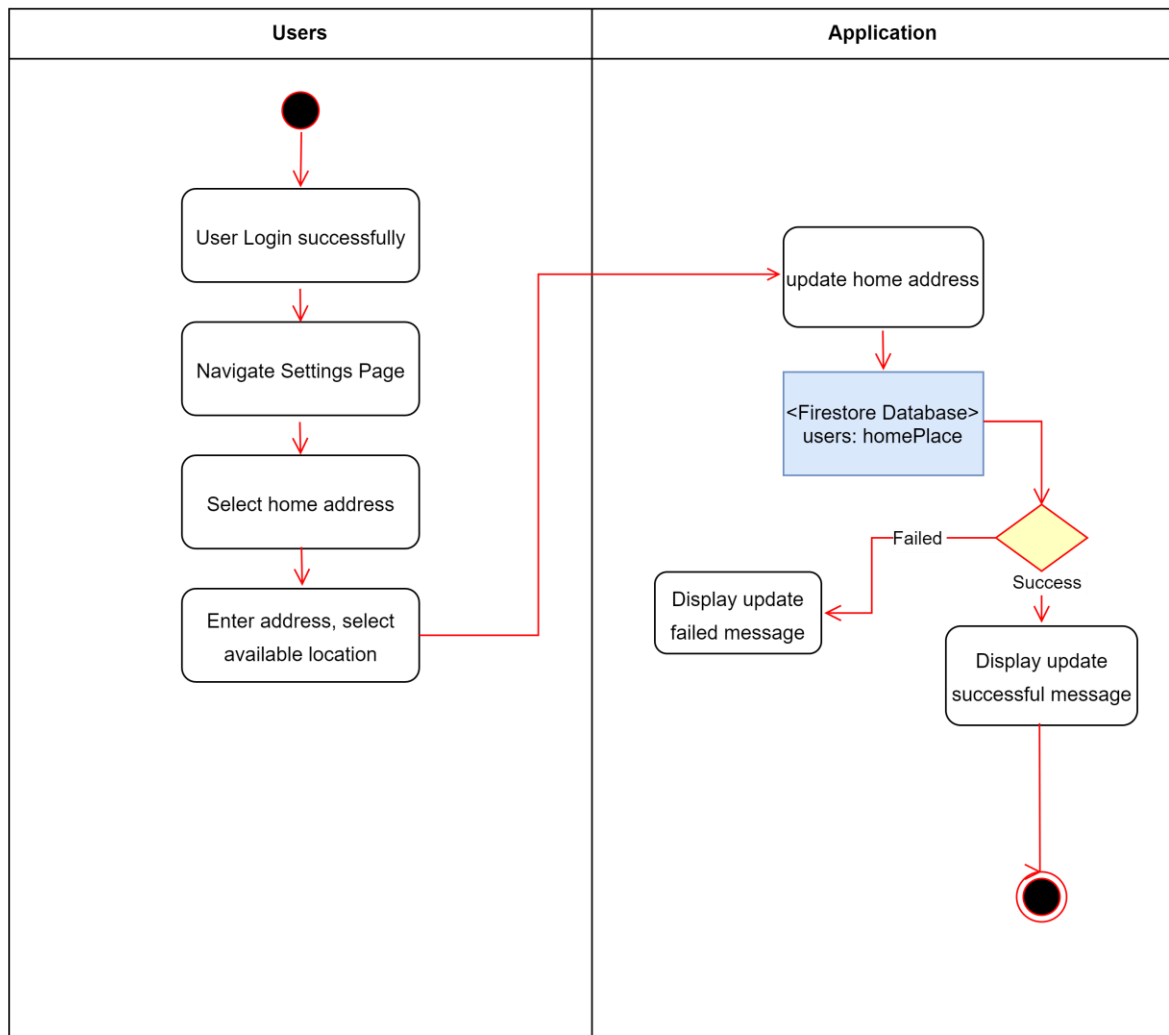


Figure 4.10 Activity Diagram for Users (Set Home Address)

Set School Address

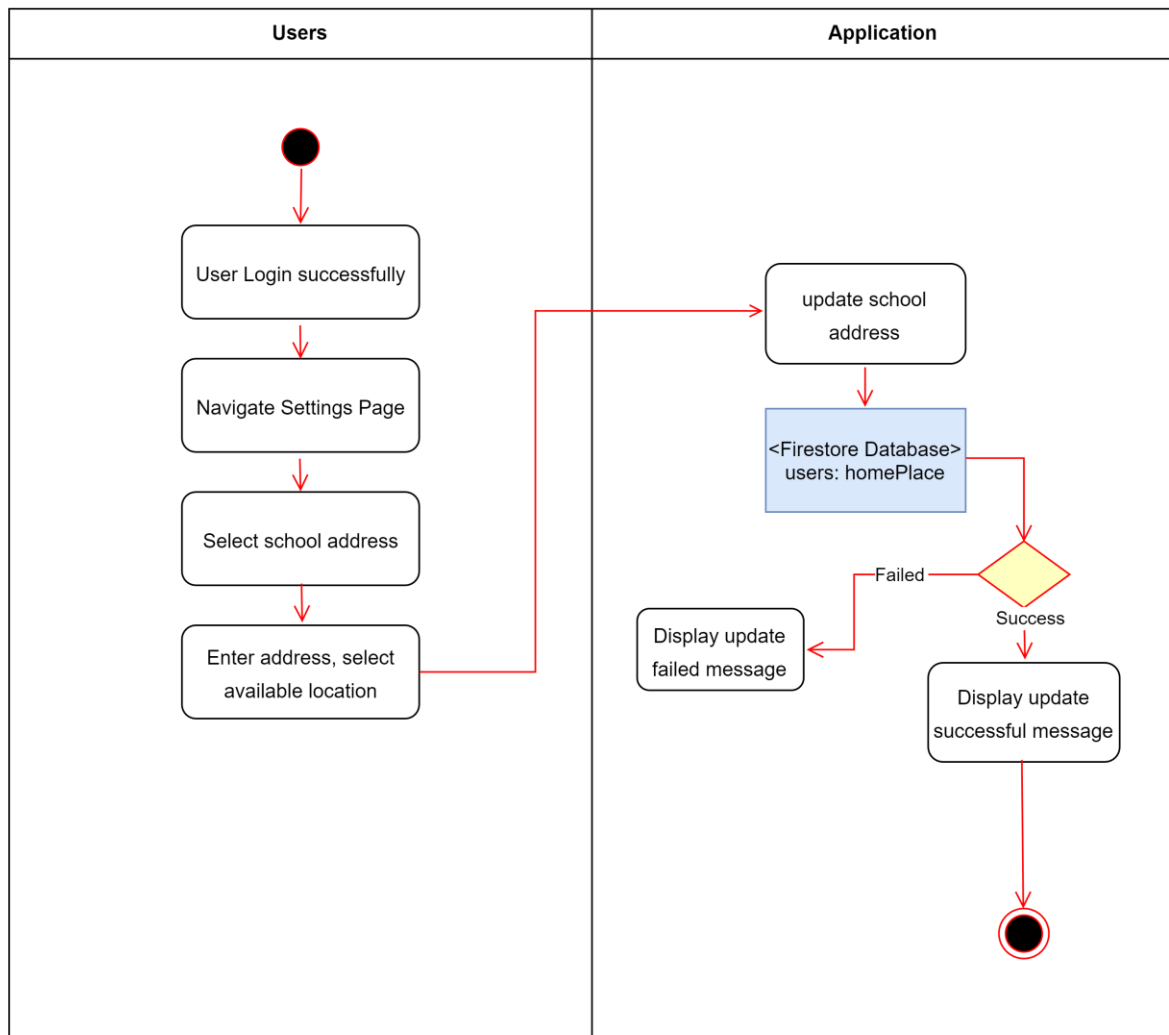


Figure 4.11 Activity Diagram for Users (Set School Address)

Request Ride Along

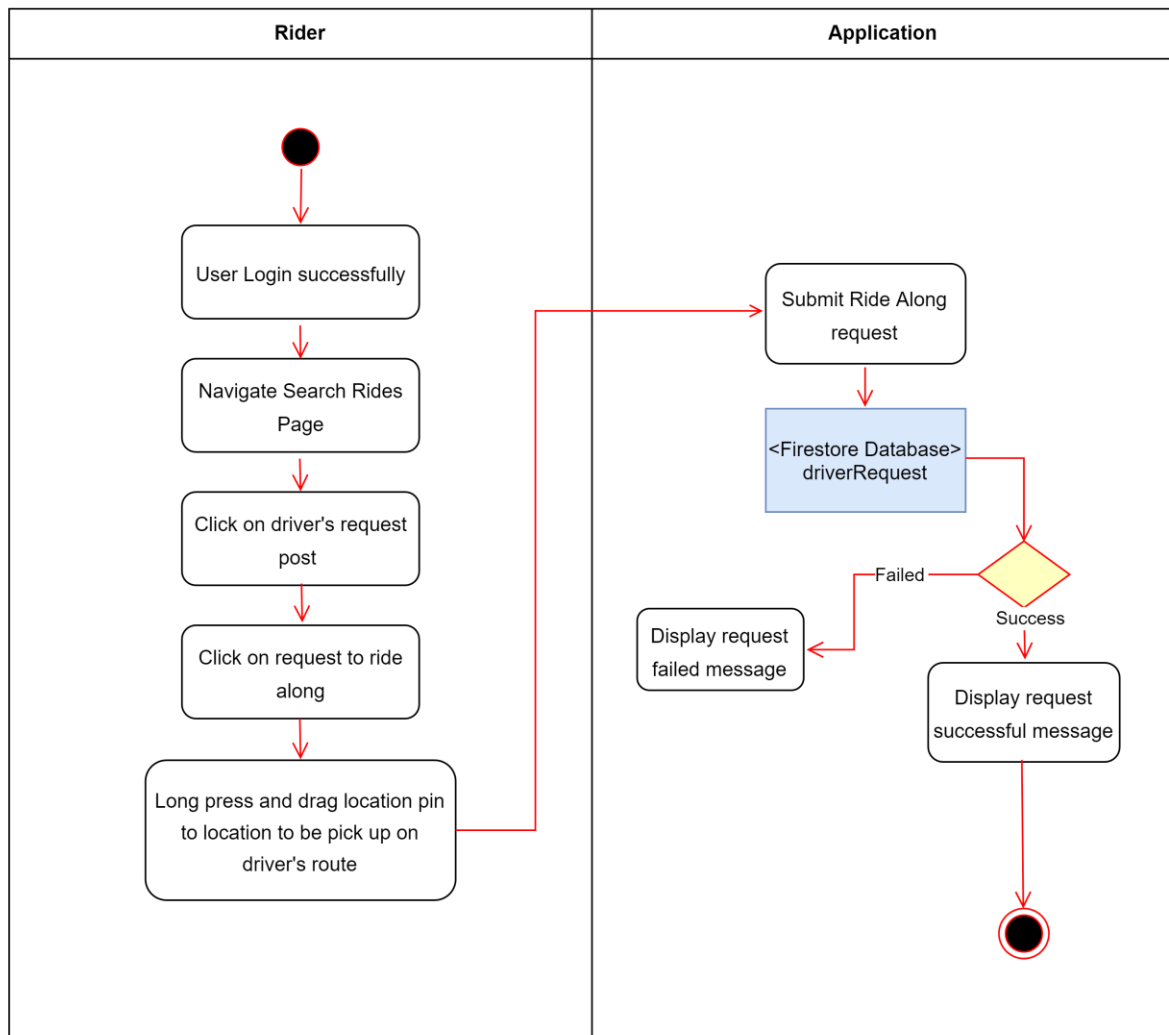


Figure 4.12 Activity Diagram for Rider (Request Ride Along)

Request Pick Up

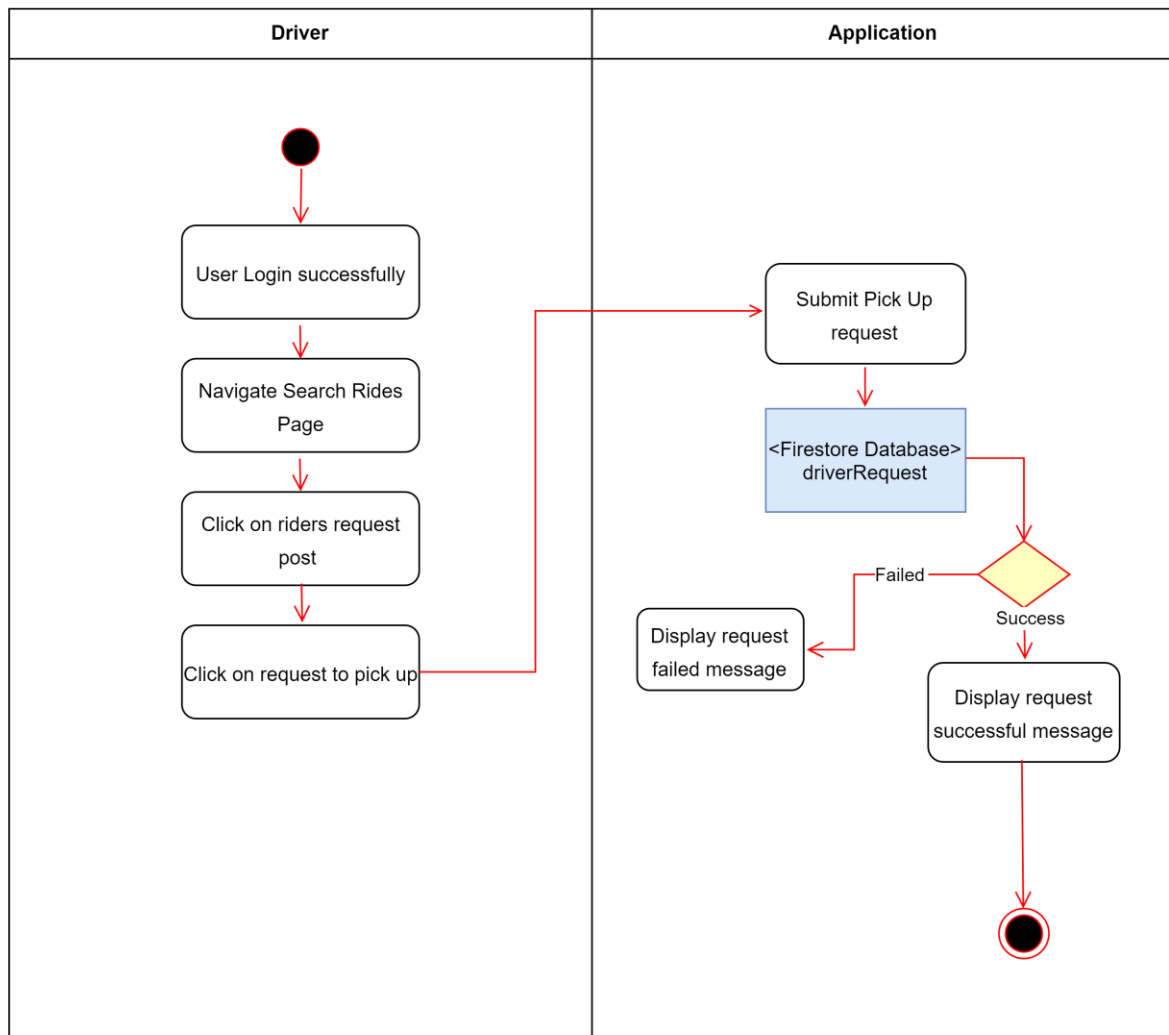


Figure 4.13 Activity Diagram for Driver (Request Pick Up)

4.1.4 System Flow Chart

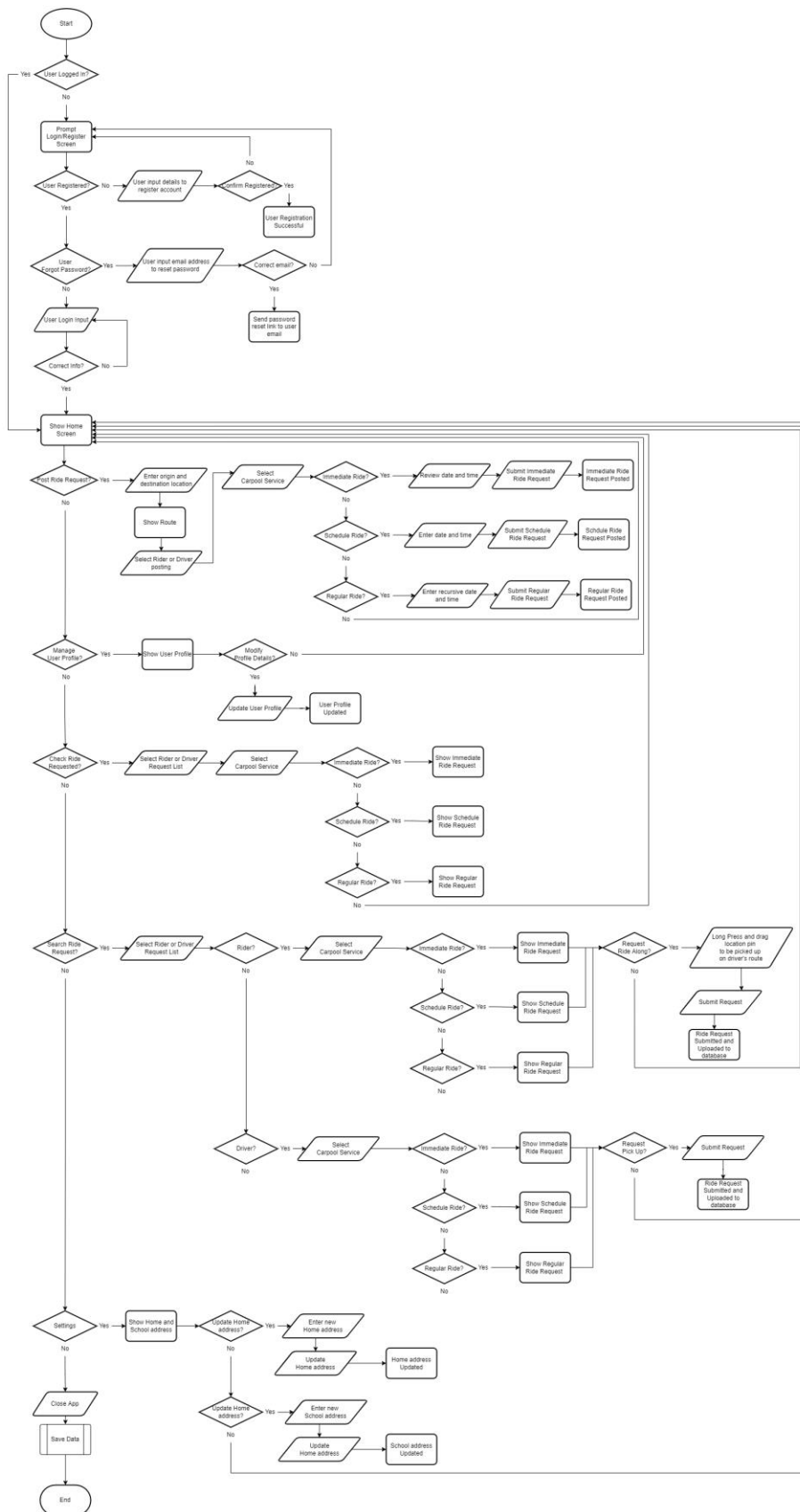


Figure 4.14 System Flow Chart

4.2 Technologies Specification

The technology and tools used for mobile development are critical and necessary for its success. Therefore, the chosen technology must be able to provide a good solution to implement the mobile application features that need to be implemented.

In this section, I will write about the technologies and tools that I plan to use in this project.

4.2.1 Framework

I. React Native

React Native is a popular JavaScript mobile application framework that uses JavaScript to create the interface of an application. React Native was first released by Facebook as an open-source project in 2015. In just a few years, it has become one of the top solutions used for mobile development. Popular mobile apps on the market such as Instagram, Facebook and Skype are created using the React Native framework.

Benefits

a) Code Reusability

The biggest advantage is it enables the creation of apps for various platforms by using the same code base. They can integrate 90% of the native framework for reusing the code for both Android and IOS platform.

b) Cost-Efficiency

It eliminates the needs for organization to hired two separate Android and IOS development teams for creating a project. The cost of development with React Native are much lower than other programming language that does not support cross platform development.

Therefore, React Native will be used in this project as a fundamental tool for building the mobile application. Through the characteristic of React Native, the application should be worked on both IOS and Android devices.

II. APIs

Node Package Manager (NPM) will be using in this project, NPM is a repository that has been around for ages and consist of lots of packages and libraries that is ready to be used.

a) React Functional Components

React functional component is a simple JavaScript function that accepts props and returns a React element. These components are common components that will be used when working with React.

b) React Navigation

React Navigation is a library that enable developers to implement functionality of navigation through different screens in a React Native application.

c) React Native Vector Icons

React Native Vector Icons is a library that contains over thousands of usable icons that can be directly implemented into a React Native application. There is also a website that shows all the available icons that ease for developers to search for their favorite or most suitable icons.

d) React Native Google Places Autocomplete

React Native Google Places Autocomplete is a library that can provide a standalone search function which can be automatically fills in the name or address of a place by user input based on keywords.

e) React Native Maps

React Native Maps provide library that can be directly implement in the application to show a map view. The map view is required to show the users the trip route and markers on the map to show specific point of places.

f) React Native Maps Directions

React Native Maps Directions allowed to draw a route between two coordinates which is powered by Google Maps Direction API. It allowed to show the routes between origin and destination of the user's trip.

g) Geocoding API

Geocoding API enable to retrieve the user current locations and show the current user locations on map. User may also quickly locate themselves on the map without the hassle of scrolling all over the map.

i) React Native Drawer Navigation

React Native Drawer Navigation is a user interface panel which displays the app's navigation menu. By default, it is hidden when not in use, but it appears when user swipes a finger from the edge of the screen.

j) React Native Image Picker

React Native Image Picker allow user to select photo/video from the local device library. This component is used for allowing users to upload their student id during the registration process.

k) React Native Date Time Picker

React native Date Time Picker is a component that provide access to the native UI for date and time selection. This component is used for users to select date and time for ride request posting.

l) React Native Splash Screen

React Native Splash Screen is a component allow application to display a splash screen which also known as a launch screen, before user opening into the application, it stay visible while the application is loading.

III.Backends

a) Firebase

Still a multi-node, key-value database optimized for synchronizing data between a user's desktop or smartphone and centralized cloud storage, the Real-Time Database is the most well-known feature of Firebase's cloud-based development platform. It is designed to make developers' jobs easier by handling large amounts of data pushes and pulls for them. This eliminates the coding issues associated with managing the versioning and location of applications. They can update bits in Firebase to ensure that data is consistent across the system.

The Real-Time Database is a NoSQL database hosted in the cloud that allows users to store and synchronize data in real time. This data is synchronized in real time between all customers and remains accessible when the application is offline. It is used to hold all user data collected in the application when utilizing the carpooling service.

The authentication feature simplifies the development of secure authentication systems for developers, while improving the user login and onboarding experience. The feature provides a complete authentication solution, including email and password accounts, phone authentication, and Google, Facebook, and Twitter logins. As such, it will be used to provide authentication to all users of the application's carpooling service.

The storage feature also been used in this project to store all the student id images uploaded by all the users. The images will then publish as a download link and store in the Firestore database.

b) Google Maps API

The Google Maps Platform is a set of APIs and SDKs that developers can use to integrate Google Maps into mobile applications and websites, and to retrieve data from Google Maps. There are multiple options. Developers can use one or a combination of APIs and SDKs depending on their requirements. The Google Maps API will be used to retrieve the user's location, configure directions for carpool travel, and show the user the route and pick-up location.

c) Node.js

Node.js is a cross-platform, open source back-end JavaScript runtime environment for executing JavaScript code outside of the web browser. It is also used to run the React Native application in this project.

4.2.2 Hardware Specification

I. Desktop Simulator

During the whole development of the project, I was using Android Studio's simulator feature to do the debugging. Due to the limitation of Xcode which is a desktop app provide IOS simulation is limited to Apple devices, while I am using the window operating system, the development and outcome will be only limited to Android device compatible. The simulator device I have been using is Pixel 3a with android 10.0 operating system and play store integrated.

II. Physical Device

During the deployment, I had tried to export the whole project into APK to do the application testing on the physical device. The physical device I have been using is Oppo A16, with 4GB RAM and 8 cores CPU processor, which is good enough to do the application debugging.

Chapter 5

System Implementation

In this chapter we present the hardware setup, software setup, system operation, implementation issues and challenges in this project.

5.1 Hardware Setup

5.1.1 Desktop Debugging

For desktop debugging, I am using Android Studio's integrated virtual device to do all the debugging process on desktop.

The virtual device that I have created on Android Studio is Pixel 3a, with Android 10.0 operating system and Google Play services integrated.

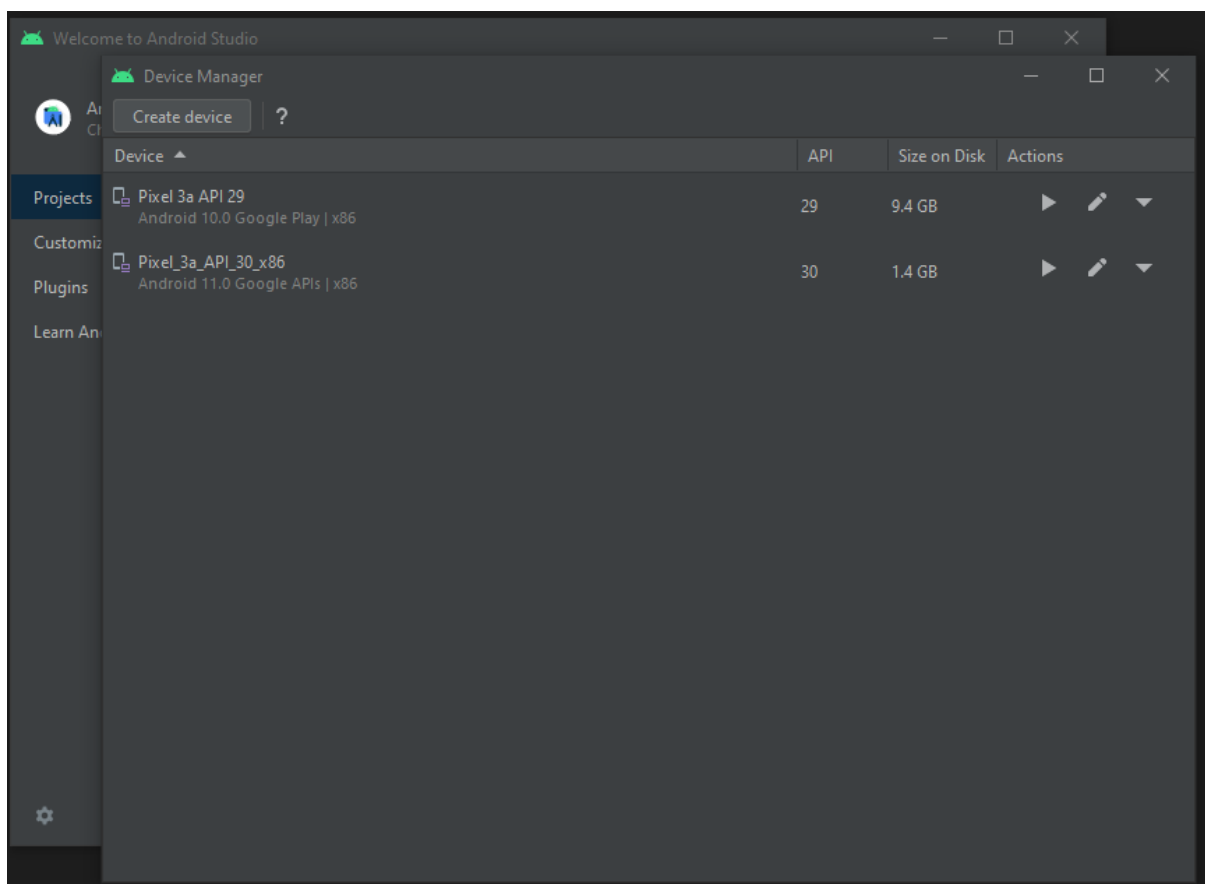


Figure 5.1 Android Studio Virtual Device

The application has successfully installed and running on the android virtual device.

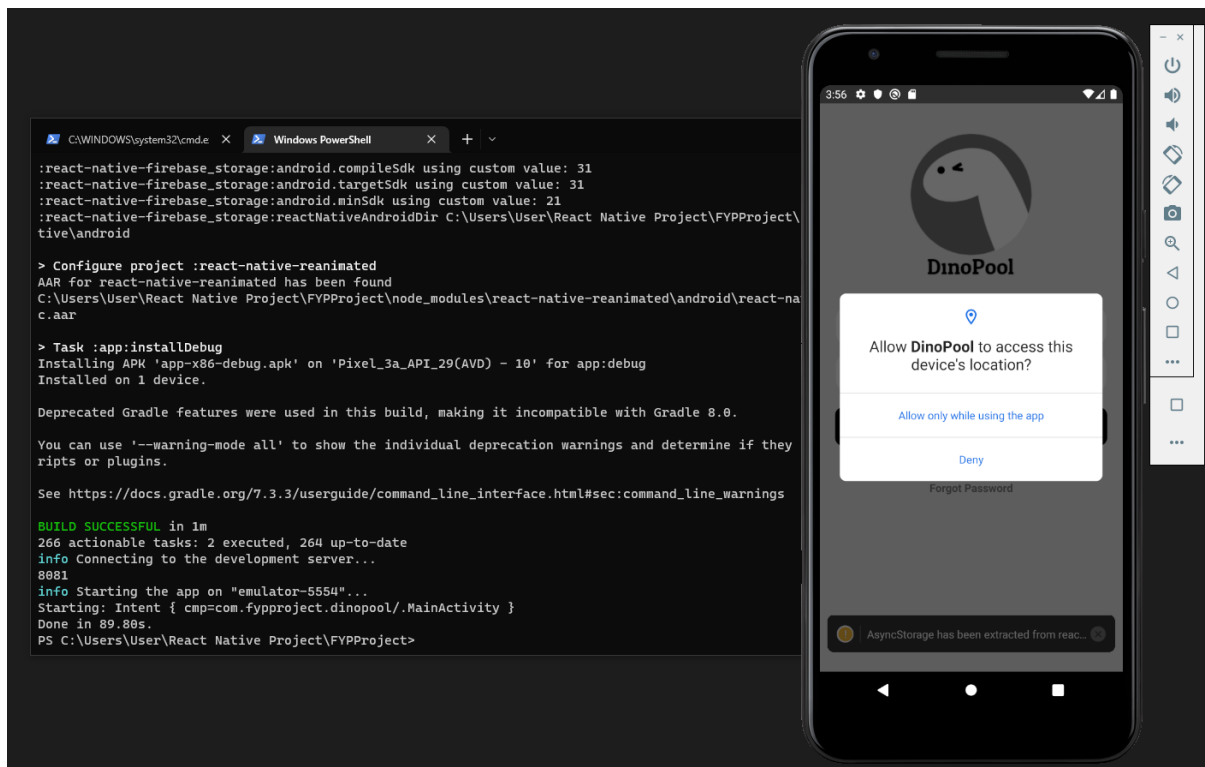


Figure 5.4 Window Terminal (run android)

5.1.2 Physical Android Device Debugging

Sometimes debugging using virtual device might not be efficient to discover any application bugs or defect, Therefore, I will be doing a physical debugging once I finish implementing all the functions and APIs.

For physical Android device debugging, I am using Oppo A16 android device with 4GB RAM and 8 cores CPU which is more than enough to debug the application.

To debug on physical device, I will generate a APK file to install it into the physical Android device. I will be using the “app-universal-release.apk” file to do the physical device debug.

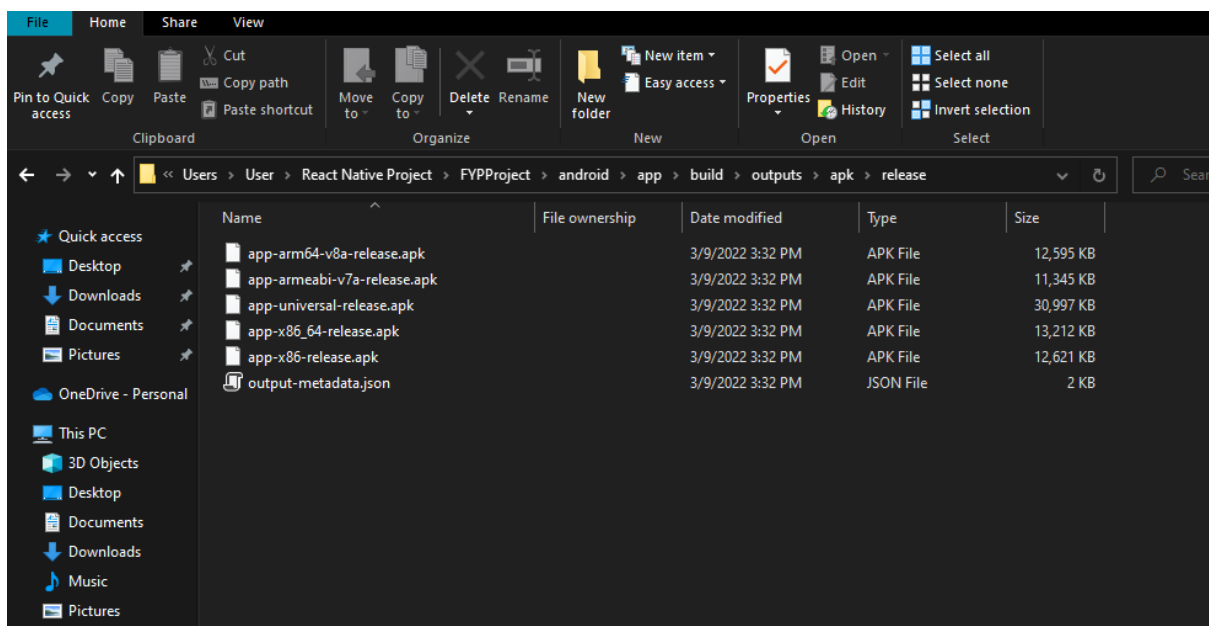


Figure 5.5 Project APK files

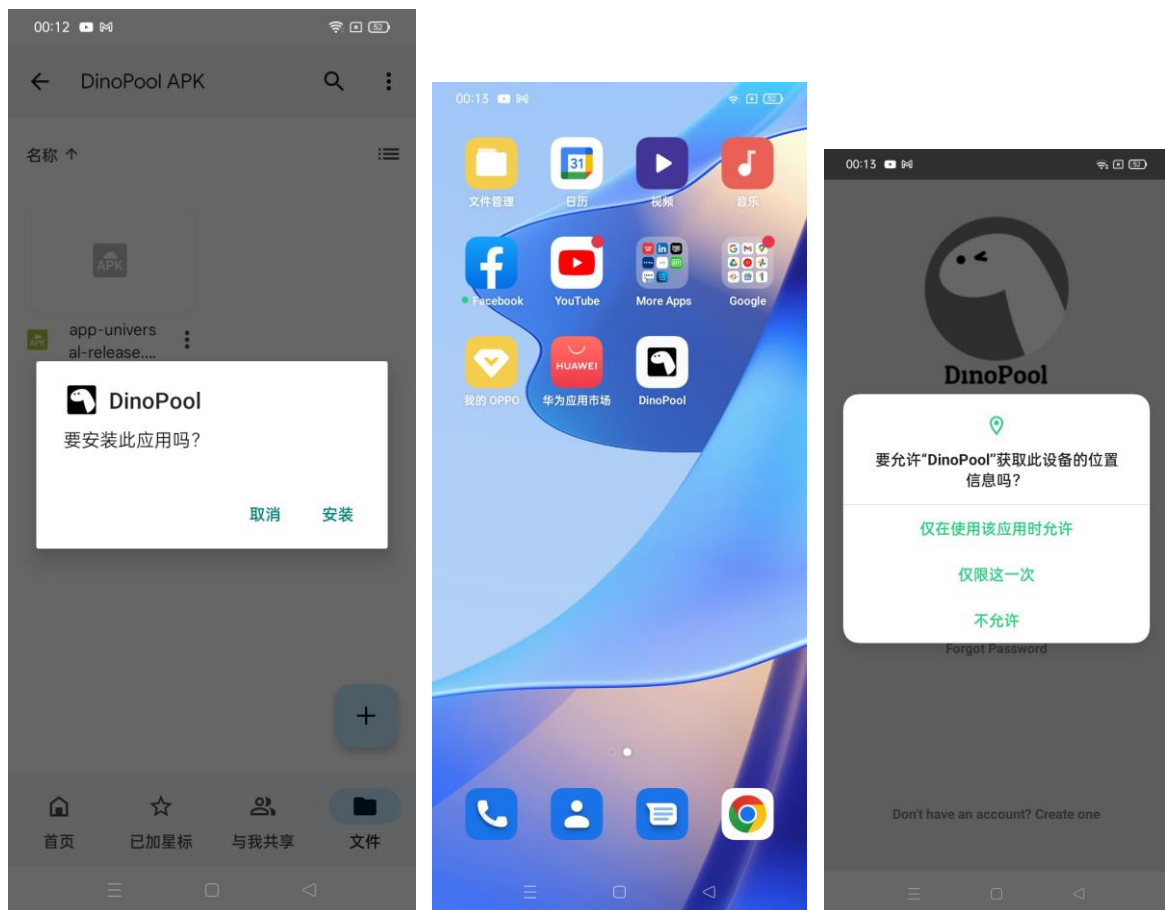


Figure 5.6 Physical Device APK installed

5.2 Software Setup

5.2.1 React Native environment setup on window machine

React Native require some software installation and configuration to start the development.

Below are the steps to set up the react native development environment.

- I. Install Window Terminal
- II. Install Git
- III. Install Node.js
- IV. Install NPM package manager
- V. Install React Native CLI
- VI. Install Android Studio (for desktop debug)
- VII. Install Visual Studio Code (IDE)

5.2.2 Export release APKs and bundles

To generate APKs and bundle release for the application, the below command are being used to achieve that.

Bundle Release (AAB file)

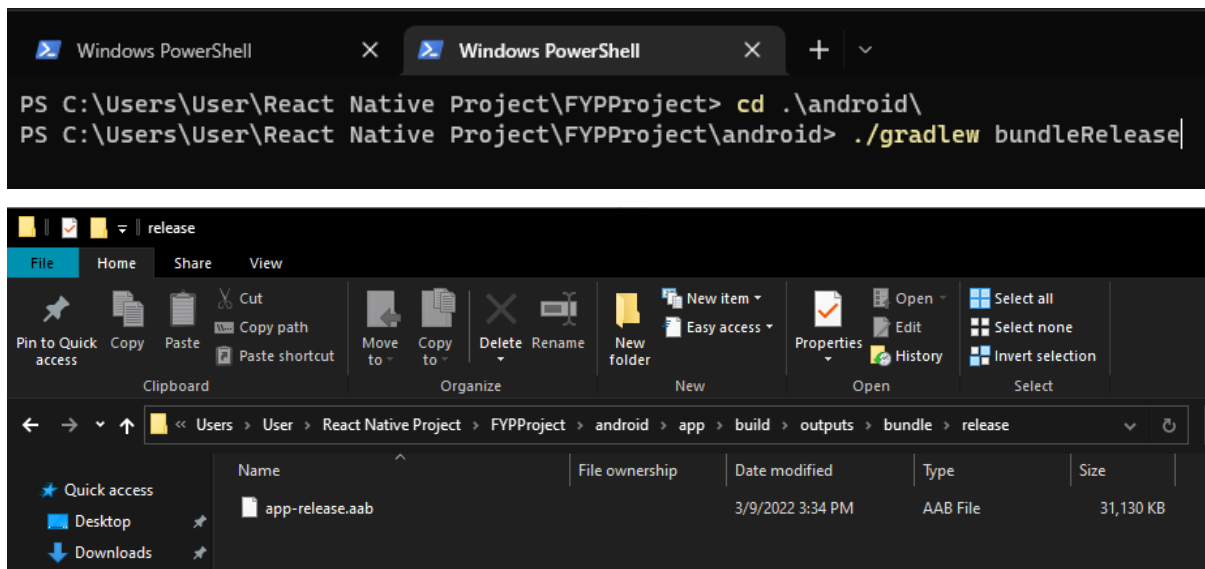


Figure 5.7 Project Bundle Release

Assemble Release (APK files)

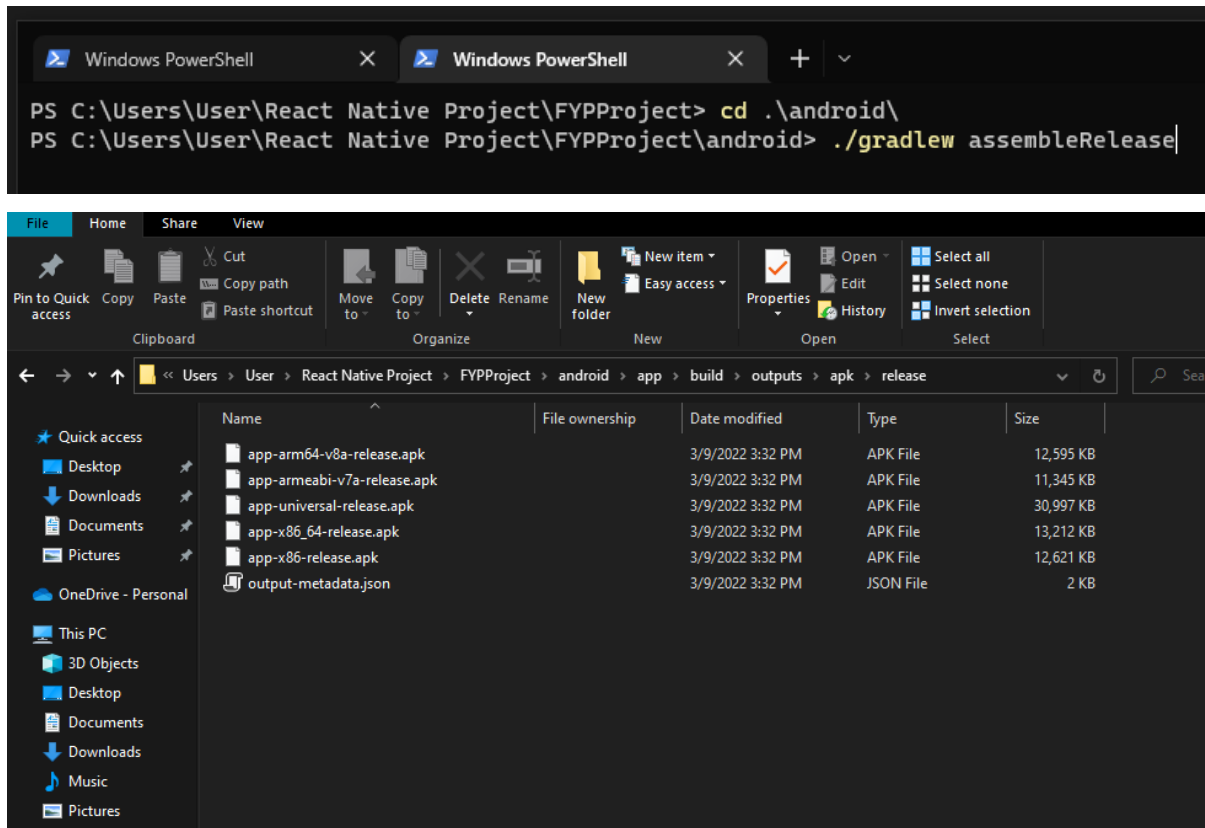


Figure 5.8 Project Assemble Release (APK files)

5.2.3 Published to Huawei App Gallery

To allow the application can be reached to more audience, I have also applied to publish the application to Huawei App Gallery as it is free to publish app.

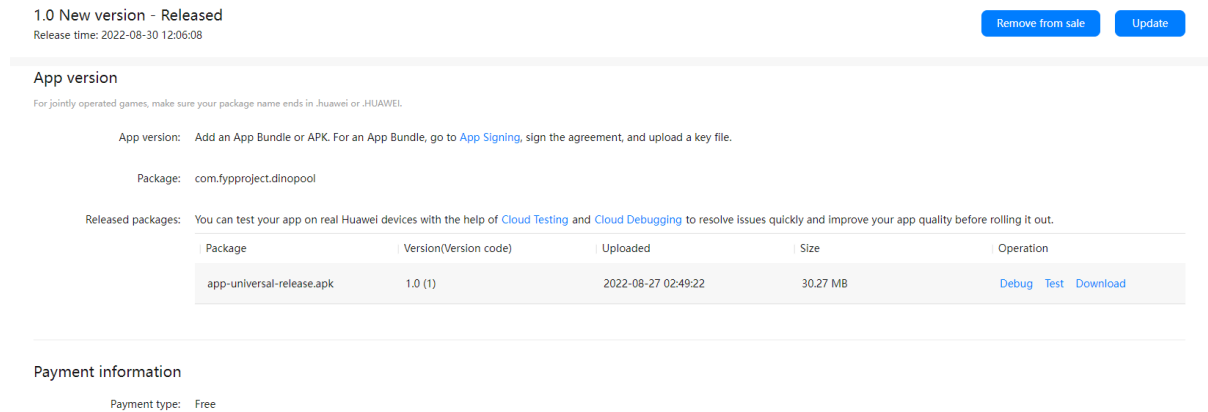


Figure 5.9 Huawei App Gallery Console (Publish App)

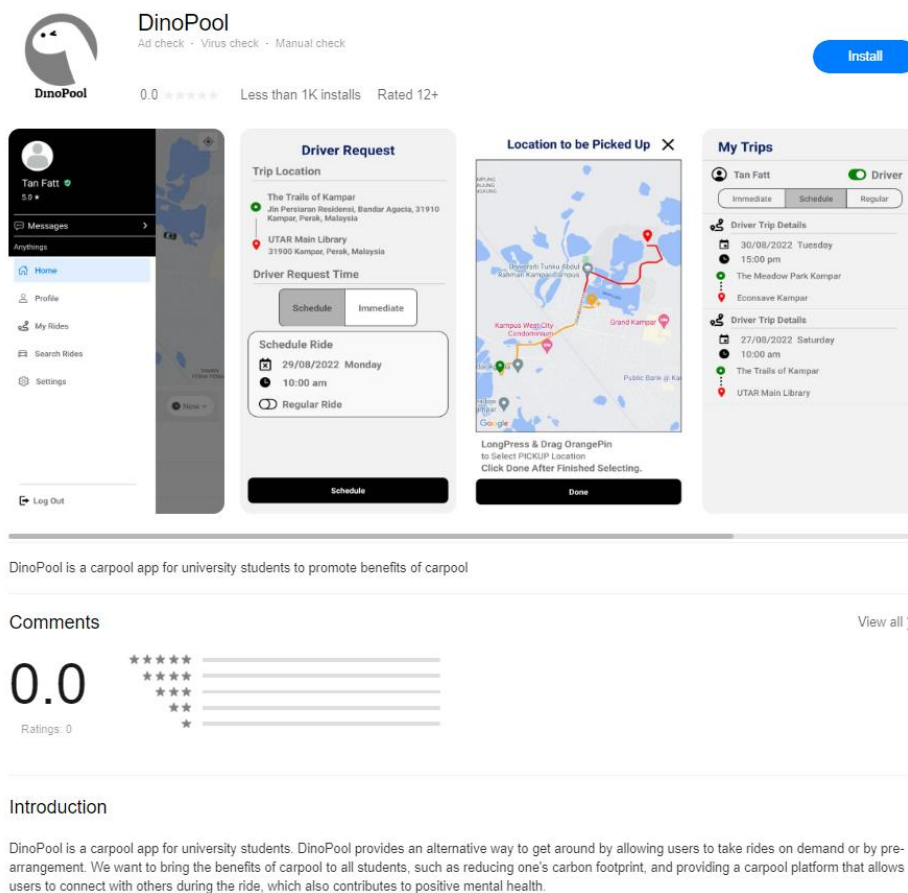


Figure 5.10 Huawei App Gallery (DinoPool)

5.3 System Operation (with Screenshot)

5.3.1 Introduction

The application will have an admin to manage all the users' accounts and users which can further to categories into two types of users which are Rider and Driver. Both types of the users will have similar interface but slightly different features in term of ride request services. Besides, users with verified account will be unlocked with full features of the application, while users with account that is not been verified will have limited features in term of functionality. The table below shows the users and the module associated:

| User | Modules/Features |
|----------------------|---|
| Admin | Firestore Console to review and verified users' account and profile. |
| Users (verified) | Login Forgot Password Sign Up Post Ride Requests Manage Profile Check Requested Rides Search Rides Request Ride Along Request Pick Up Set Home Address Set School Address |
| Users (not verified) | Post Ride Requests Request Ride Along Request Pick Up |
| Users (Rider) | Request Ride Along |
| Users (Driver) | Request Pick Up |

Table 5.1 Modules/Features for user

5.3.2 Module/Features for Admin

Admin will have the database account that associated with the application. Admin can login into Firebase account and using the Firebase console to manage all the users account and review on personal details.

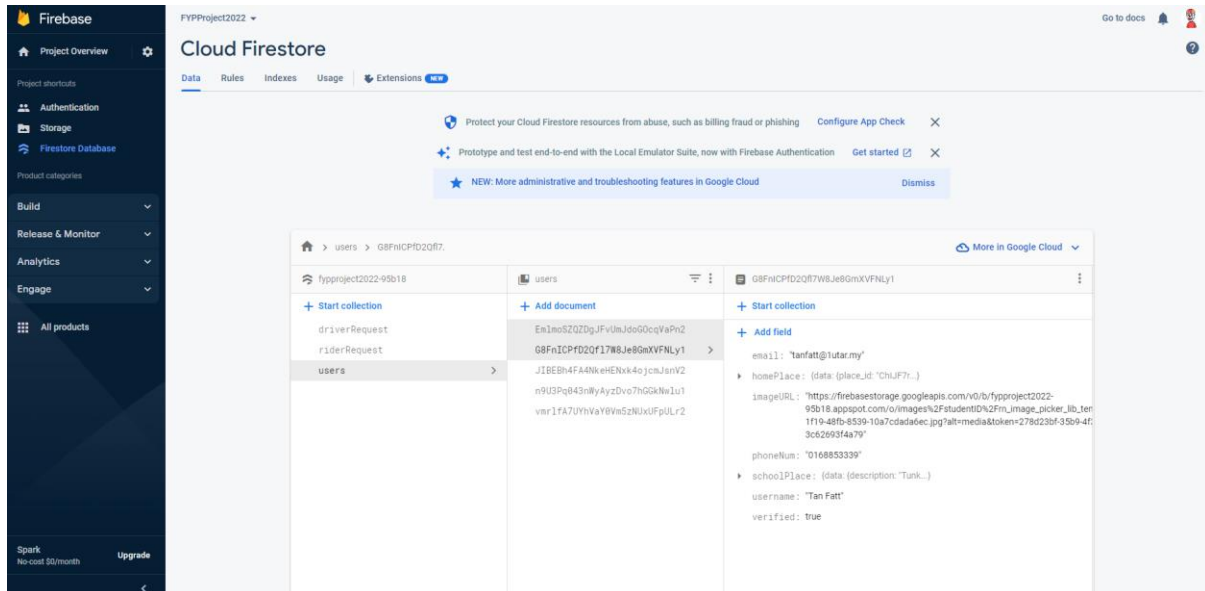


Figure 5.11 Firebase Console (users)

Admin can use the imageURL to review on the users' uploaded student id image.

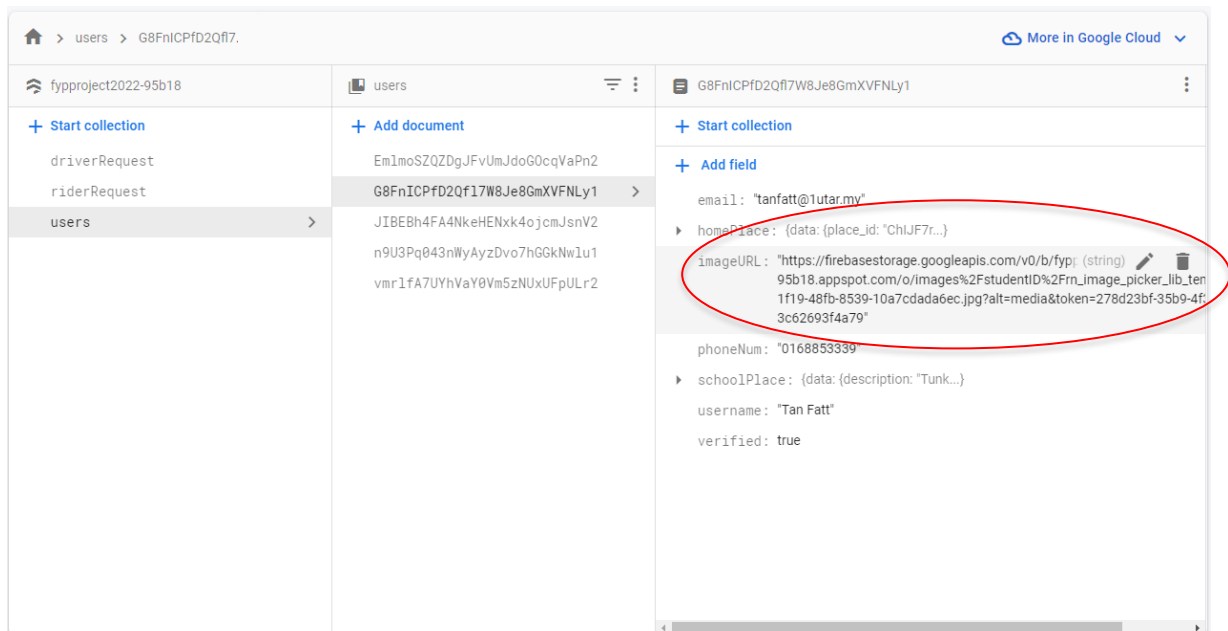


Figure 5.12 Firebase Console (users - imageURL)

Chapter 5 System Implementation

From there, admin can review the user account details and decide whether to verify the user account.

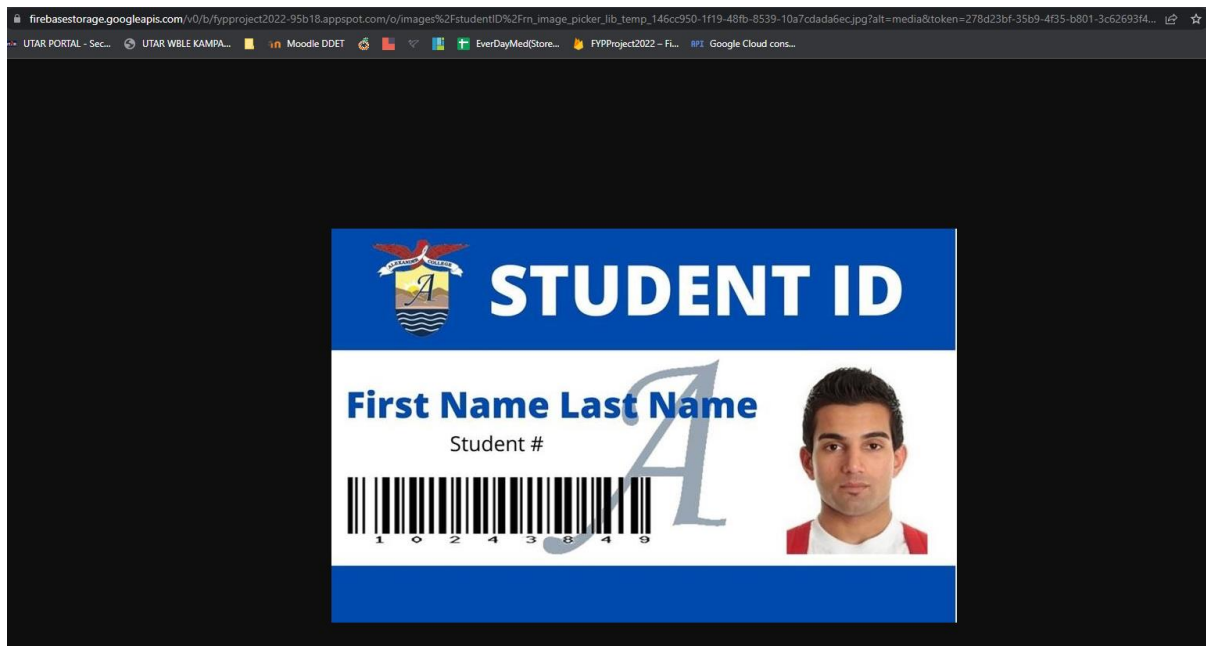


Figure 5.13 Firebase Console (imageUrl)

If the admin decides to verify the user account, admin can update the verified value to be true to unlocked full feature to the user.

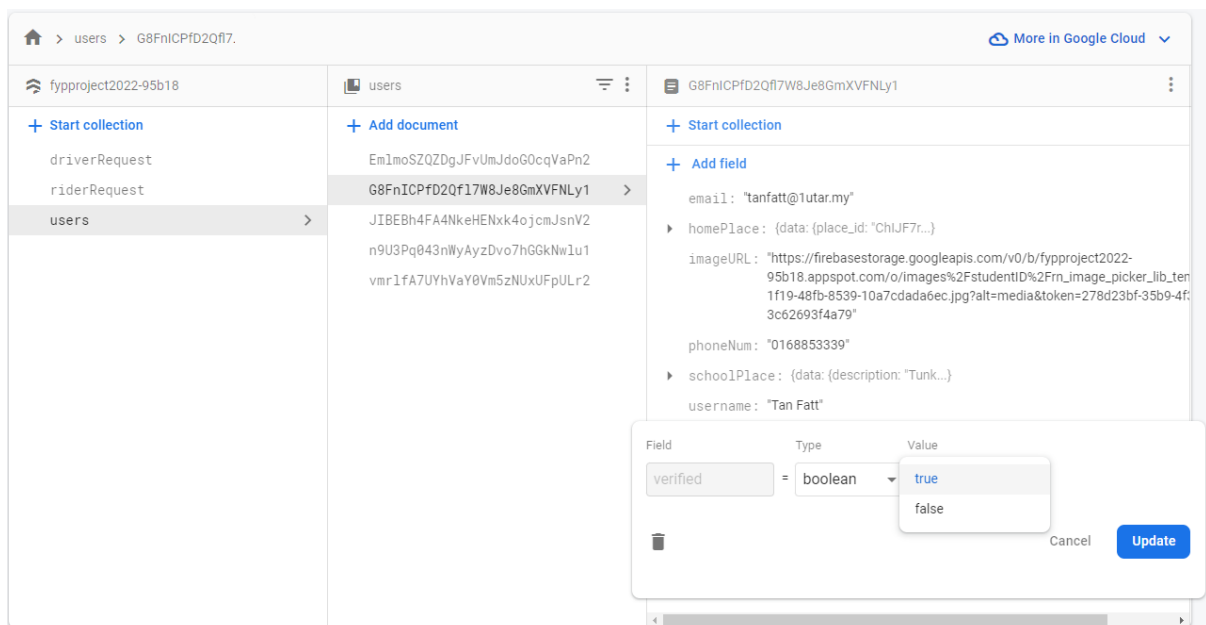


Figure 5.14 Firebase Console (verified status)

5.3.3 Application Features

5.3.3.1 Splash Screen

When user launched the application, a splash screen will display to the user to informed user what application they are using and to provide better user experience.

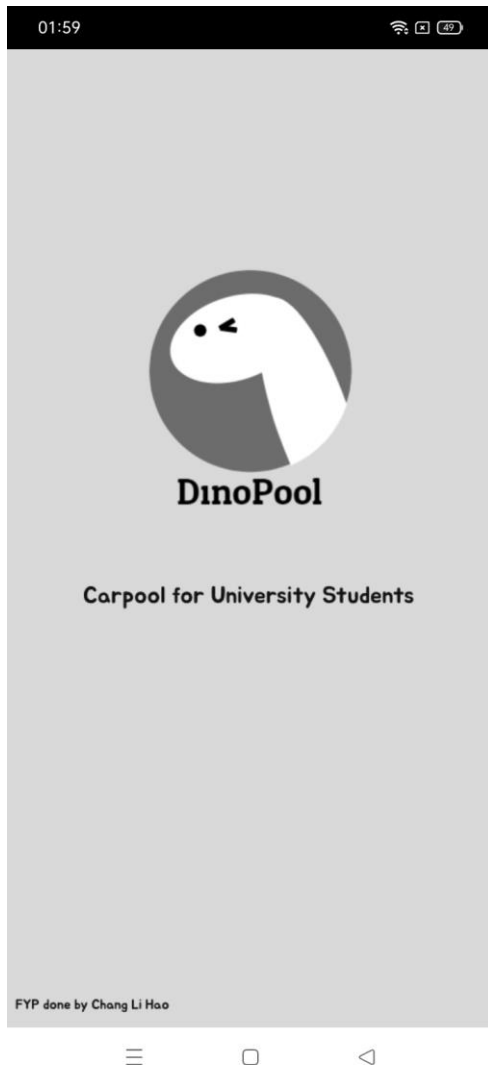


Figure 5.15 Splash Screen (DinoPool)

5.3.3.2 Location Permission

When user first launched the application after installed, an alert box will prompt to ask users whether they allow the application to access their device location or not.

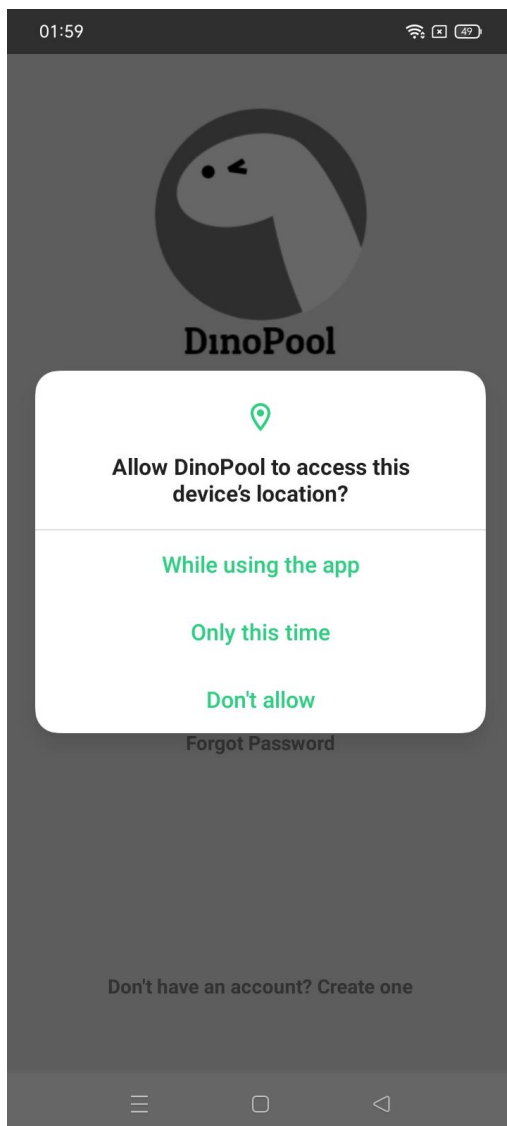


Figure 5.16 Location (DinoPool)

5.3.3.3 Drawer Menu

If user have successfully login into the application, user can swipe a finger from the left edge of the screen and a Drawer Menu will appear. The Drawer Menu will show current username and a small user verified status icon on the right side of the username. If the user has been verified, the icon will be in green color. The Drawer Menu also include all the navigation to all the different modules of the application such as Home, Profile, My Rides, Search Rides, and Settings. Different modules provide different functionality and features. At the bottom of the Drawer Menu will have a log out button for user to be logged out.

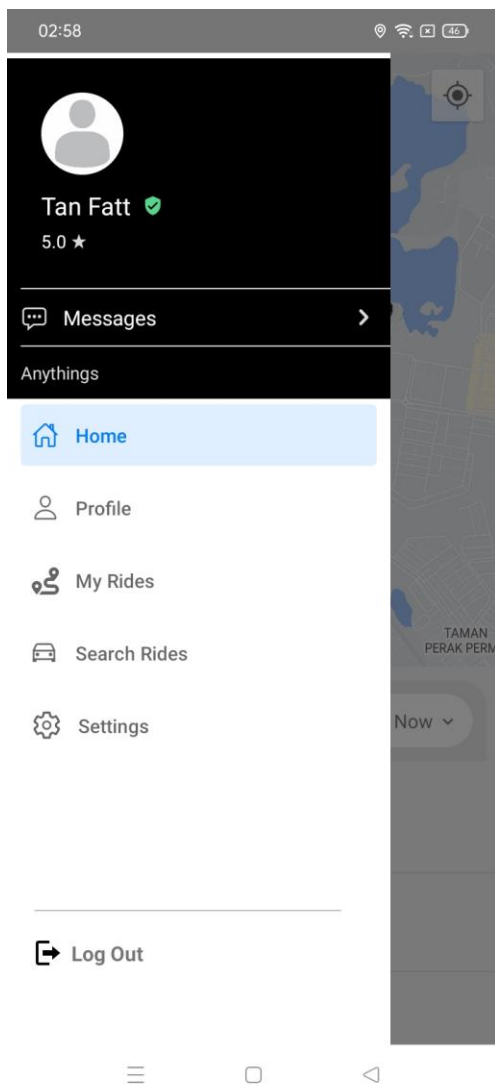


Figure 5.17 Drawer Menu (DinoPool)

5.3.3.4 Home Map

The home screen will have a map view, which user will immediately see when they are logging into the application. The map will access to user's Geolocation to track on user's current location, user can press on the locate button on the top right corner to quickly locate themselves. User can also further interact with the map view by zoom in or out or drag around.

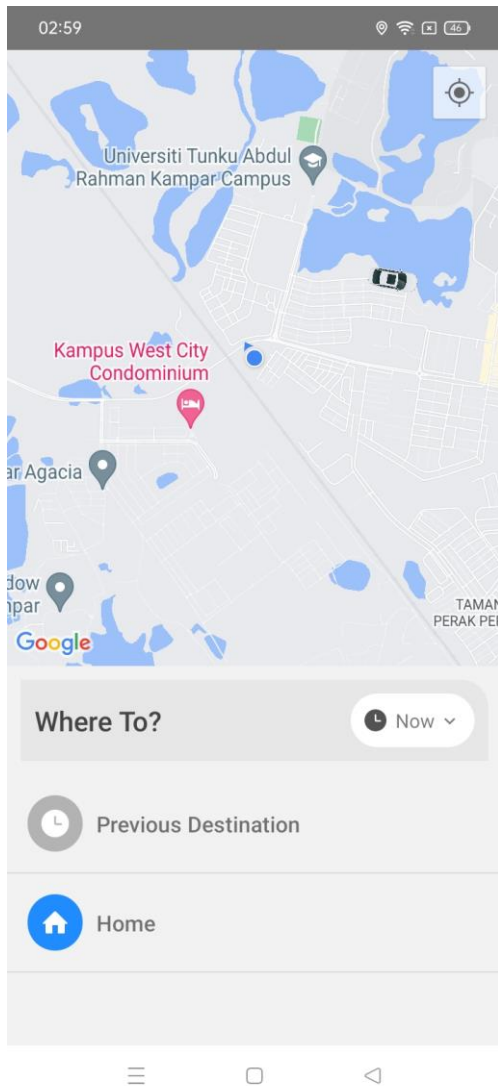


Figure 5.18 Home Map (DinoPool)

5.3.3.5 Route Map

The route map is being presented to the user after user has enter their trip origin and destination, this is for the user to make sure that they did not enter the wrong address and also giving a overview of how the route from origin to destination will be like. User can also further interact with the map view by zoom in or out or drag around.

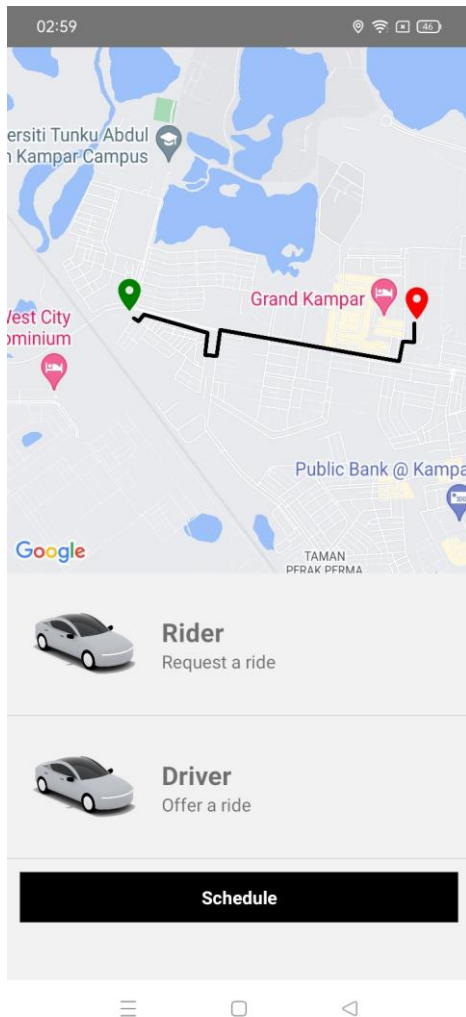


Figure 5.19 Route Map (DinoPool)

5.3.3.6 Google Place Autocomplete

The Google Place Autocomplete features is used on the ride request origin and destination input; it is to make sure that all the location and address are exist and accuracy to prevent any location not found situation. This will also make the application more consistent as the all the location are based on Google Map's address.

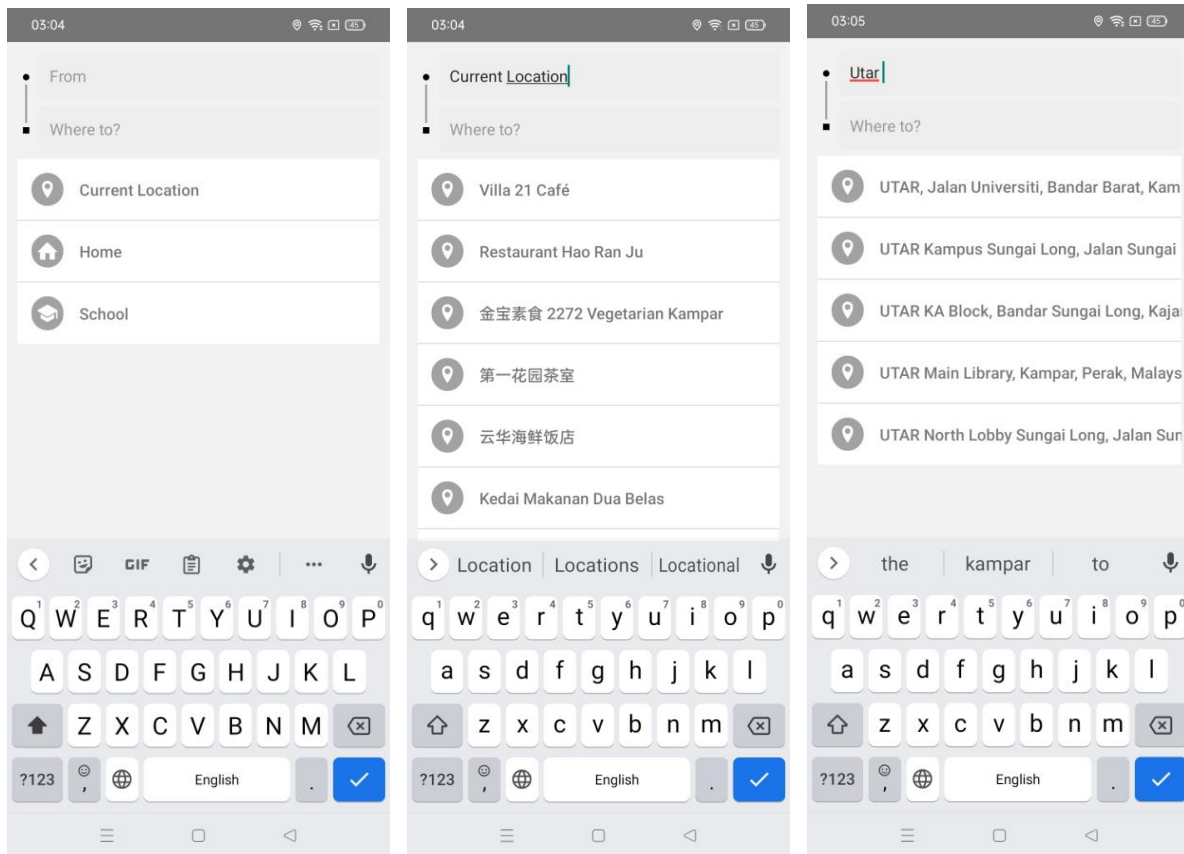


Figure 5.20 Google Place Autocomplete (DinoPool)

5.3.4 Module/Features for Users (verified)

5.3.4.1 Login

Before starting to use the application feature, users are required to have a verified account to unlocked with full features.

On the Login page, users are required to enter their email address and password to login into the application. If the user credential is matched with the data in database, the application will allow users to access to the application and a toast message will prompt to indicate user successfully login, or else an error message will prompt to inform users somethings went wrong.

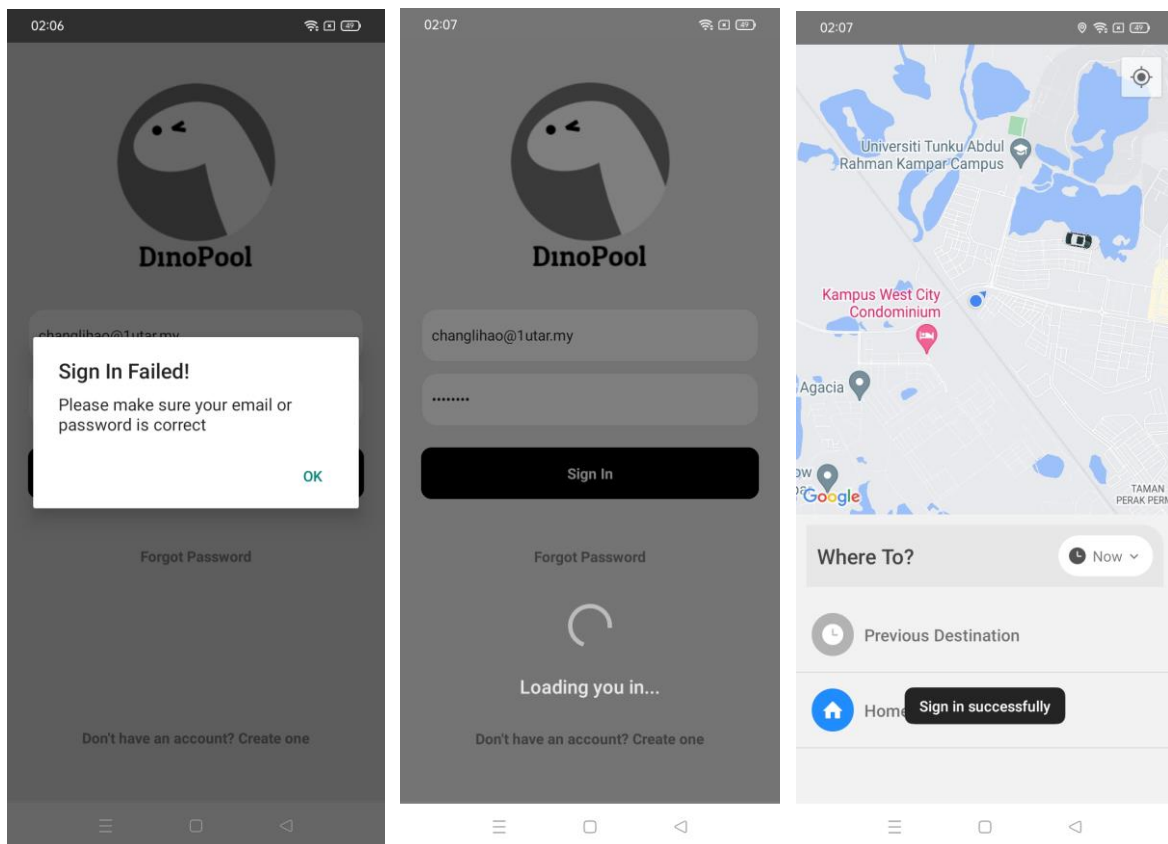


Figure 5.21 Module for verified users (Login)

The application will also prompt an alert message if user did not enter any value in email address or password input.

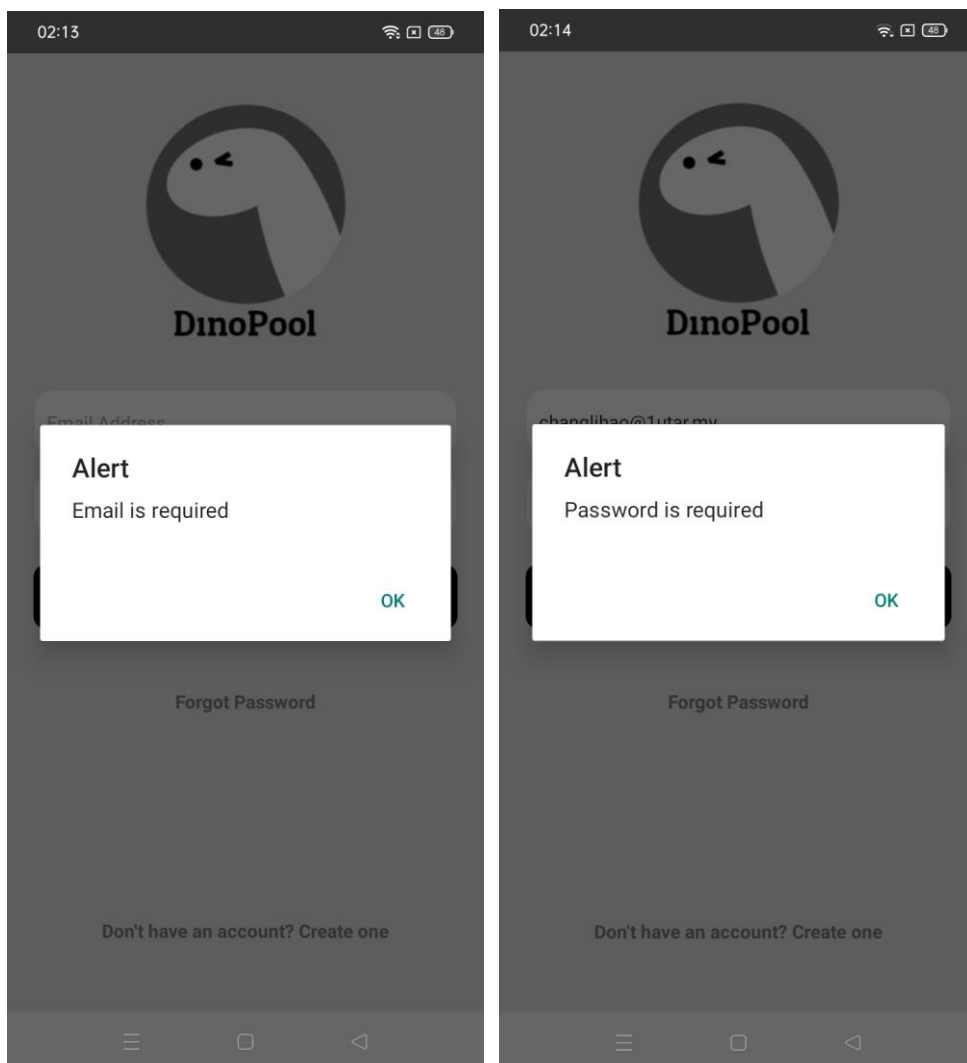


Figure 5.22 Module for verified users (Login)

5.3.4.2 Sign Up

On the Sign-Up page, users are required to enter their username, email address, password, repeat password, phone number, and upload their student id image. If user have registered successfully, a toast message will prompt to indicate user successfully register, or else an error message will prompt to inform users somethings went wrong.

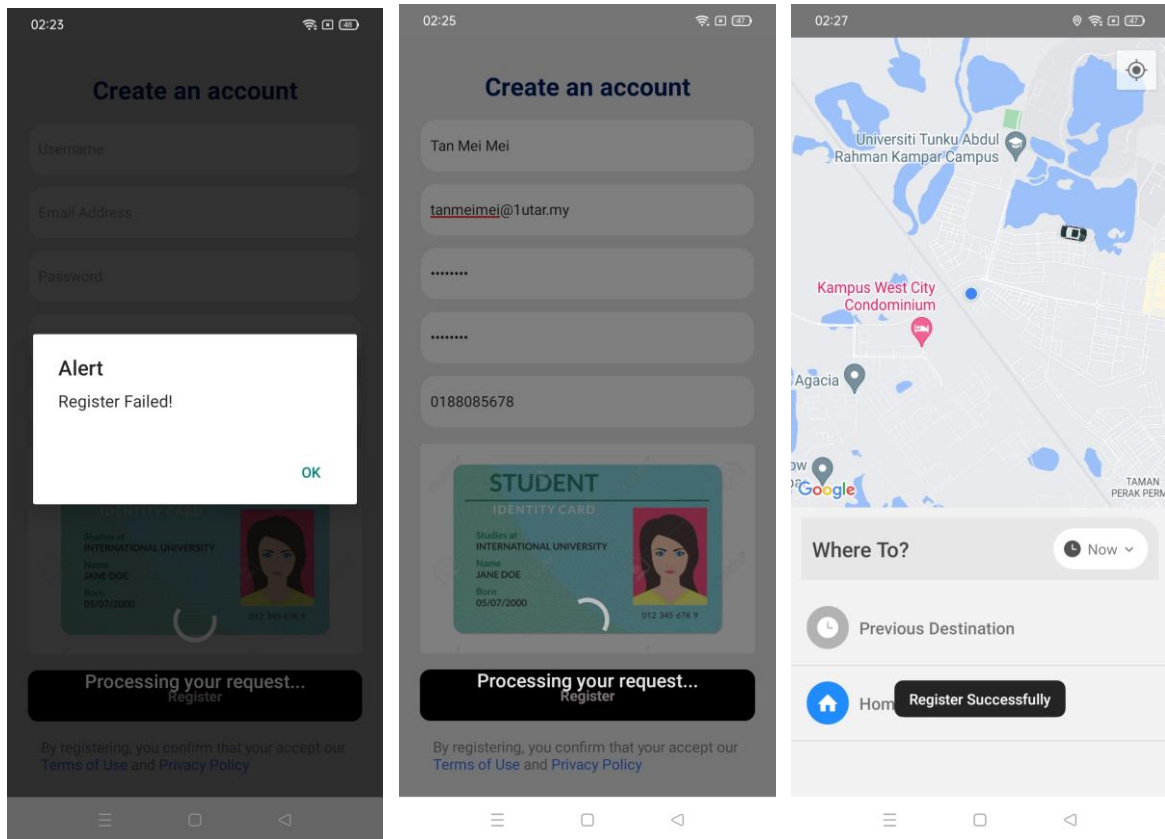


Figure 5.23 Module for verified users (Sign Up)

Chapter 5 System Implementation

Besides, all the input are required user to enter, or else the application will also prompt an alert message if user did not enter any value in username, email address, password, repeat password, phone number, and not upload their student id image.

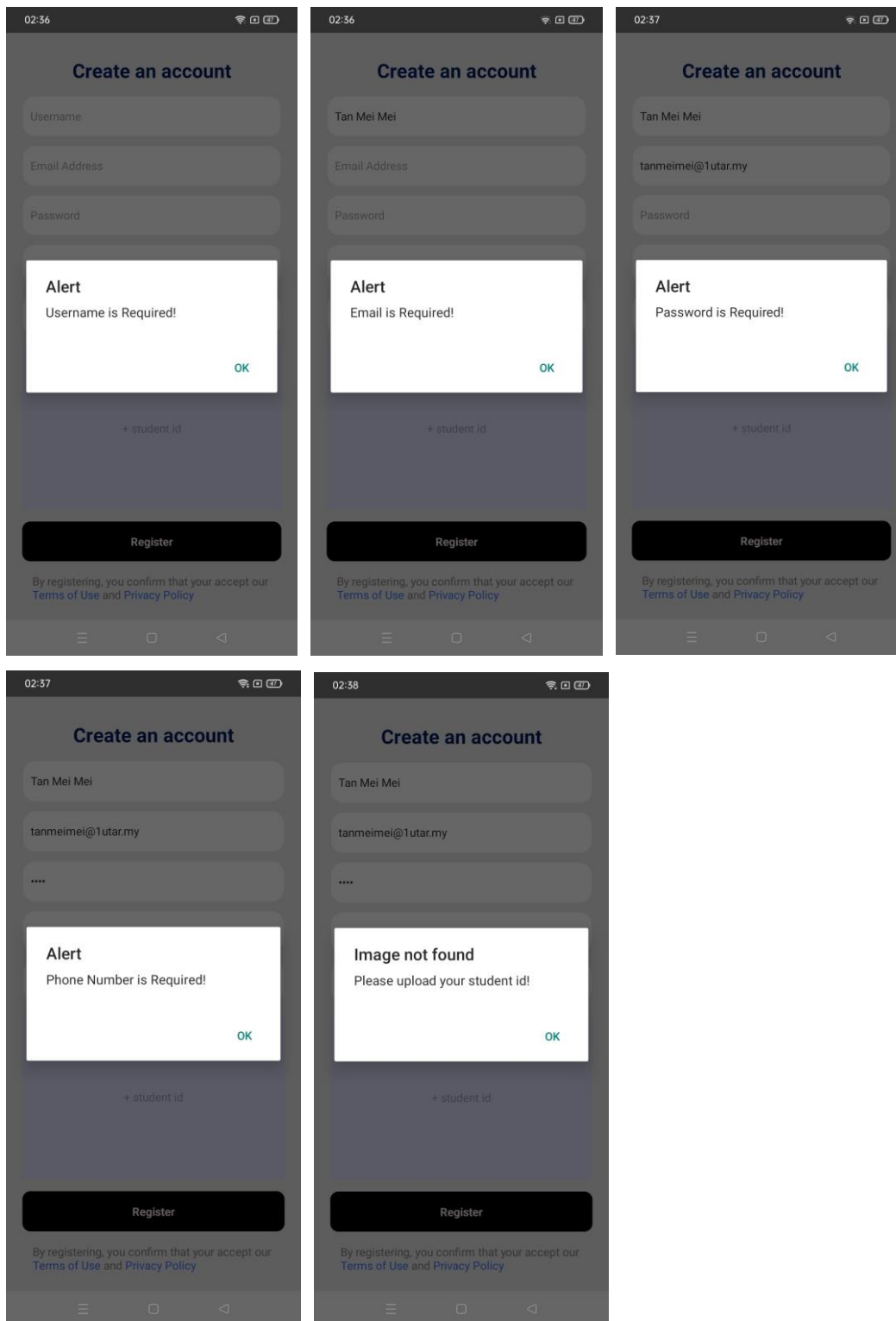


Figure 5.24 Module for verified users (Sign Up)

Chapter 5 System Implementation

If user entered password that is not matched with the repeat password input, an alert message will also be prompt to inform user. This feature is to prevent user from entering the wrong password while registering the account.

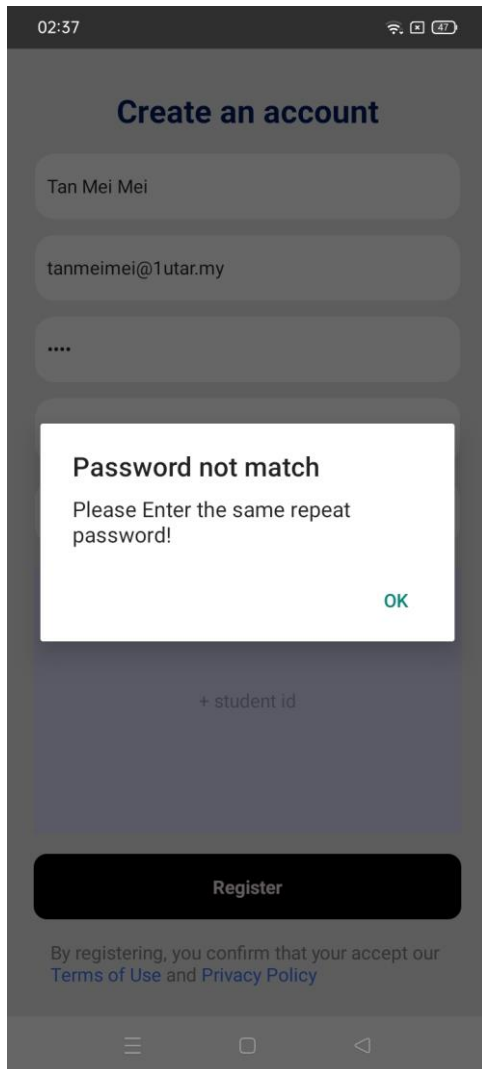


Figure 5.25 Module for verified users (Sign Up)

5.3.4.3 Forgot Password

If user have forgot their account password, user can enter their email address and request for a reset password link.

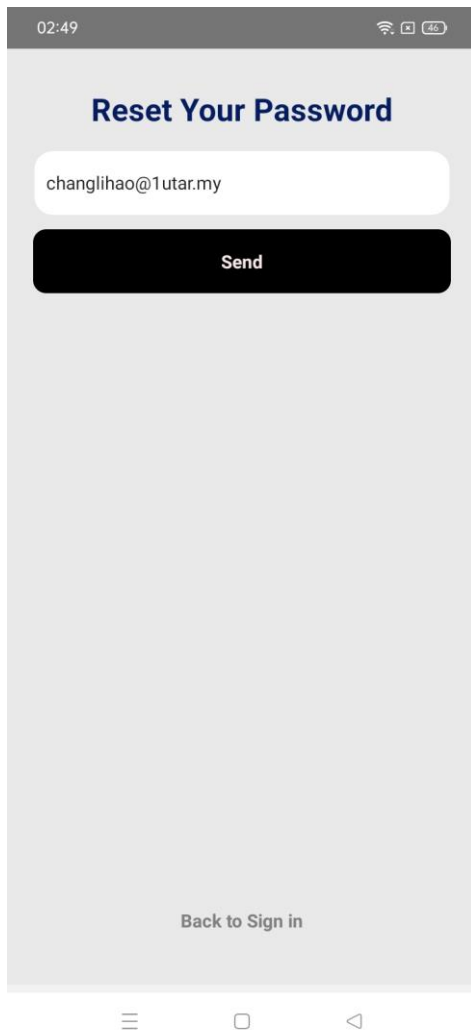


Figure 5.26 Module for verified users (Forgot Password)

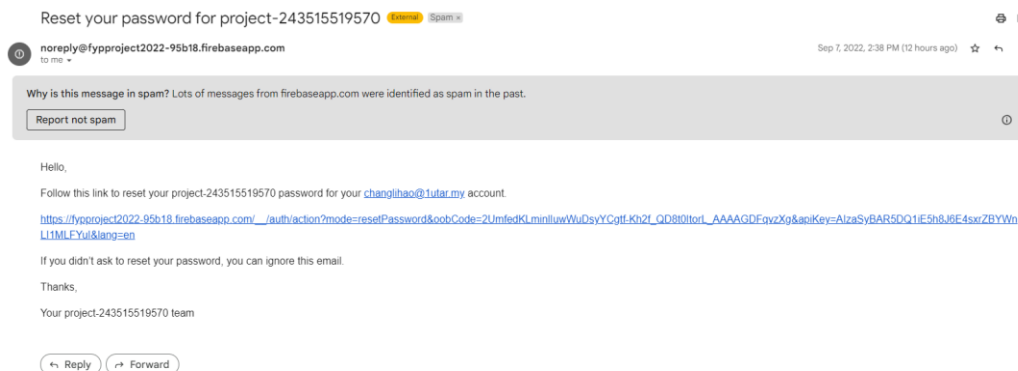


Figure 5.27 Password Reset Link (Forgot Password)

5.3.4.4 Post Ride Requests

If user have successfully login into the application, user can press on the “where to” button on the home screen to post ride request.

User will be navigated to a screen where user can enter their origin and destination, the text input on the screen has implemented with Google Place Autocomplete features which allow user to search for location based on Google’s map address.

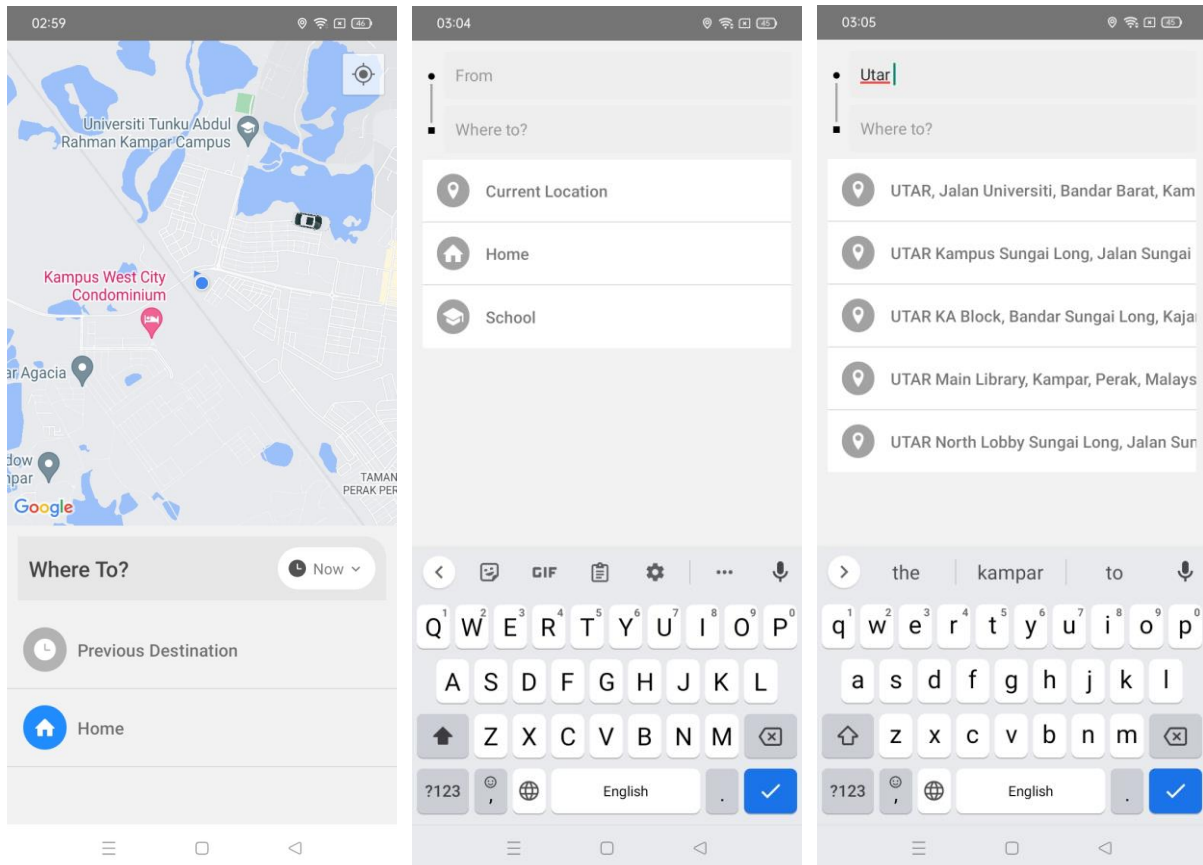


Figure 5.28 Module for verified users (Post Ride Requests)

Chapter 5 System Implementation

User can also press on the “Current Location” to search for nearby location on the origin text input.

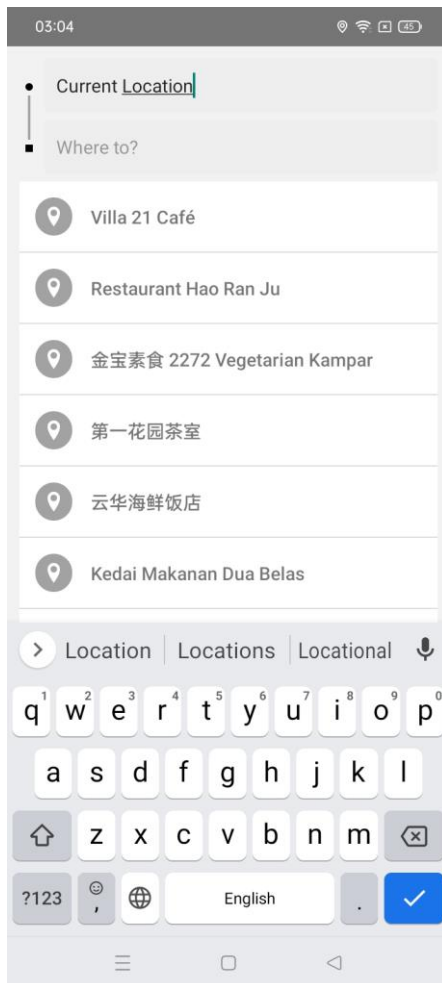


Figure 5.29 Module for verified users (Post Ride Requests)

Chapter 5 System Implementation

After user have enter the origin and destination text input, user will automatically navigate to the route map screen and user are required to choose which role they want to request, either Rider or Driver options. If user did not select any of the role, an alert message will prompt to inform user that selection is required.

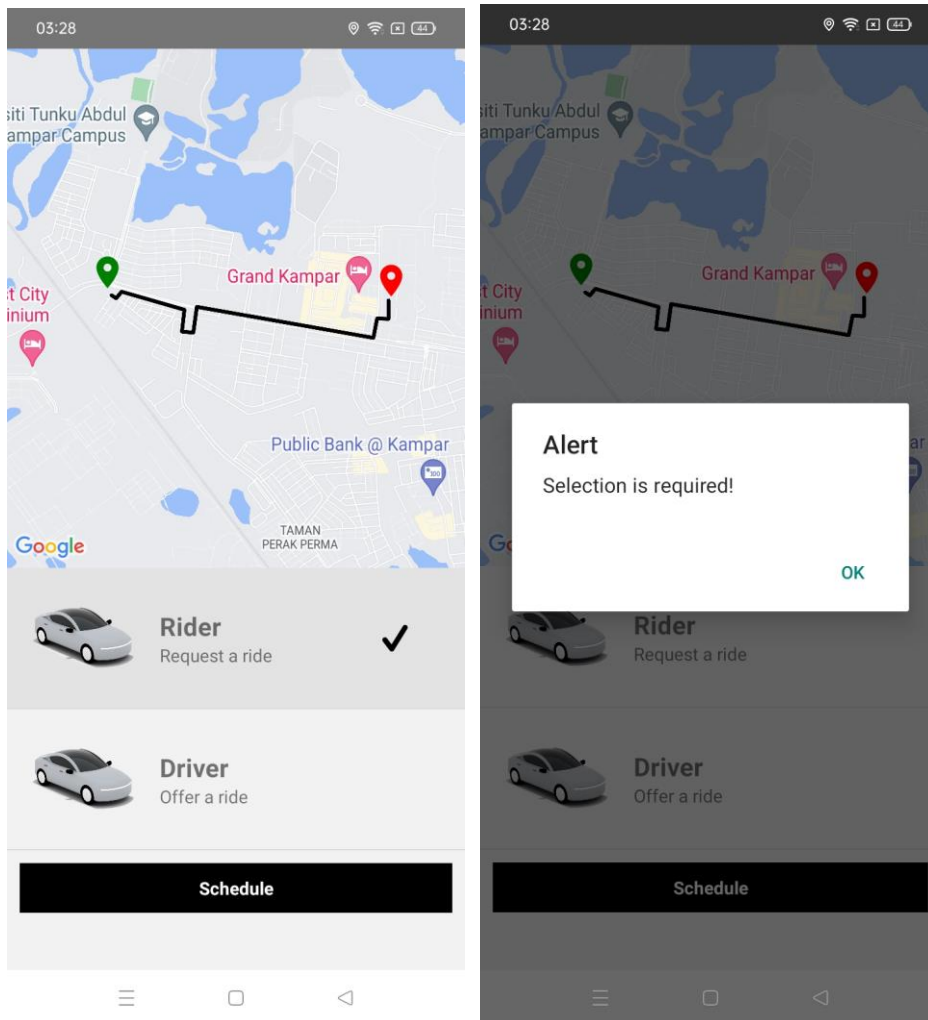


Figure 5.30 Module for verified users (Post Ride Requests)

After user have select a role and press the schedule button to proceed, user will be navigated to ride request screen. The screen will display the origin and destination address for the user, and user are required to select one of the carpool services and select date and time. If user did not select any carpool service or date and time, an alert message will prompt to inform user that selection is required.

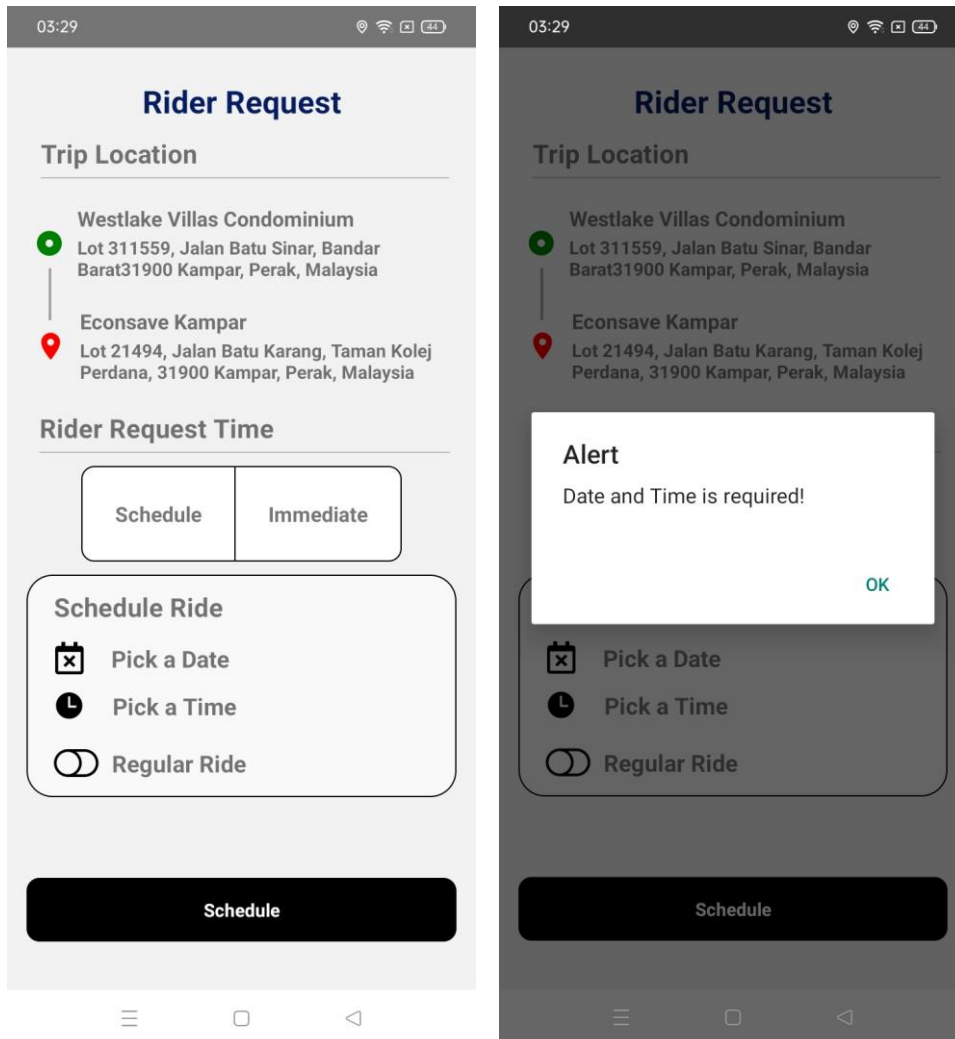


Figure 5.31 Module for verified users (Post Ride Requests)

For the carpool service, user can choose from either Schedule or Immediate.

If user choose to request an immediate ride, the date and time will automatically fill with current system date and time.

Else if user choose to request a schedule ride, user can select a date and time to request for the ride.

Else, if user choose to request a schedule ride, but they want to travel regularly, user can press on the toggle button to switch on the regular ride service.

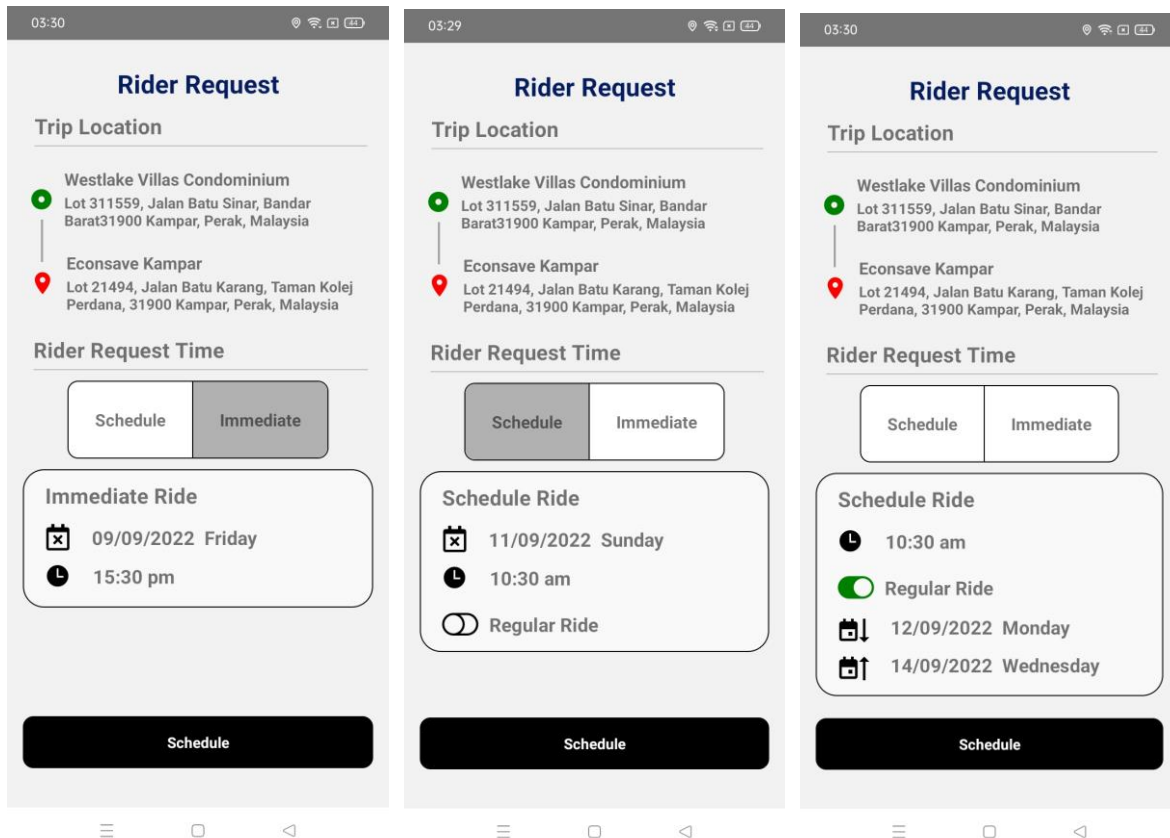


Figure 5.32 Module for verified users (Post Ride Requests)

Chapter 5 System Implementation

However, not matter which carpool service user choose, date and time is required to submit the ride request, or else the alert message will prompt and inform user date and time is required. User can press on the “Pick a Date” and “Pick a Time” button, and a date and time picker will prompt out for user to select date and time for the ride request.

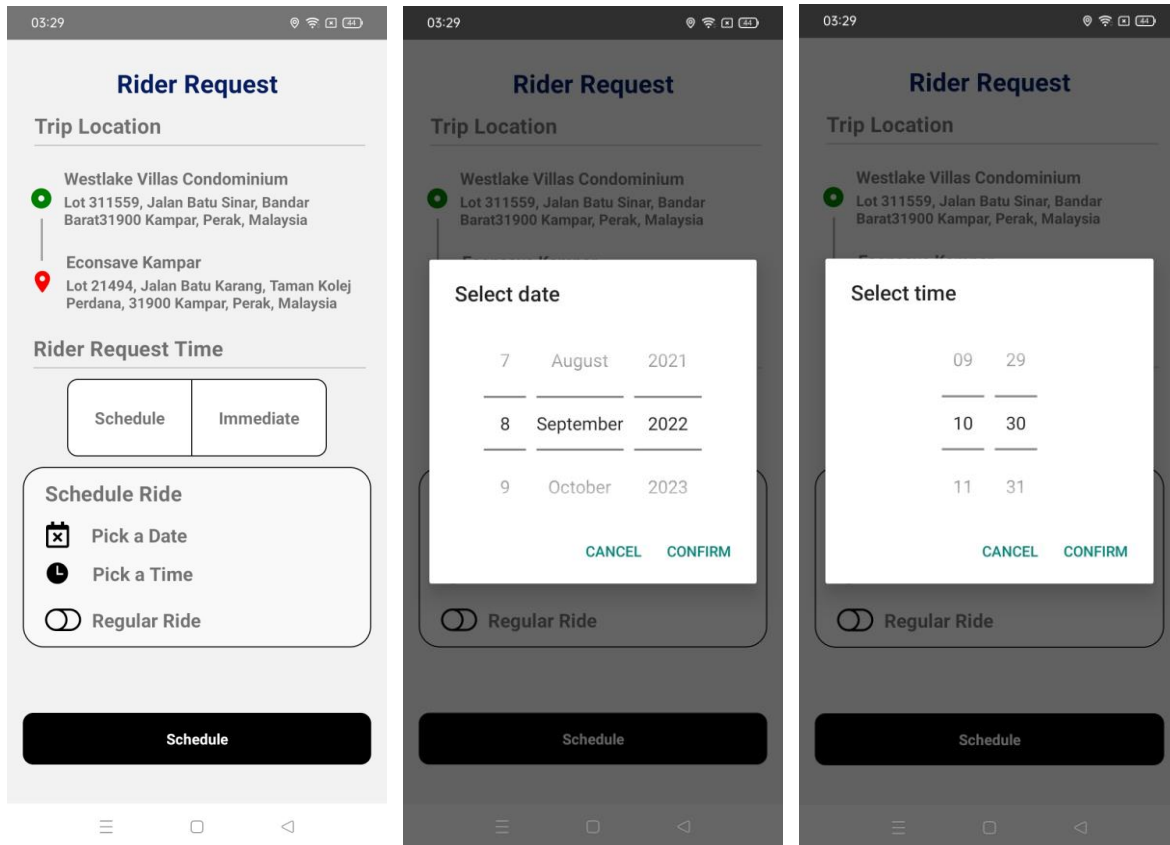


Figure 5.33 Module for verified users (Post Ride Requests)

Once the user has finish selecting carpool service and have entered the date and time, user can press on the “Schedule” button, an alert message will prompt to ask if the user is sure to submit the ride request. If the user press on Yes, the ride request will be submitted, and a toast message will prompt to indicate user have successfully submit the ride request. Else if the user press on No, the alert message box will disappear and user and further adjust their selection.

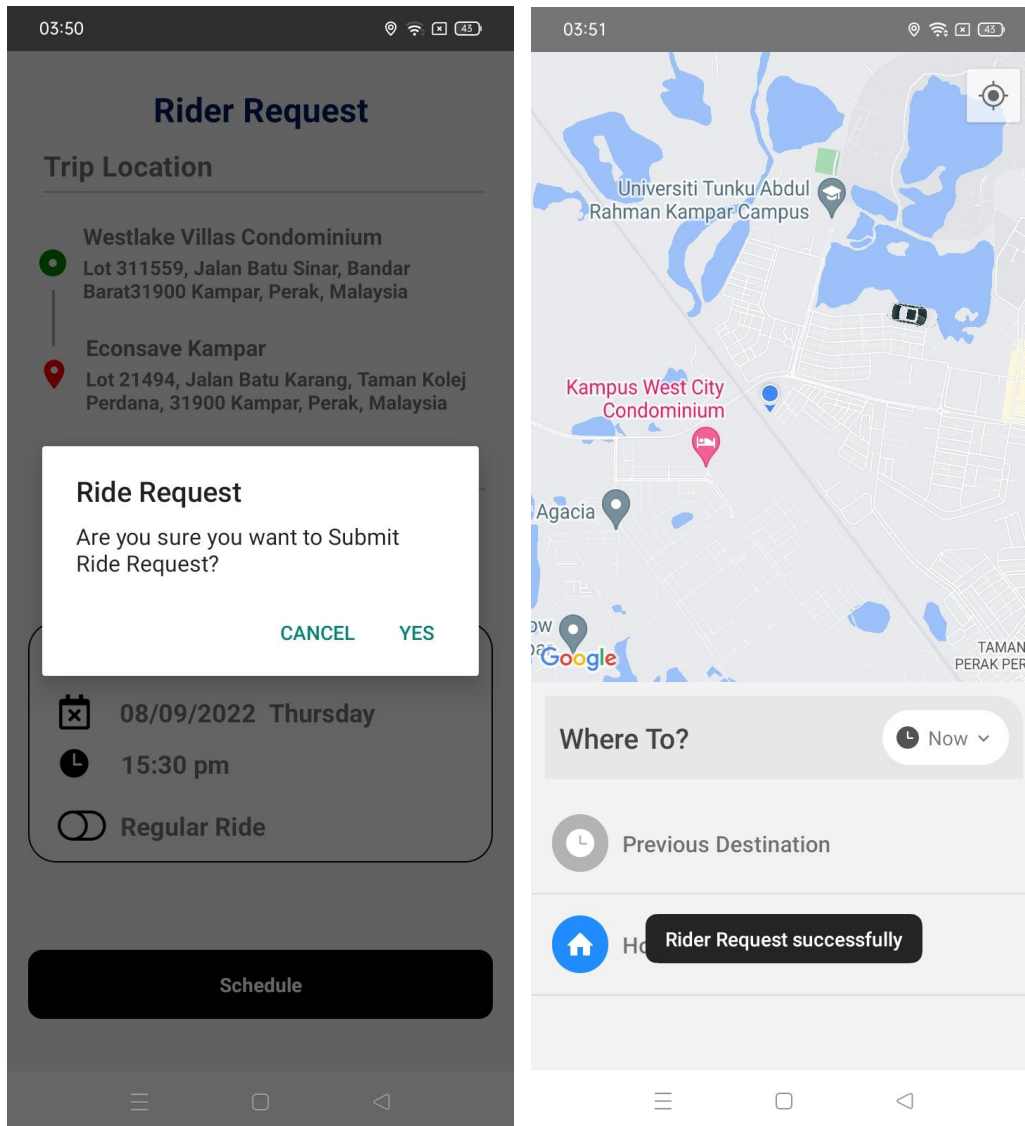


Figure 5.34 Module for verified users (Post Ride Requests)

5.3.4.5 Manage User Profile

If user have successfully login into the application, user can swipe a finger from the left edge of the screen and a Drawer Menu will appear. User can press on the “Profile” button on the drawer menu to manage their user profile.

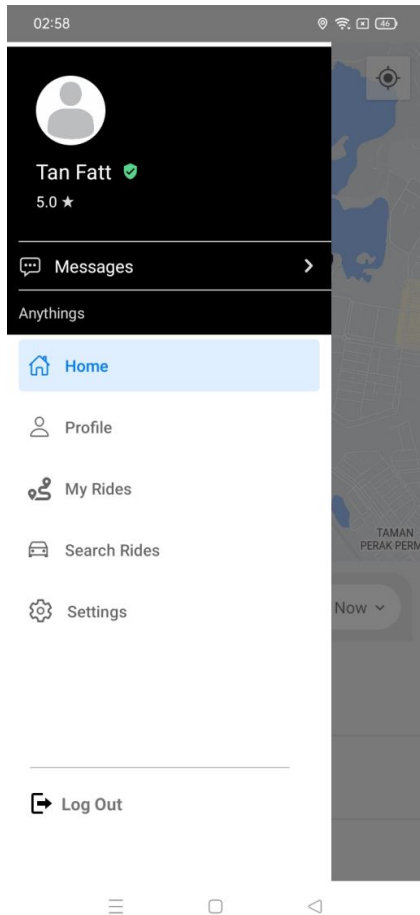


Figure 5.35 Module for verified users (Manage User Profile)

Chapter 5 System Implementation

On the profile screen, user can view their personal details such as their username, email address, verified status and phone number. If user want to modify of the details, user can press on the top right edit button, the input box and update button will appear. User can press on the input box to start modifying their details.

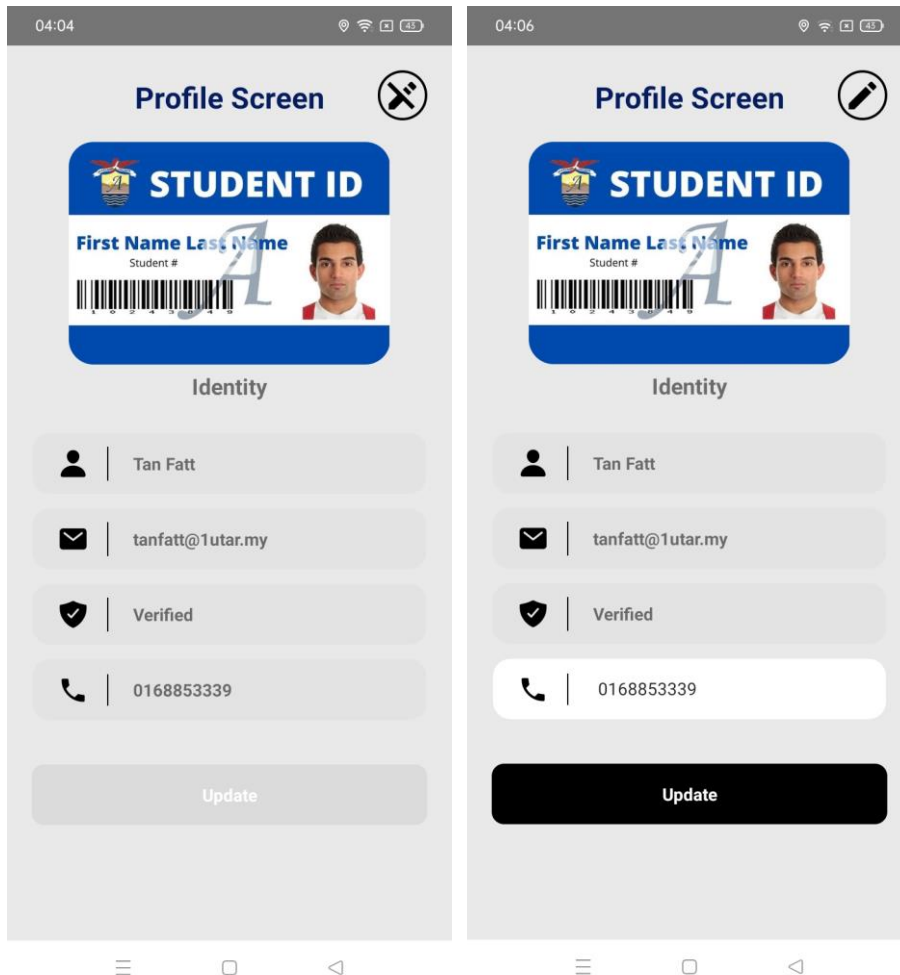


Figure 5.36 Module for verified users (Manage User Profile)

After user have finish editing their personal details, user can press on the update button to update their details, a toast message will prompt to indicate user have successfully updated their personal details.

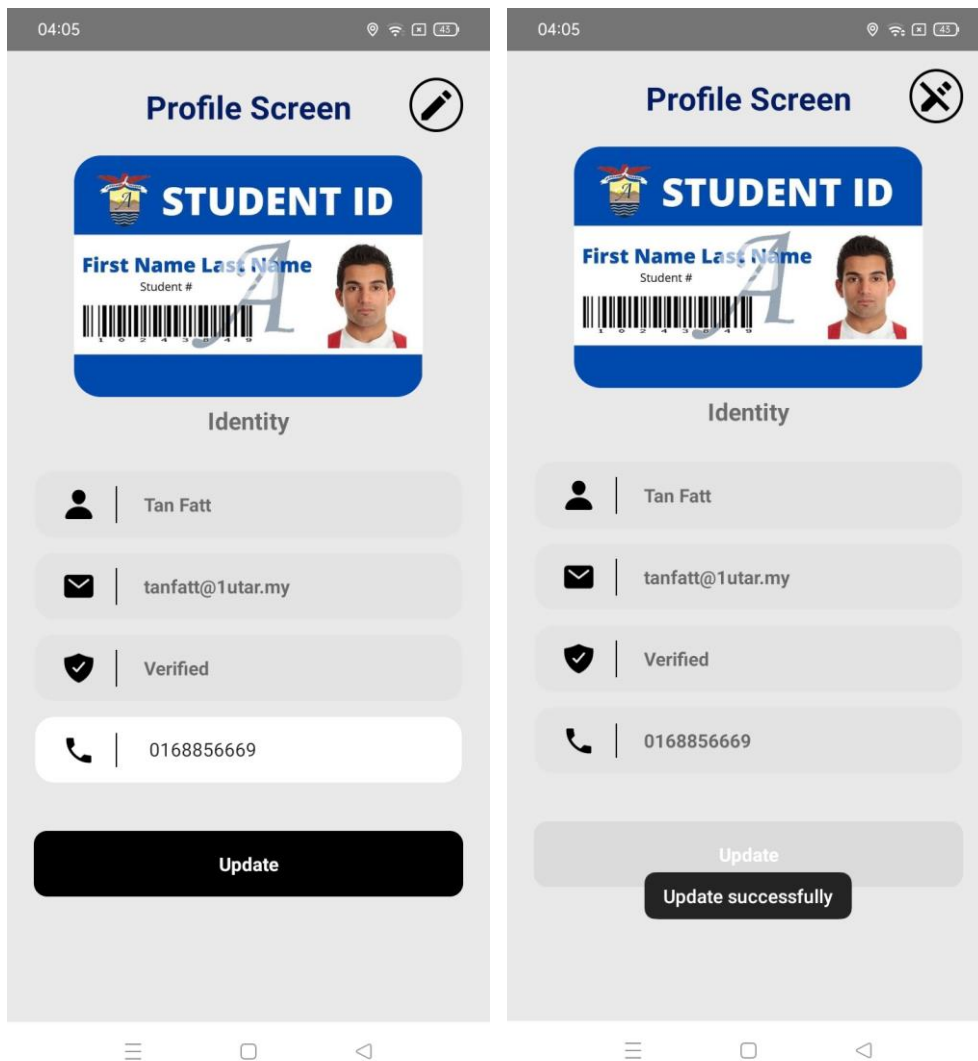


Figure 5.37 Module for verified users (Manage User Profile)

5.3.4.6 Check Requested Rides

If user have successfully login into the application, user can swipe a finger from the left edge of the screen and a Drawer Menu will appear. User can press on the “My Rides” button on the drawer menu to check their ride request posts.

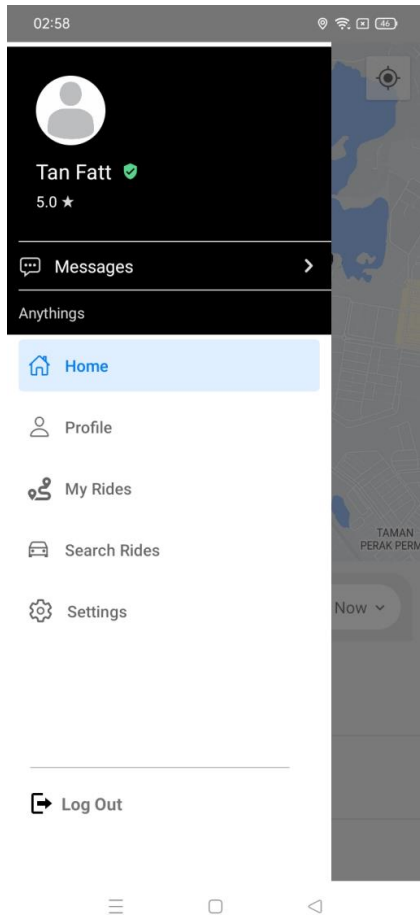


Figure 5.38 Module for verified users (Check Requested Rides)

On my rides screen, user can view their ride request post by switch between Rider or Driver role and all the rides have also been categories base on different carpool services.

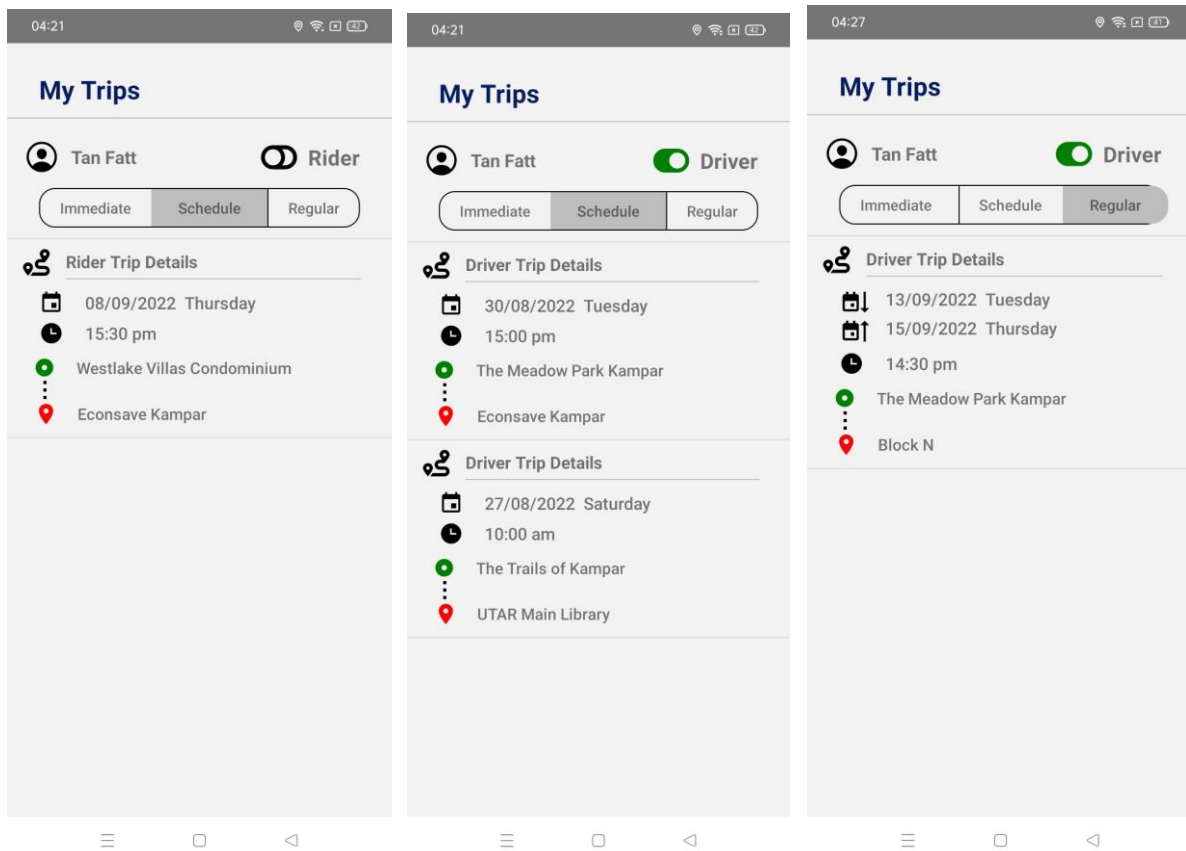


Figure 5.39 Module for verified users (Check Requested Rides)

User can press on one of the ride requests posts and it will navigate to the details page of the particular ride request post pressed on. The ride request post details include many information such as a route map of the trips from origin to destination, trip's date, and time, and requested riders' details on this trip. The location of the riders will also be rendered on the route map for the user which is the driver to fetch them.

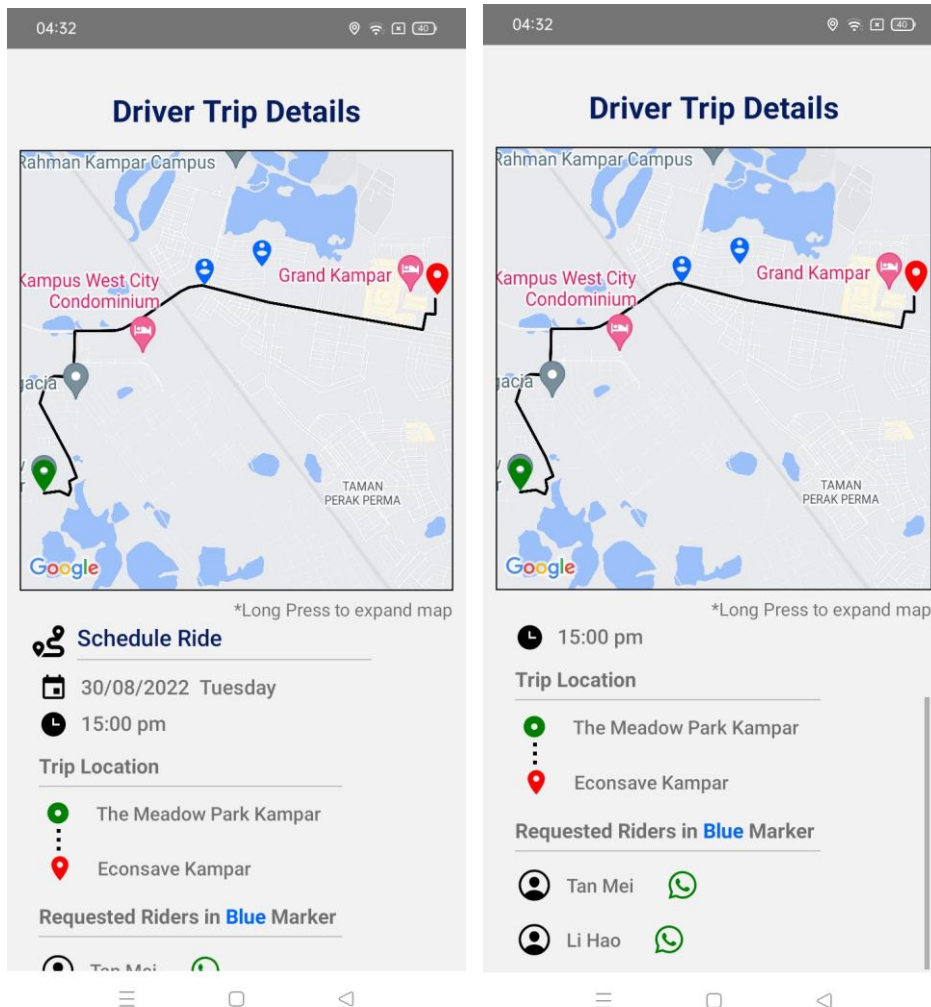


Figure 5.40 Module for verified users (Check Requested Rides)

User can long press on the route map to expand the map view for better visual view.

User can press on the close button on the top right corner to close the expanded map view.

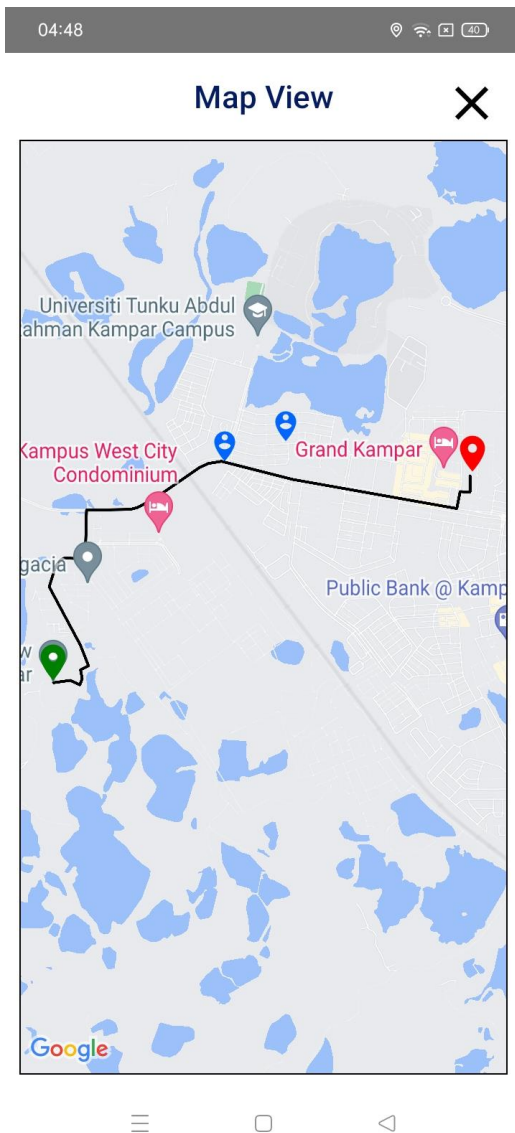


Figure 5.41 Module for verified users (Check Requested Rides)

User are allowed to contact the requested rider by press on the small WhatsApp icon on the right side of the requested riders' name. A reminder message box will prompt to remind user safety always come first when riding with strangers, if user press on OK button, user will be directed to WhatsApp and a simple message template is ready for the user to contact the request rider for further ride arrangement.

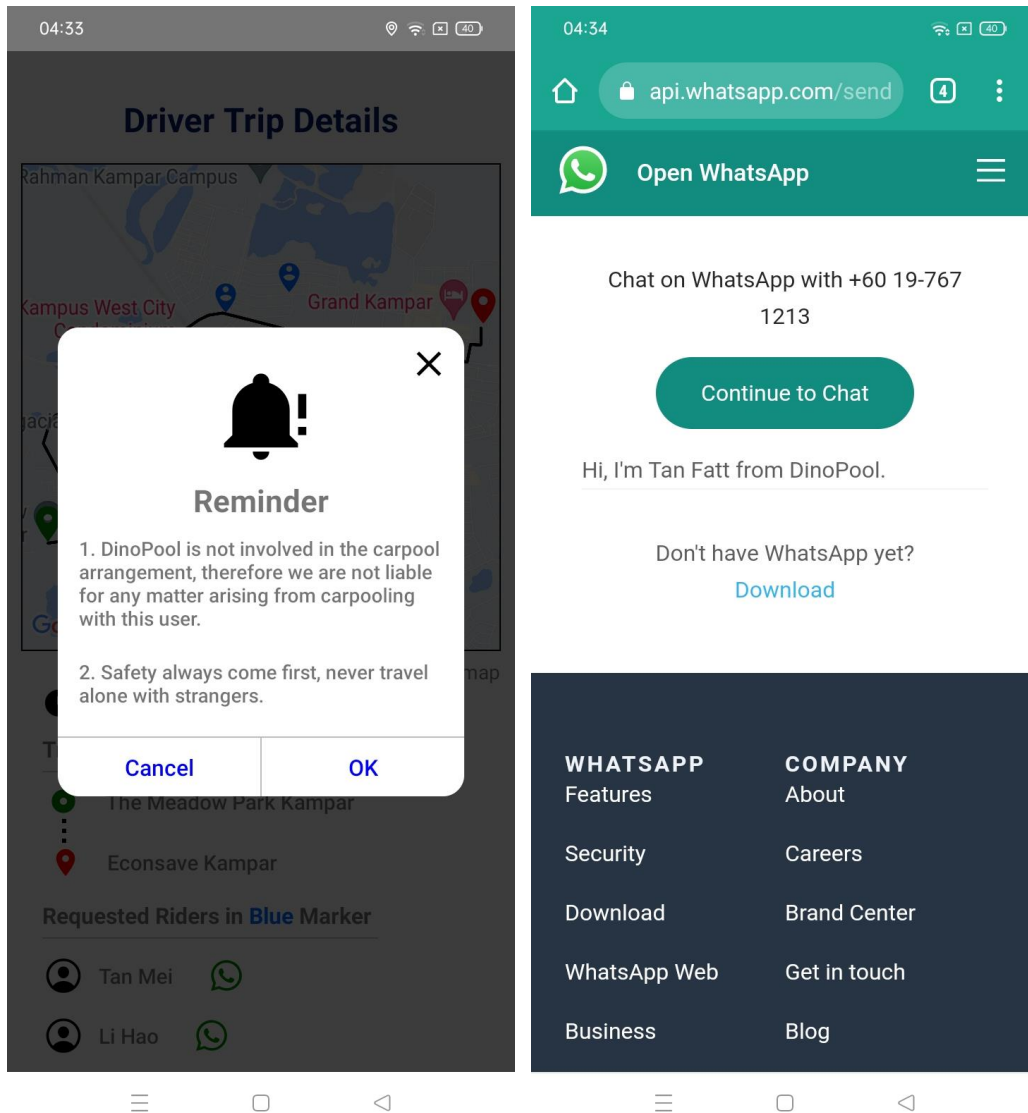


Figure 5.42 Module for verified users (Check Requested Rides)

5.3.4.7 Search Rides

If user have successfully login into the application, user can swipe a finger from the left edge of the screen and a Drawer Menu will appear. User can press on the “Search Rides” button on the drawer menu to check on the ride request posts available.

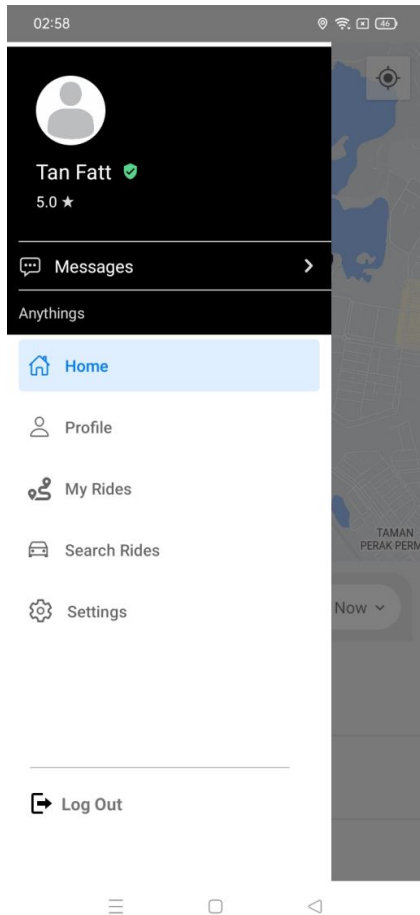


Figure 5.43 Module for verified users (Search Rides)

On search rides screen, user can view all the ride request post by switch between Rider or Driver role and all the rides have also been categories base on different carpool services.

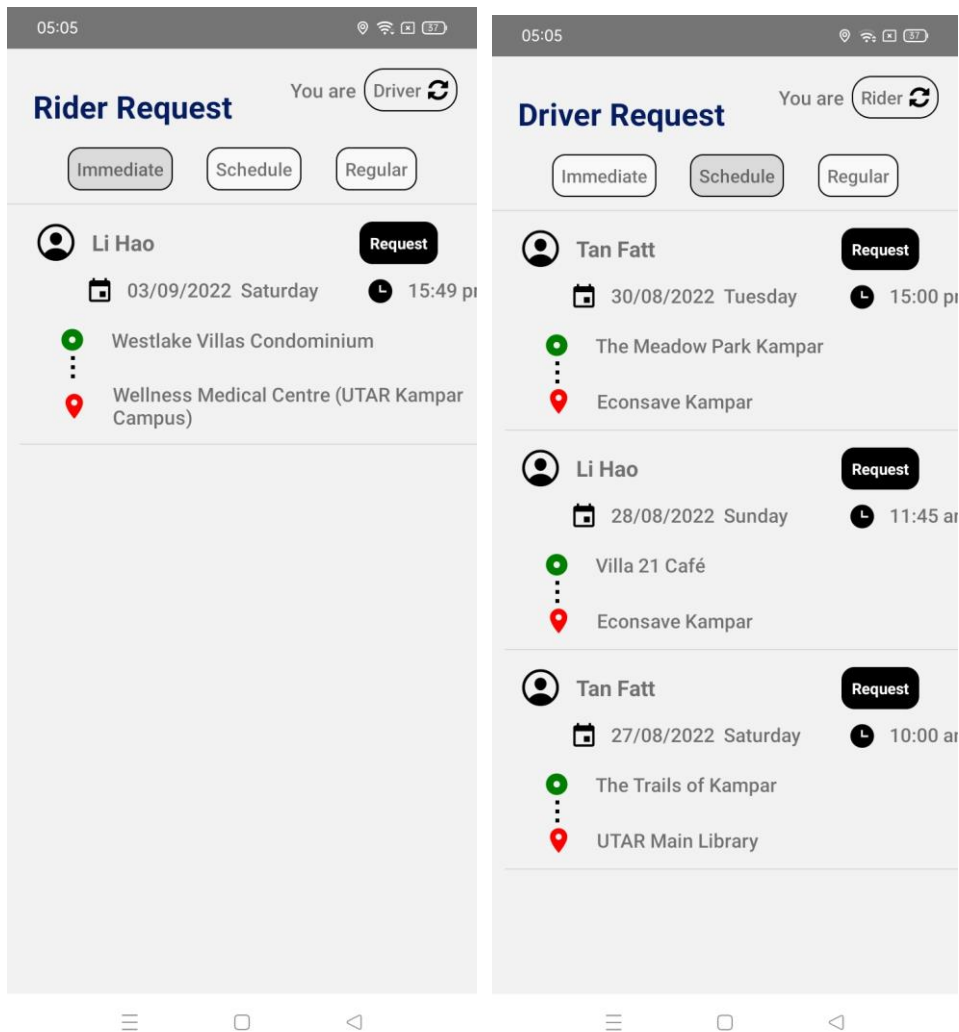


Figure 5.44 Module for verified users (Search Rides)

User can press on one of the ride requests posts and it will navigate to the details page of the particular ride request post pressed on. The ride request post details include many information such as a route map of the trips from origin to destination, the username of the ride request belongs, trip's date, and time, and requested riders' details on this trip. The location of the requested riders will also be rendered on the route map.

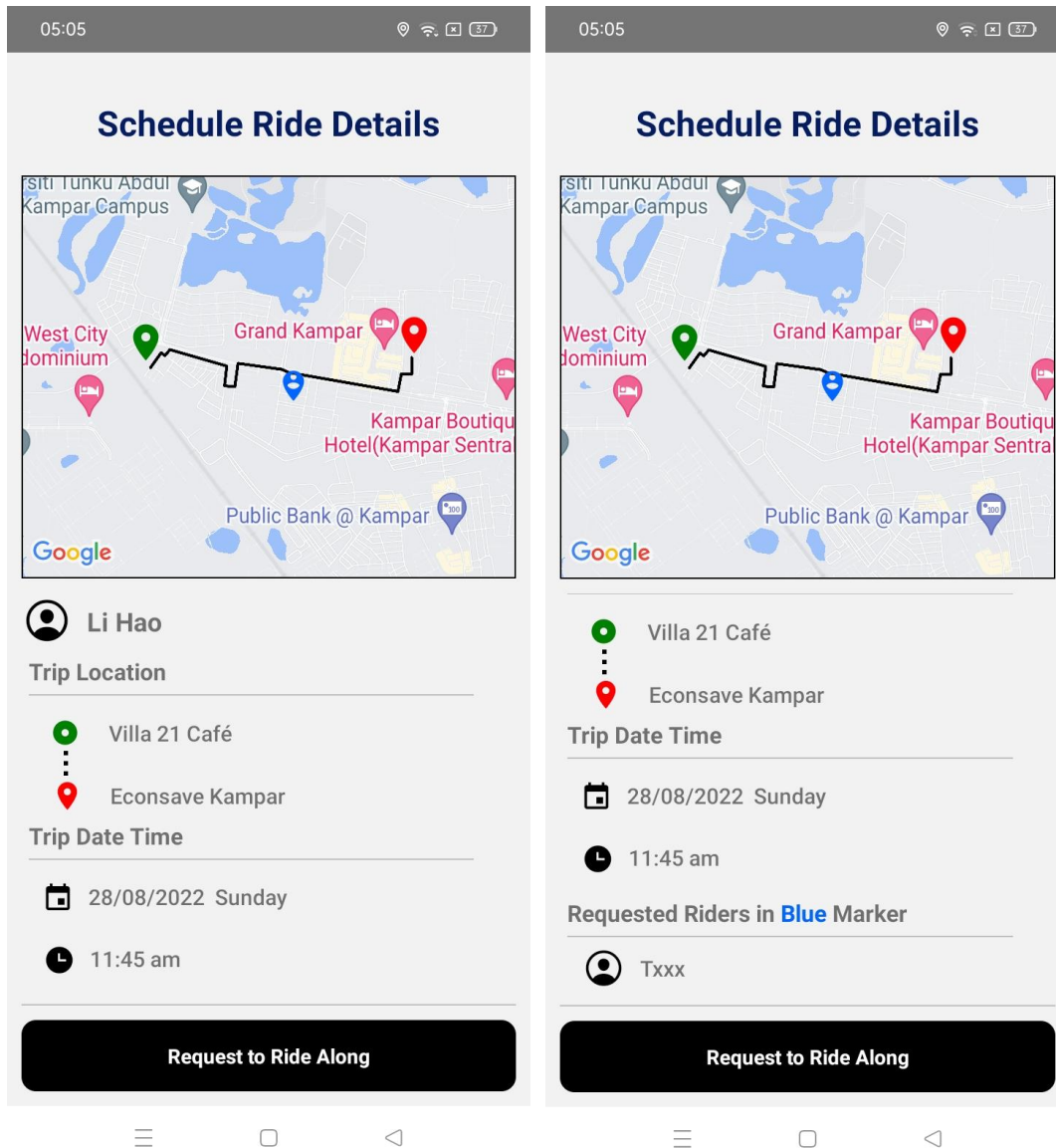


Figure 5.45 Module for verified users (Search Rides)

5.3.4.8 Request Ride Along

On the ride request post's detail screen, users are allowed to Request to Ride Along on a Driver request post.

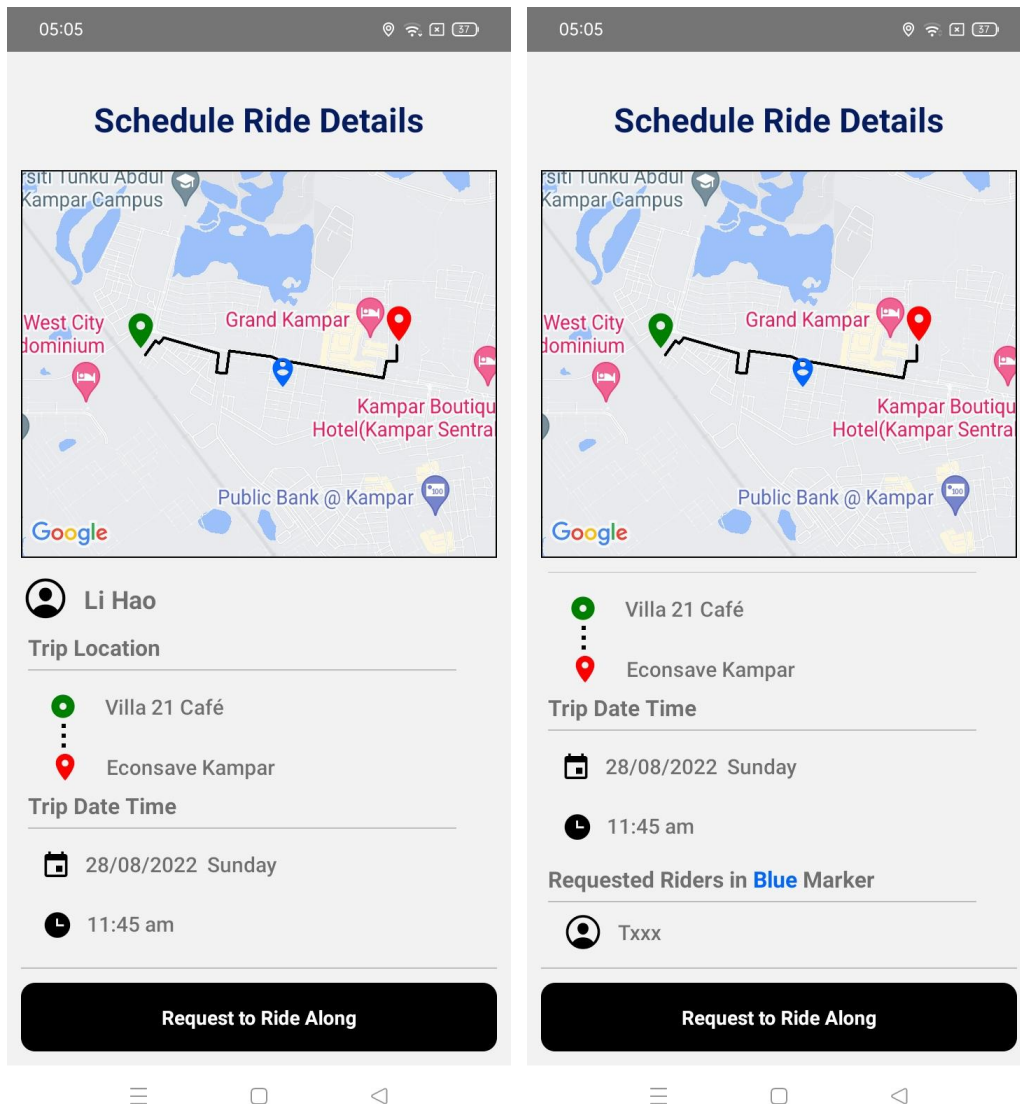


Figure 5.46 Module for verified users (Request Ride Along)

When user press on the “Request to Ride Along” button, a map view will be popped up with a “Select Location” button, user press on the “Select Location” button and a location pin will appear on the map view, user can long press and drag the location pin to select a location to be picked up along the driver’s route. Once user have finish selecting, user press on “Done” button and it will navigate back to the ride request details page with the location pin selected rendered.

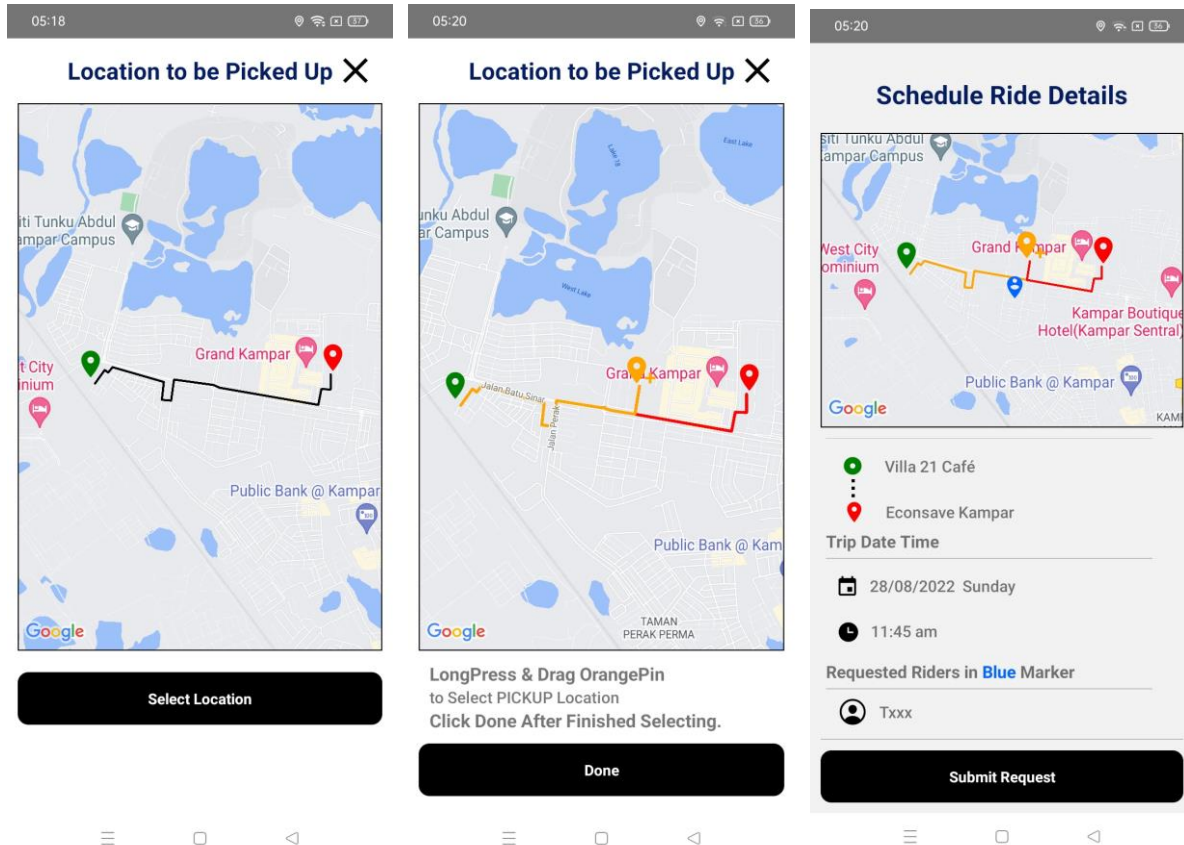


Figure 5.47 Module for verified users (Request Ride Along)

Chapter 5 System Implementation

Finally, user can press on the submit request button to submit the Ride Along request. A reminder message box will prompt to remind user safety always come first when riding with strangers, if user press on OK button, that will indicate user agree on the reminder and has submitted the ride along request. A toast message will prompt to inform user that the ride along request has successfully been submitted.

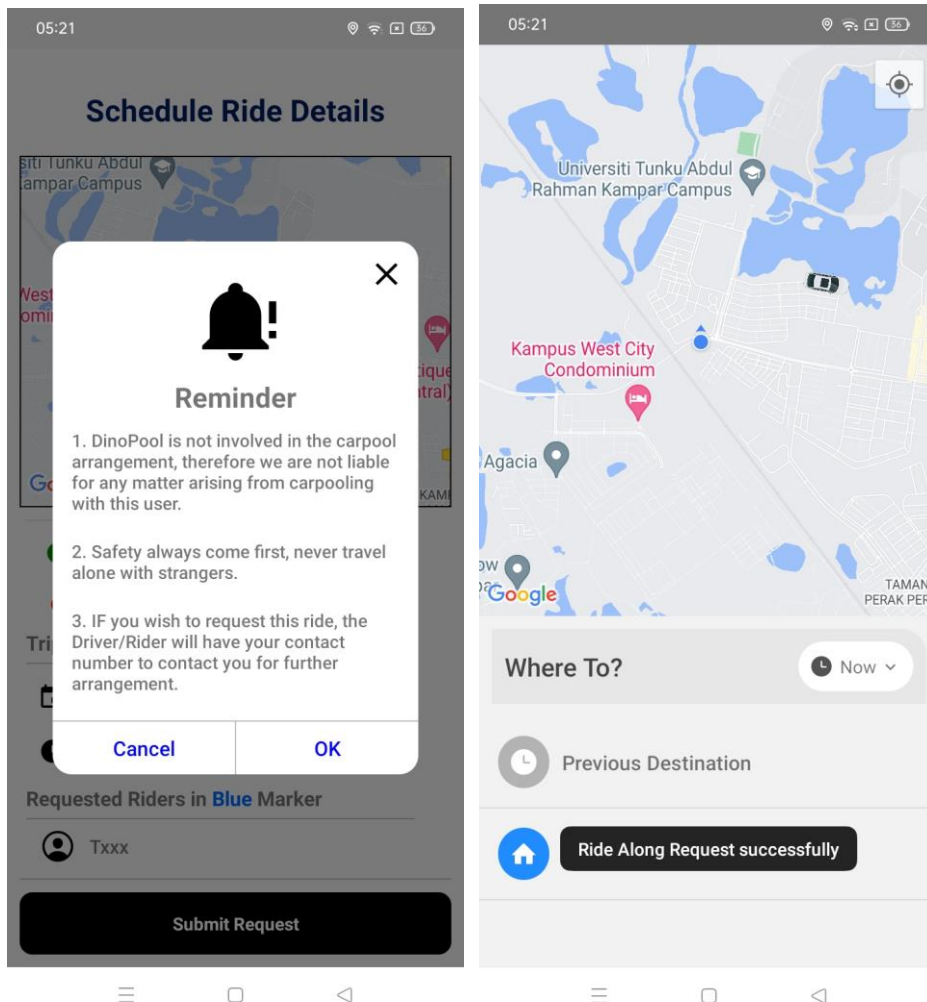


Figure 5.48 Module for verified users (Request Ride Along)

When user return to the same ride request's details screen, the button will become green in color and indicate that user have request this ride.

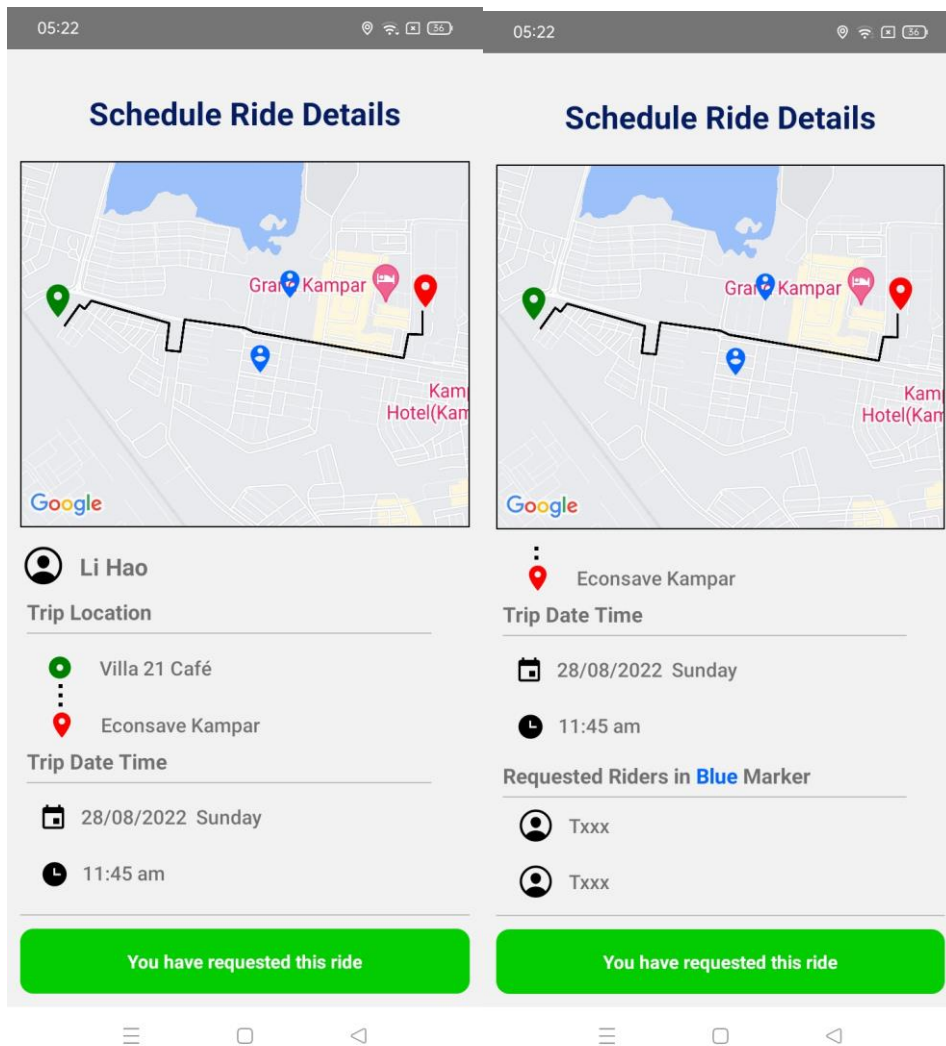


Figure 5.49 Module for verified users (Request Ride Along)

5.3.4.9 Request Pick Up

On the ride request post's detail screen, users are allowed to Request to Pick Up on a Rider request post.

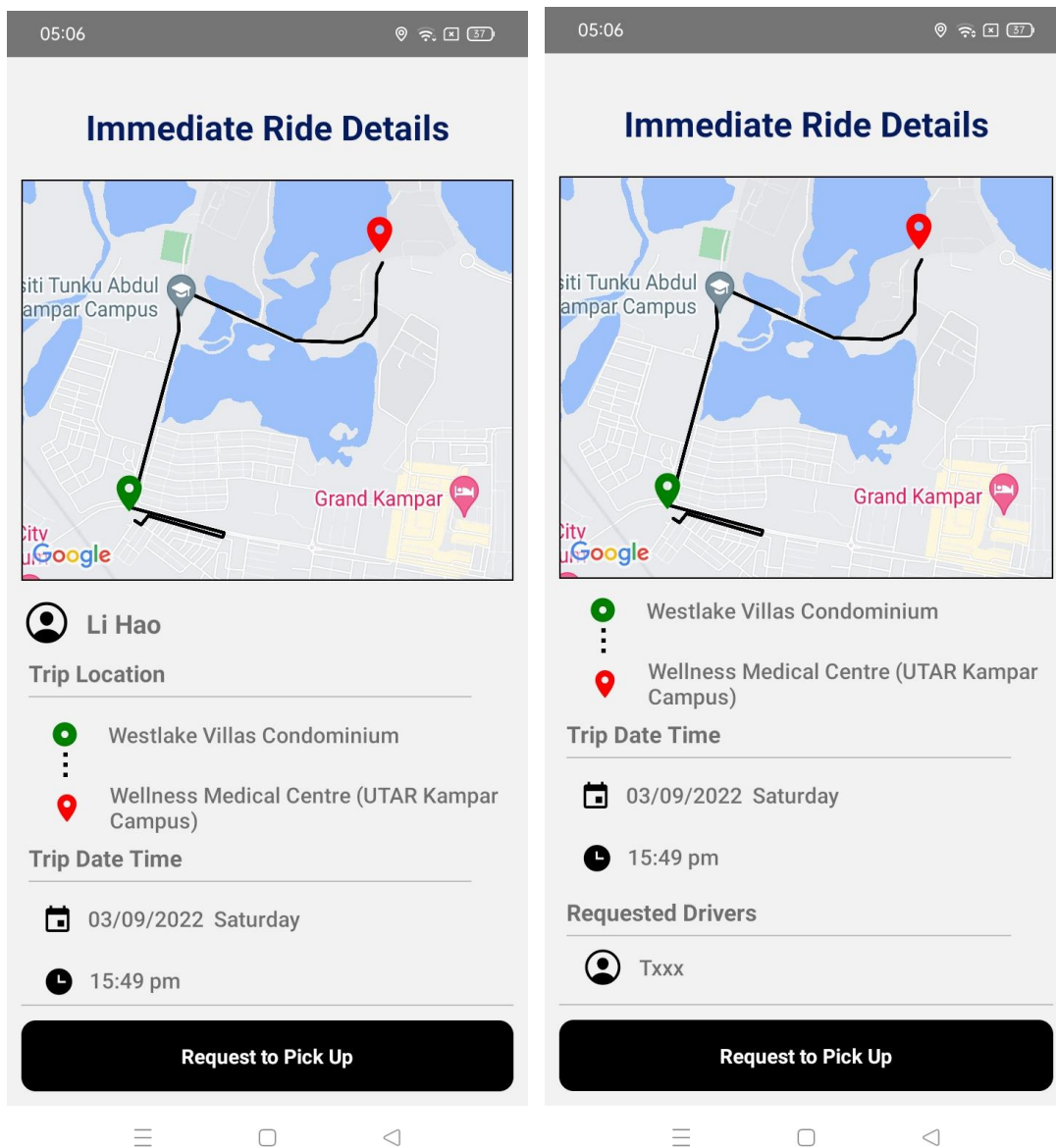


Figure 5.50 Module for verified users (Request Pick Up)

When user press on the “Request to Pick Up” button, a reminder message box will prompt to remind user safety always come first when riding with strangers, if user press on OK button, that will indicate user agree on the reminder and has submitted the ride along request. A toast message will prompt to inform user that the pick up request has successfully been submitted.

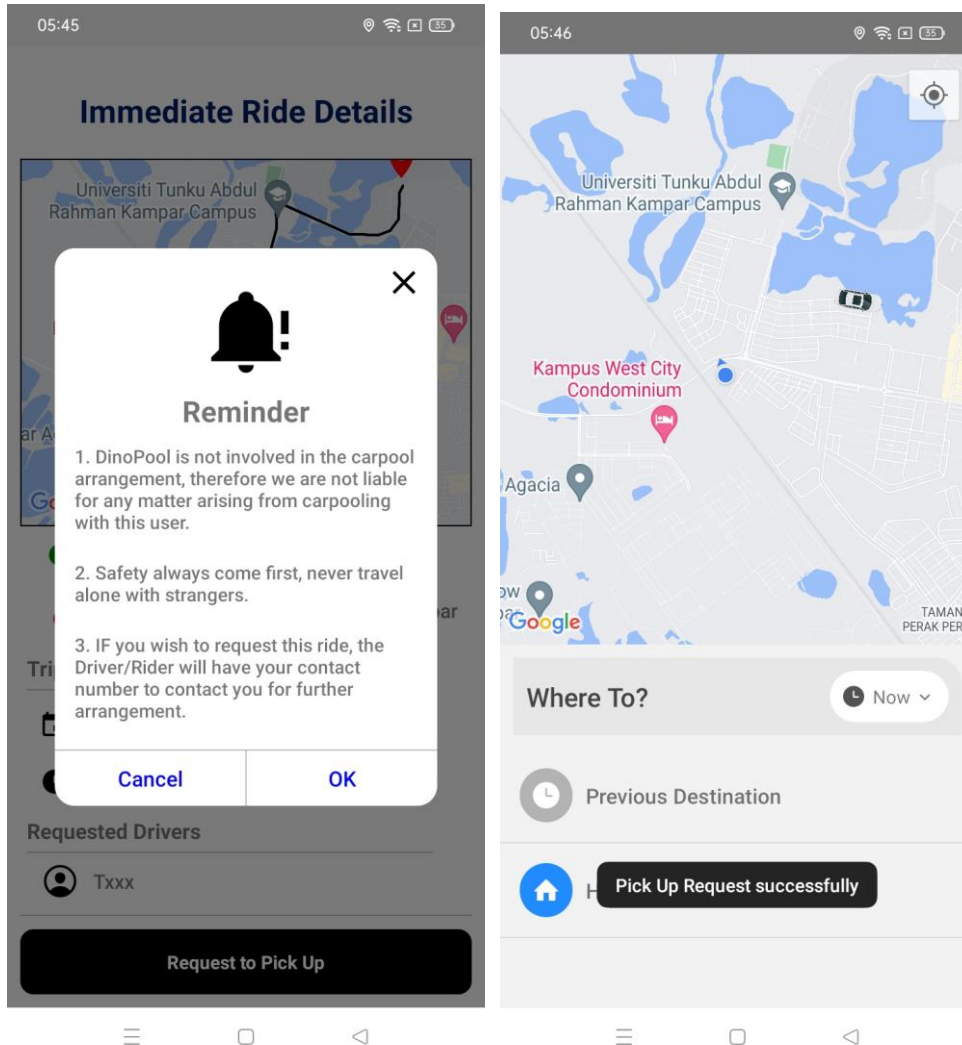


Figure 5.51 Module for verified users (Request Pick Up)

When user return to the same ride request's details screen, the button will become green in color and indicate that user have request this ride.

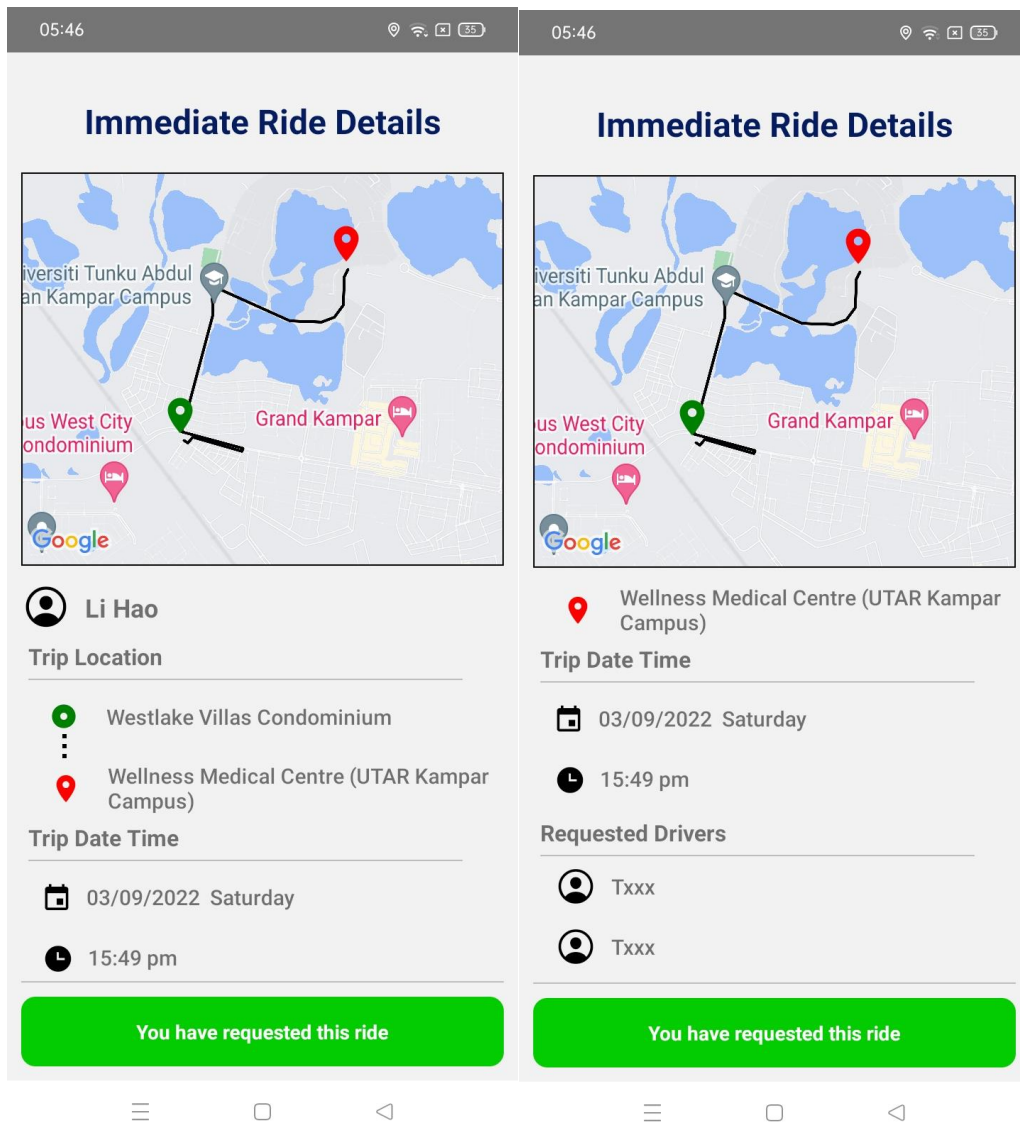


Figure 5.52 Module for verified users (Request Pick Up)

5.3.4.10 Set Home Address

If user have successfully login into the application, user can swipe a finger from the left edge of the screen and a Drawer Menu will appear. User can press on the “Settings” button on the drawer menu to set home address.

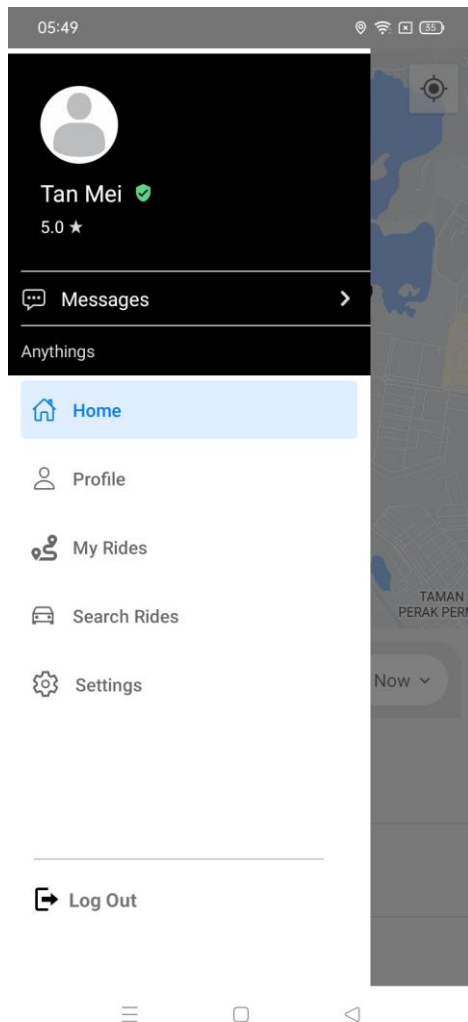


Figure 5.53 Module for verified users (Set Home Address)

On the settings screen, user can view their personal details such as their home address and school address. Initially when user create a new account, home and school address will remain empty. Therefore, user can set their home and school address for the future use.

If user want to set home address, user can press on the top right edit button, the input box and update button will appear. User can press on the home address input box to start setting home address.

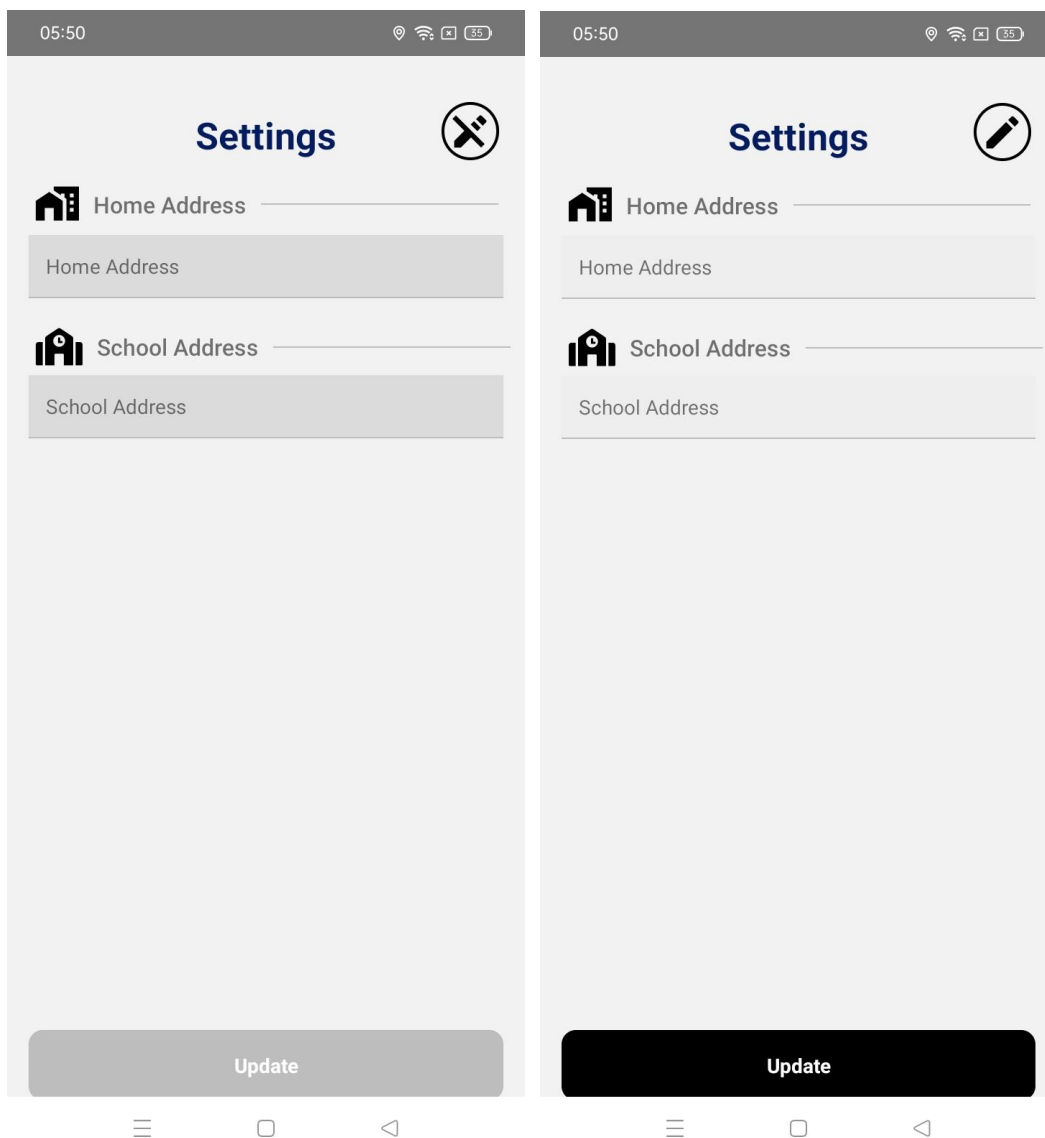


Figure 5.54 Module for verified users (Set Home Address)

When user press on the home address button, a screen with text input will prompt. The text input has integrated with Google Place Autocomplete feature which allow user to search for location based on Google’s map address.

User can also press on the “Current Location” to search for nearby location on the text input. After user have enter the location, user can press on the “Done” button.

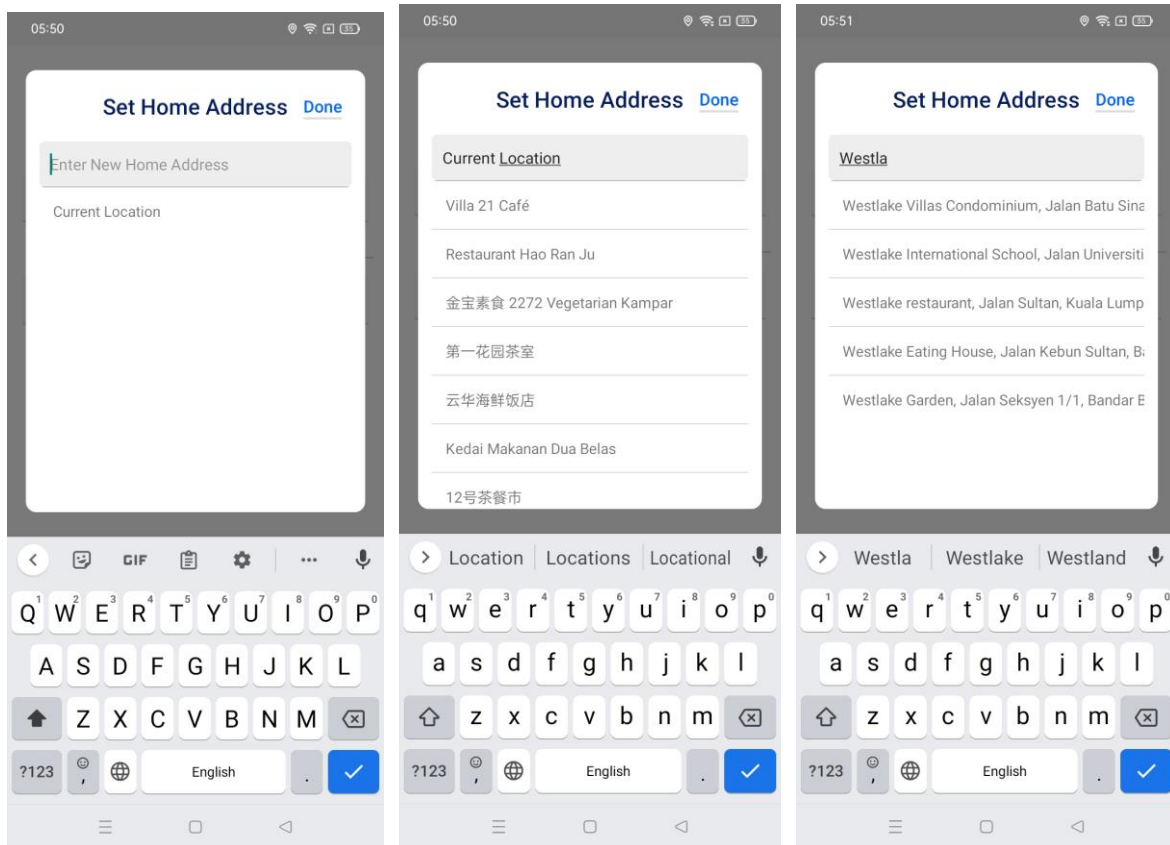


Figure 5.55 Module for verified users (Set Home Address)

After user have press on the “Done” button, the selected location’s name and address will appear on the home address column. User can now press on the update button to set the home address.

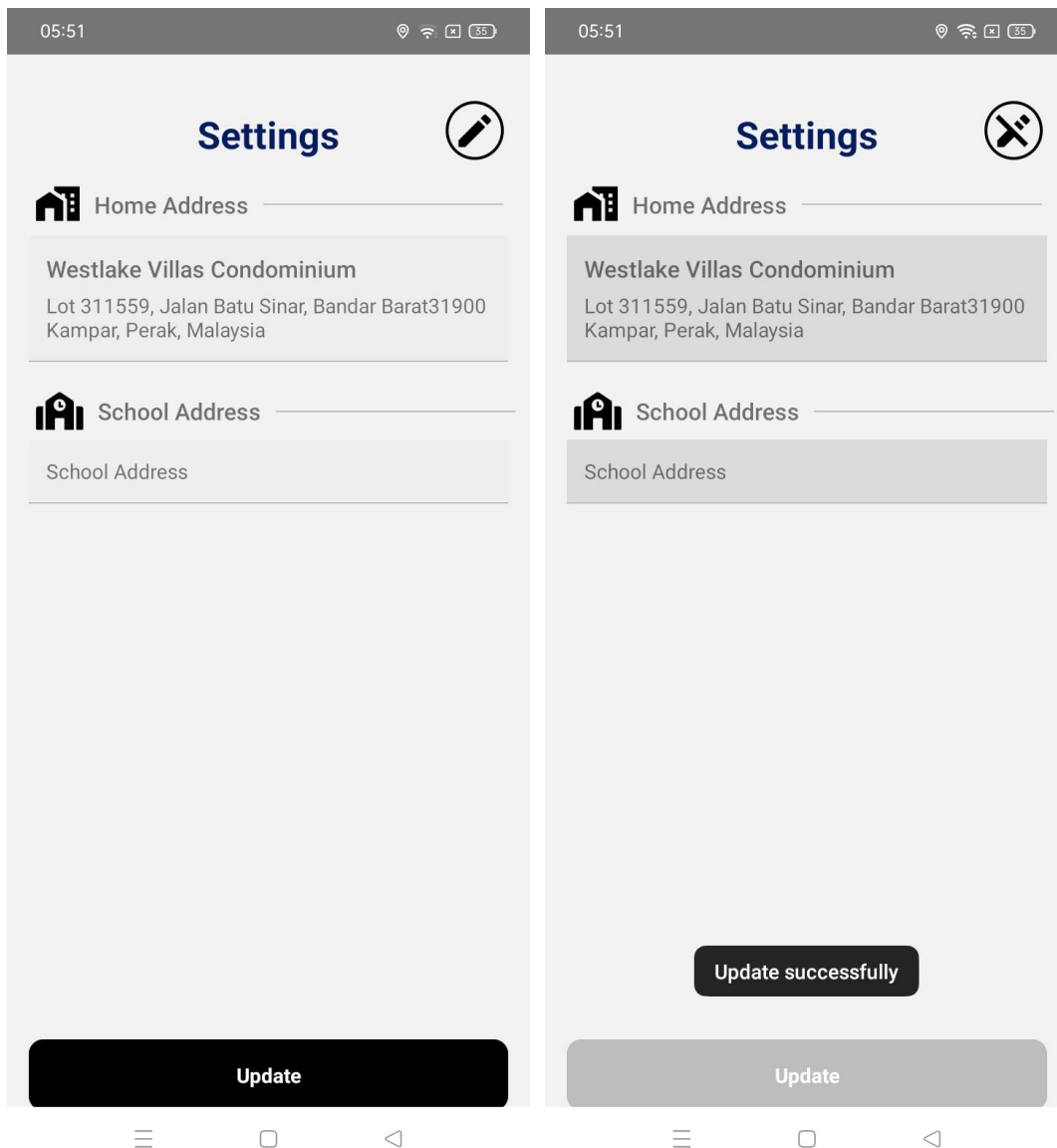


Figure 5.56 Module for verified users (Set Home Address)

5.3.4.11 Set School Address

Similar case to set home address, if user want to set school address, user can press on the top right edit button, the input box and update button will appear. User can press on the school address input box to start setting school address.

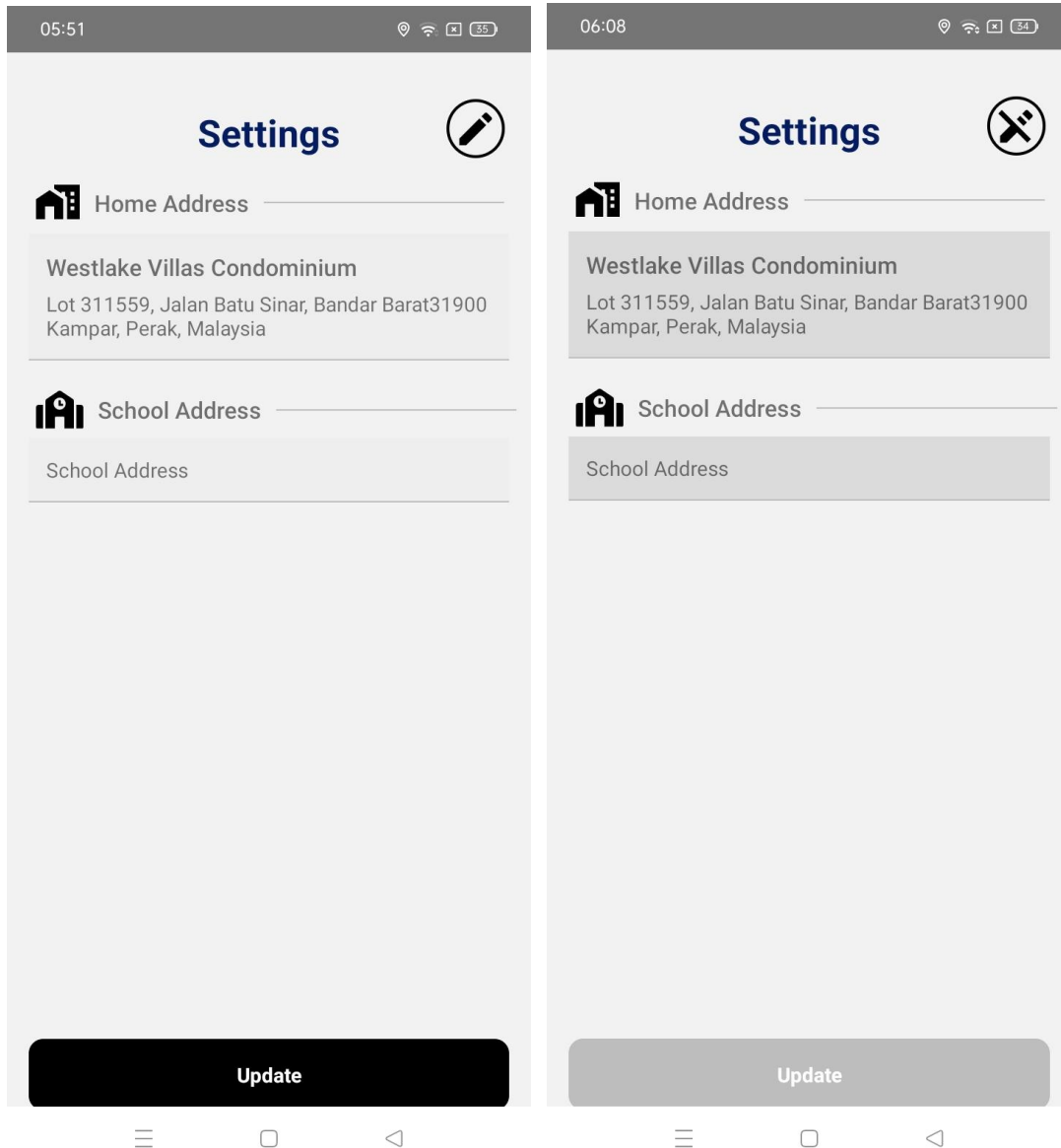


Figure 5.57 Module for verified users (Set School Address)

Chapter 5 System Implementation

When user press on the school address button, a screen with text input will prompt. The text input has integrated with Google Place Autocomplete feature which allow user to search for location based on Google's map address.

User can also press on the "Current Location" to search for nearby location on the text input. After user have enter the location, user can press on the "Done" button.

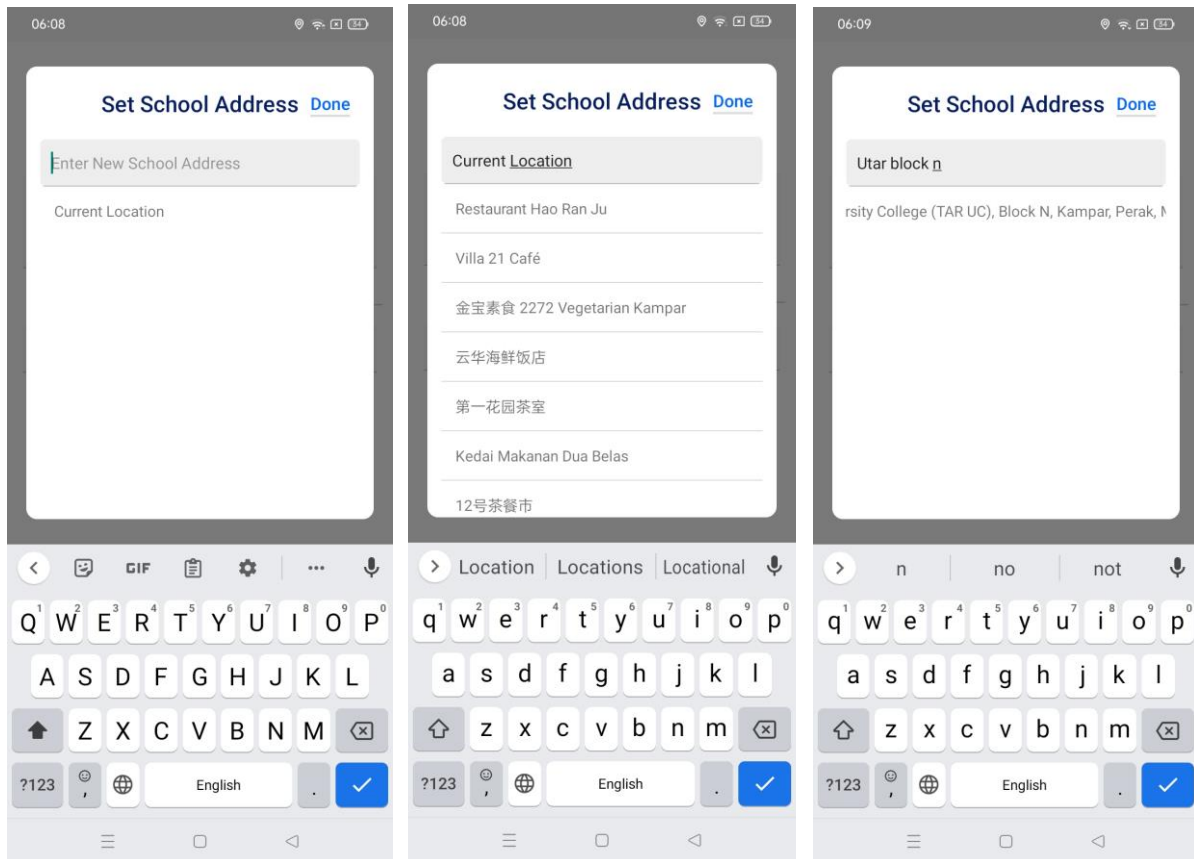


Figure 5.58 Module for verified users (Set School Address)

After user have press on the “Done” button, the selected location’s name and address will appear on the school address column. User can now press on the update button to set the school address.

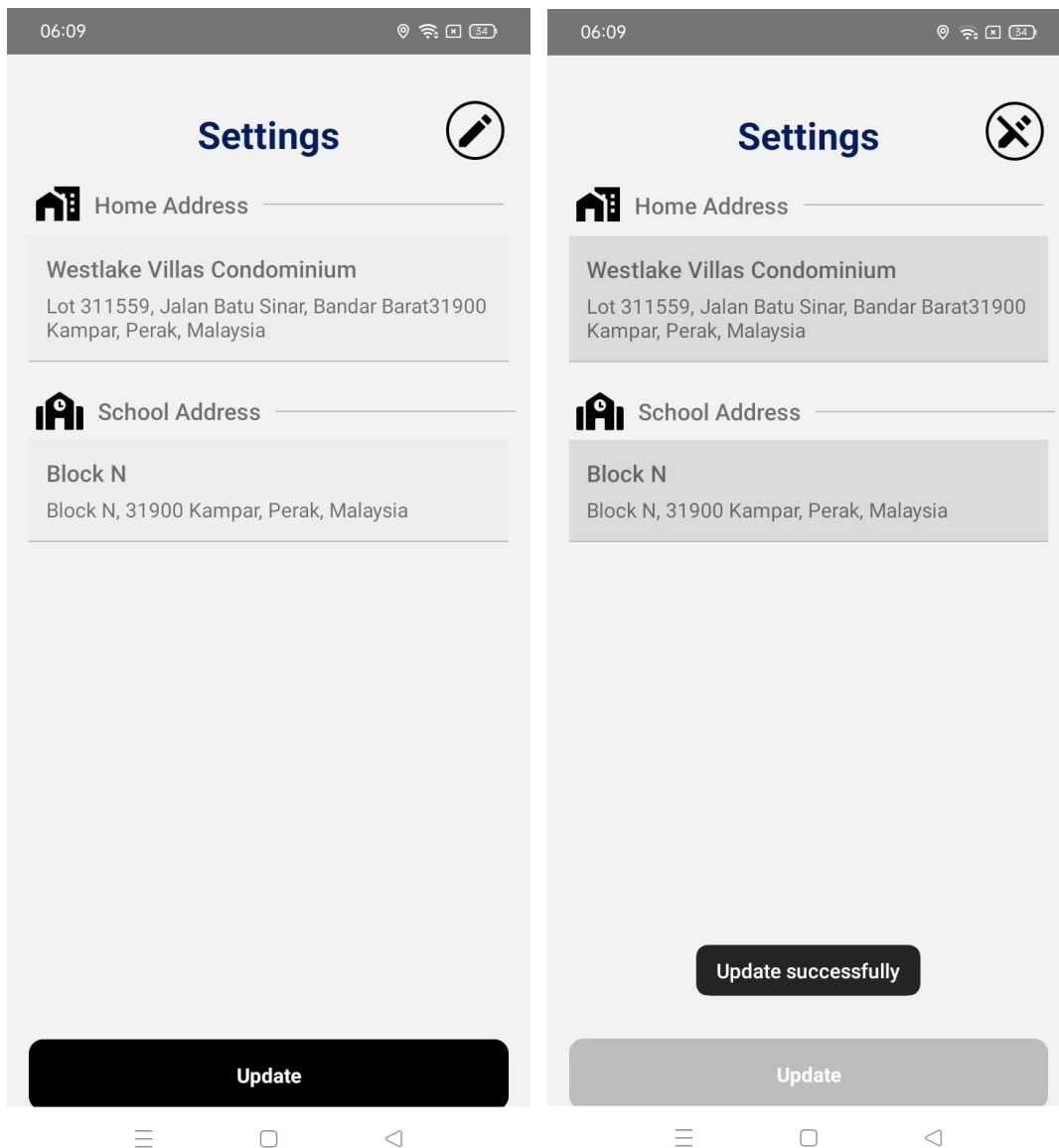


Figure 5.59 Module for verified users (Set School Address)

Chapter 5 System Implementation

After that, user will find that the home and school options will appear on the origin text input with Google Place Autocomplete features. User can use the predefined home and school address for future ride request posting.

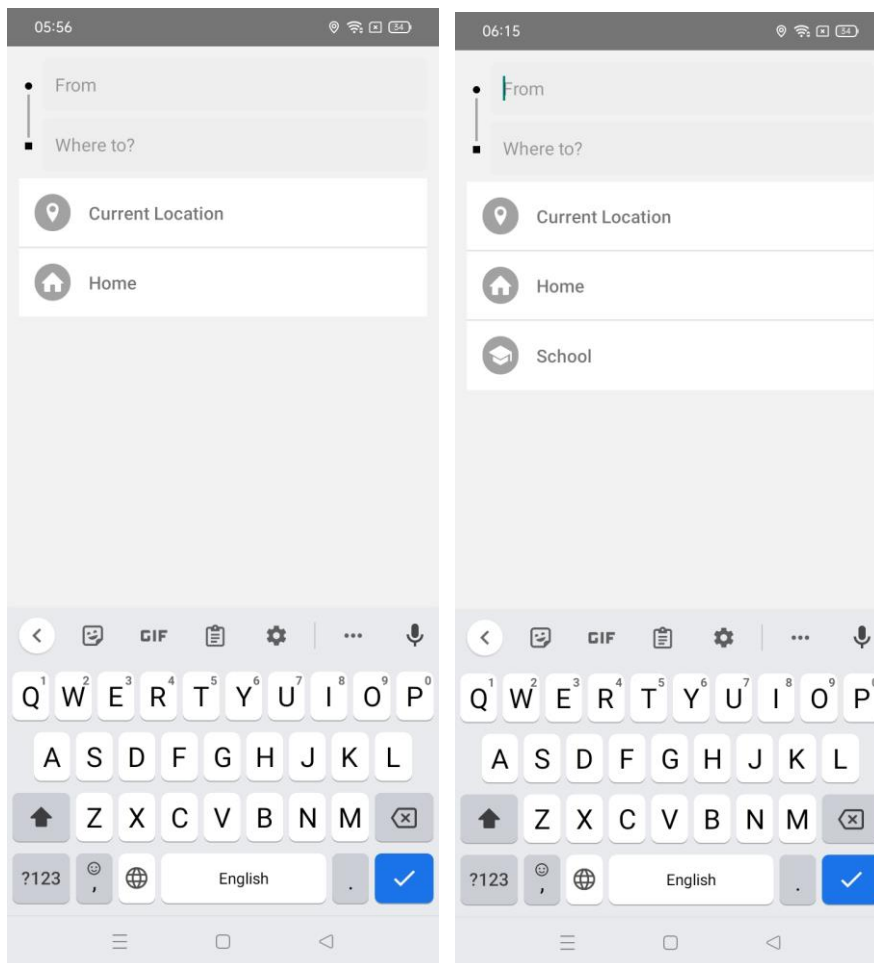


Figure 5.60 Module for verified users (Set School Address)

5.3.4 Module/Features for Users (not verified)

5.3.4.1 Post Ride Requests

If the user has not been verified yet, when user press on the “Where to” button on the home screen, a toast message will prompt to inform user that the user’s account has not been verified yet.

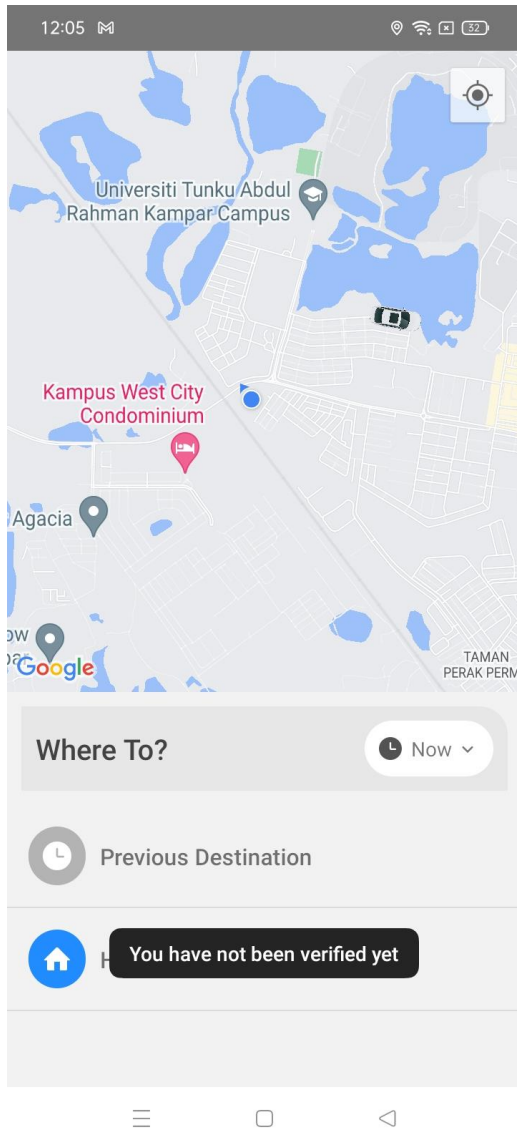


Figure 5.61 Module for not verified users (Post Ride Requests)

5.3.4.2 Request Ride Along

If the user has not been verified yet, when user press on the “Request to Ride Along” button on the driver’s ride request post details, a toast message will prompt to inform user that the user’s account has not been verified yet.

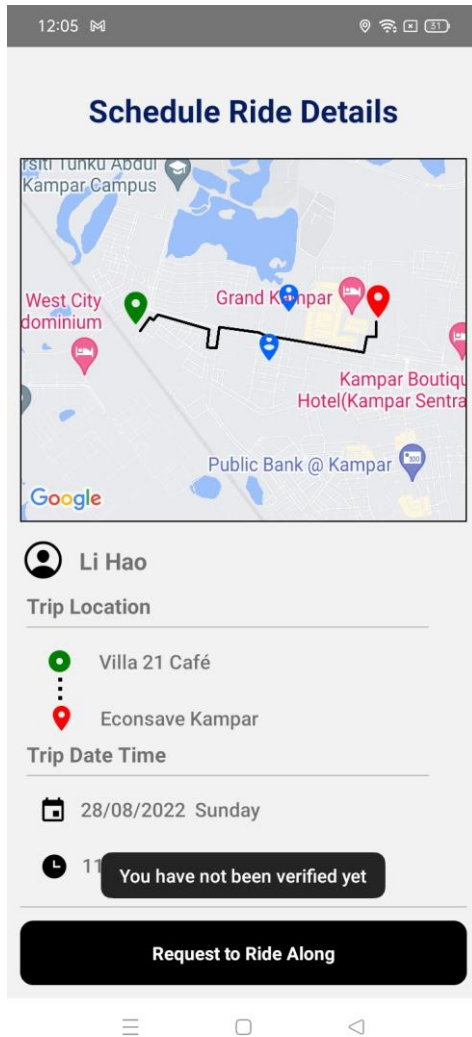


Figure 5.62 Module for not verified users (Request Ride Along)

5.3.4.3 Request Pick Up

If the user has not been verified yet, when user press on the “Request to Pick Up” button on the rider’s ride request post details, a toast message will prompt to inform user that the user’s account has not been verified yet.

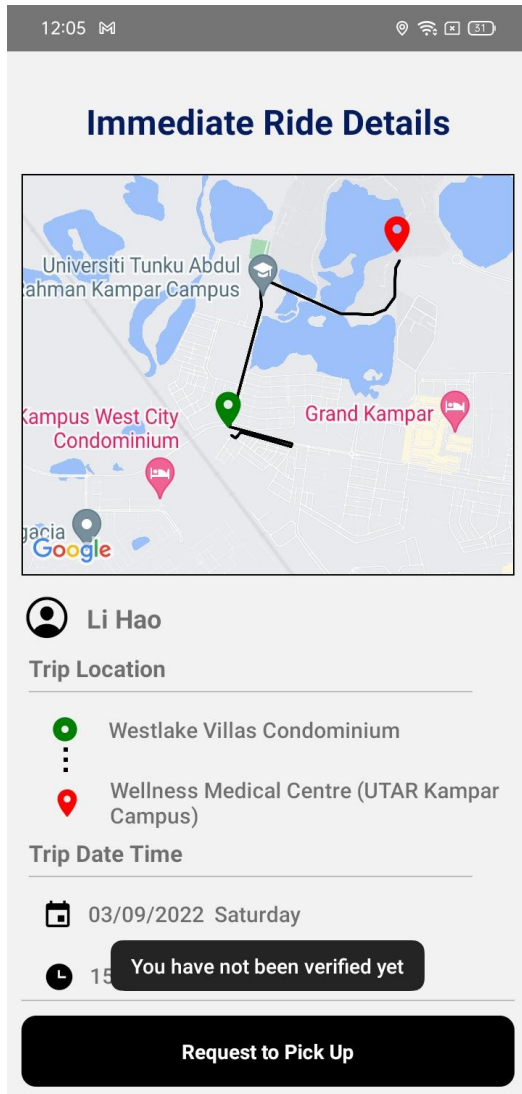


Figure 5.63 Module for not verified users (Request Pick Up)

5.4 Implementation Issues and Challenges

Development of an application is not always as easy as it seems though, as this is my first project to be implemented in my development career. I think the challenges may be insufficient knowledge or skills in mobile application development. Such a concern is raised during the implementation of the overall UI design of the carpooling application. Although React Native has lots of component libraries that are ready to be used in the application, but implementation is different compared to developing the component from scratch as the components are pre-created by React Native, which means they are not flexible when the application requires UI design that is different from the component design. However, such concerns were alleviated when I attempted to find some related information online. Some of the developers that are facing the same problem will share their solutions online, and some will even create a discussion section to help with the solutions.

Other challenges will be the implementation of the Google Maps APIs, as I have never implemented or used Google Maps in my project before. Therefore, during my implementation of the Google Maps API, I was confused as I had installed all the dependencies and components needed, but the autocomplete feature was not working. After some online surfing on the related issue, I have found that it is all because I did not enable billing on my Google Developer Account. After that, I try to find my supervisor to consult on whether our university has provided some access to the Google Developer Account for students. I have been told that I need to enable billing by providing my debit card details on my Google Developer Account to have full access as our university does not provide any access to Google Developer Account for students. I have also been told that I will not be charged as their free access limits are quite generous and should be enough for me to test my project.

Chapter 6

System Evaluation and Discussion

In this chapter we present the unit testing, database testing, user acceptance testing and usability testing in this project.

6.1 Unit Testing

The testing will be conducted by splitting the project into small and individual unit to be able to validate each of the functionality of the application, and to make sure all the functions is performed as requirement.

| Test Case No: 01 | | User: Rider/Driver | |
|---|---|---|-------------|
| Module: Login | | | |
| Test Case | Steps Involved | Expected Result | Status(P/F) |
| Login without input | Click Login button | Login fails with alert message. | (pass) |
| Login with incorrect email address | Enter email address Enter password Click Login Button | Login fails with alert message. | (pass) |
| Login with incorrect password | Enter email address Enter password Click Login Button | Login fails with alert message. | (pass) |
| Login with correct email address and password | Enter email address Enter password Click Login Button | Login successful and directed to home screen. | (pass) |

Table 6.1 Unit Testing for Rider/Driver (Login)

| Test Case No: 02 | | User: Rider/Driver | |
|--|--|---|--------------------|
| Module: Sign Up | | | |
| Test Case | Steps Involved | Expected Result | Status(P/F) |
| Sign Up without input | Click Register button | Sign Up fails with alert message. | (pass) |
| Sign Up with email address used | Enter email address Click Register Button | Sign Up fails with alert message. | (pass) |
| Sign Up with password not match with repeat password | Enter email address Enter password Click Register Button | Sign Up fails with alert message. | (pass) |
| Sign Up without student id image upload. | Upload student id image Click Register Button | Sign Up fails with alert message. | (pass) |
| Sign Up with username, email address, password, repeat password, phone number, and student id image uploaded | Enter username Enter email address Enter password Enter repeat password Enter phone number Upload student id Click Register Button | Sign Up successful and directed to home screen. | (pass) |

Table 6.2 Unit Testing for Rider/Driver (Sign Up)

| Test Case No: 03 | | User: Rider/Driver | |
|---|--|--|--------------------|
| Module: Forgot Password | | | |
| Test Case | Steps Involved | Expected Result | Status(P/F) |
| Reset password send without input | Click Send button | Password Reset fails with alert message. | (pass) |
| Reset password with incorrect email address | Enter email address Click Send Button | User will not receive any password reset link. | (pass) |
| Reset password with correct email address | Enter email address Click Send Button | User will receive a password reset link. | (pass) |

Table 6.3 Unit Testing for Rider/Driver (Forgot Password)

| Test Case No: 04 | | User: Rider/Driver | |
|---|--|--|--------------------|
| Module: Manage User Profile | | | |
| Pre-condition: User have logged in successfully. | | | |
| Test Case | Steps Involved | Expected Result | Status(P/F) |
| View user profile details | Click Profile button on Drawer Menu | User details such as username, email address, verified status, and phone number will be display. | (pass) |
| Enable editing state to edit details | Click edit Button on the top right corner | User will see input box and update button are visible. | (pass) |
| Update phone number with number input | Enter new phone number in the input box Click Update button | A toast message will display update successfully. The phone number value been updated. | (pass) |

Table 6.4 Unit Testing for Rider/Driver (Manage User Profile)

| Test Case No: 05 | | User: Rider/Driver | |
|---|--|--|--------------------|
| Module: Check Rides Requested | | | |
| Pre-condition: User have logged in successfully. | | | |
| Test Case | Steps Involved | Expected Result | Status(P/F) |
| View Rides Requested | Click My Rides button on Drawer Menu | User can view their ride request post by switch between Rider or Driver role. Ride request categories base on different carpool services. | (pass) |
| Switch between Rider and Driver role | Click toggle button on top right corner | User can view their ride request post by switch between Rider or Driver role. been categories base on different carpool services. | (pass) |
| Switch between immediate, schedule and regular ride | Click on the ride services button | User can view their ride request post by switch between immediate, schedule and regular ride service. | (pass) |
| Press on ride request to view ride details | Click on ride request | User can view the ride request details such as route map, origin and destination address, date, time and requested riders' name. The requested pick-up location pin show on the route map. | (pass) |
| Contact requested riders. | Click on WhatsApp icon on the right of requested riders' name. | A reminder alert box will prompt for user confirmation. | (pass) |
| Confirm understand the reminder | Click OK on reminder alert box | User has confirmed they understand the reminder and | (pass) |

| | | | |
|--|--|--|--|
| | | will be redirected to WhatsApp link to contact requested riders. | |
|--|--|--|--|

Table 6.5 Unit Testing for Rider/Driver (Check Rides Requested)

| Test Case No: 06 | | User: Rider/Driver | |
|---|--|--|--------------------|
| Module: Post Ride Request | | | |
| Pre-condition: User have logged in successfully with verified account. | | | |
| Test Case | Steps Involved | Expected Result | Status(P/F) |
| Navigate to origin and destination input screen | Click Where to button on home screen | User redirected to origin and destination input screen | (pass) |
| Input origin and list of related results display | Enter location name on origin input | User allowed to enter location name in the origin input. List of location related will be displayed. | (pass) |
| Input destination and list of related results display | Enter location name on destination input | User allowed to enter location name in the destination input. List of location related will be displayed. | (pass) |
| Navigate to route map and role selection screen | Entered both origin and destination location | After user enter both origin and destination input, user automatically navigate to route map and role selection screen | (pass) |
| Select rider or driver role | Click on Rider or Driver button checked. | User allowed to choose rider or driver role. | (pass) |
| Navigate to ride request details screen | Click on schedule button | After user choose a role, user click on schedule button to | (pass) |

| | | | |
|-------------------------|--|---|--------|
| | | proceed and navigate to ride request details screen. | |
| Select one ride service | Click on schedule, regular or immediate button | User can view the origin and destination name and address. User can choose one carpool service, either schedule, regular or immediate. | (pass) |
| Pick date and time | Click on Pick a Date and Pick a Time button | User can choose date and time for the ride request. | (pass) |

Table 6.6 Unit Testing for Rider/Driver (Post Ride Request)

| Test Case No: 07 | | User: Rider/Driver | |
|--------------------------------------|--|--|-------------|
| Module: Set Home Address | | | |
| Test Case | Steps Involved | Expected Result | Status(P/F) |
| View home address and school address | Click Settings button on Drawer Menu | User can view home address and school address displayed | (pass) |
| Enable editing state to edit details | Click edit Button on the top right corner | User will see input box and update button are visible. | (pass) |
| Edit home address | Click home address input box Enter new home address | An input screen will pop up. User can click on the input box to enter new home address | (pass) |
| Update with new home address | Click Update button | A toast message will display update successfully. The home address value will be updated. | (pass) |

Table 6.7 Unit Testing for Rider/Driver (Set Home Address)

| Test Case No: 08 | | User: Rider/Driver | |
|--------------------------------------|--|--|--------------------|
| Module: Set School Address | | | |
| Test Case | Steps Involved | Expected Result | Status(P/F) |
| View home address and school address | Click Settings button on Drawer Menu | User can view home address and school address displayed | (pass) |
| Enable editing state to edit details | Click edit Button on the top right corner | User will see input box and update button are visible. | (pass) |
| Edit school address | Click school address input box Enter new school address | An input screen will pop up. User can click on the input box to enter new school address | (pass) |
| Update with new school address | Click Update button | A toast message will display update successfully. The school address value will be updated. | (pass) |

Table 6.8 Unit Testing for Rider/Driver (Set School Address)

| Test Case No: 09 | | User: Rider | |
|---|--|--|--------------------|
| Module: Request Ride Along | | | |
| Pre-condition: User have logged in successfully with verified account. | | | |
| Test Case | Steps Involved | Expected Result | Status(P/F) |
| View ride request posts | Click Search Rides button on Drawer Menu | User can view their ride request post by switch between Rider or Driver role. Ride request categories base on different carpool services. | (pass) |
| View driver's ride request details | Click on driver's ride request post | User can view the ride request details such as route map, origin and destination address, date, time and requested riders' name. The requested pick-up location pin show on the route map. | (pass) |
| Request to ride along | Click on Request to Ride Along button | A map view will be popped up with a Select Location button. | (pass) |
| Select location to be pick up | Click on Select Location button | A location pin will appear on the map view. | (pass) |
| Select location to be pick up | Long press and drag the location pin | Long press and drag the location pin to select a location to be picked up along the driver's route. | (pass) |
| Select location to be pick up | Click Done button | Navigate back to the ride request details page with the location pin selected rendered | (pass) |
| Select location to be pick up | Click Submit button | A reminder message box will prompt to remind user safety | (pass) |

| | | | |
|-------------------------------|-----------------|--|--------|
| | | always come first when riding with strangers | |
| Select location to be pick up | Click OK button | A toast message will prompt to inform user that the ride along request has successfully been submitted | (pass) |

Table 6.9 Unit Testing for Rider (Request Ride Along)

| Test Case No: 10 | | User: Driver | |
|---|--|---|-------------|
| Module: Request Pick Up | | | |
| Pre-condition: User have logged in successfully with verified account. | | | |
| Test Case | Steps Involved | Expected Result | Status(P/F) |
| View ride request posts | Click Search Rides button on Drawer Menu | User can view their ride request post by switch between Rider or Driver role. Ride request categories base on different carpool services. | (pass) |
| View rider's ride request details | Click on rider's ride request post | User can view the ride request details such as route map, origin and destination address, date, time and requested drivers' name. | (pass) |
| Request to pick up | Click on Request to Pick Up button | A reminder message box will prompt to remind user safety always come first when riding with strangers | (pass) |
| Select location to be pick up | Click OK button | A toast message will prompt to inform user that the pickup request has successfully been submitted | (pass) |

Table 6.10 Unit Testing for Driver (Request Pick Up)

6.2 Database Testing

The database used in this project include Firebase Authentication, Firestore Database, Firebase Storage.

| | |
|---|---|
| Test Case No: 1 | |
| Database: Firebase Storage | |
| Condition: User sign up a new account and upload student id image. | |
| Data | image |
| Description | Upload student id image to Firebase Storage and generate download URL. |
| Expected Result | Register successfully Student id image uploaded to storage, generate download URL. |
| Status(P/F) | (Pass) |

Table 6.11 Database Testing for Firebase Storage (sign up)

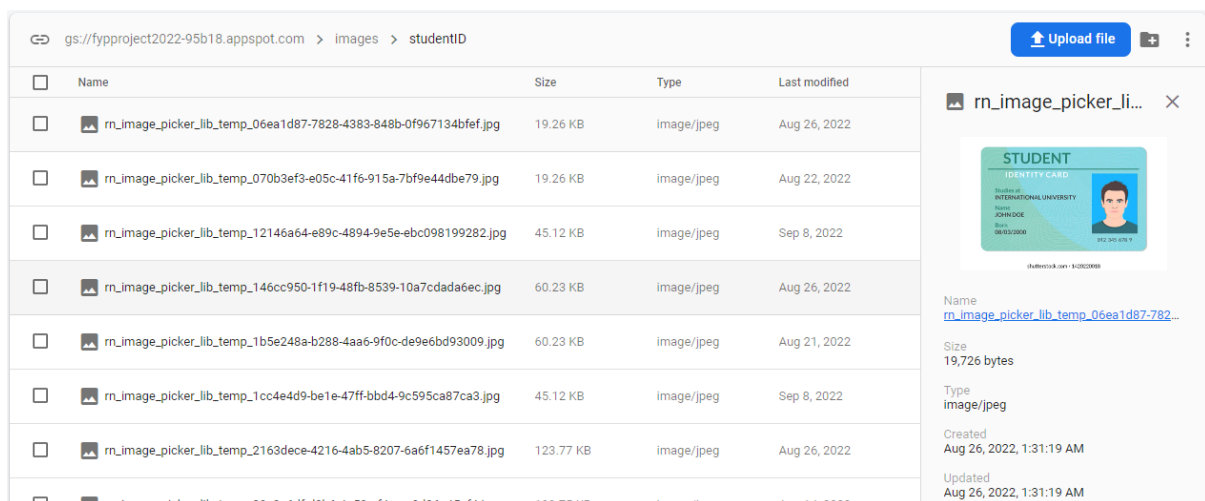


Figure 6.1 Firebase Storage Console

| | |
|---|--|
| Test Case No: 1 | |
| Database: Firebase Authentication | |
| Condition: User sign up a new account. | |
| Data | email address, password |
| Description | Send email address, password to create account. |
| Expected Result | Register successfully Email address and password uploaded to database |
| Status(P/F) | (Pass) |

Table 6.12 Database Testing for Firebase Authentication (sign up)

| Identifier | Providers | Created ↓ | Signed In | User UID |
|-----------------------|-----------|--------------|--------------|-------------------------------|
| tanmeimei@1utar.my | ✉ | Sep 8, 2022 | Sep 8, 2022 | ajxmW2Cn9WP7e7Qv6Pg3PG7gb... |
| usertest@gmail.com | ✉ | Aug 27, 2022 | Aug 30, 2022 | vmrIfA7UYhVaY0Vm5zNUxUFpULr2 |
| tanmei@1utar.my | ✉ | Aug 26, 2022 | Sep 8, 2022 | n9U3Pq043nWyAyzDvo7hGGkNwl... |
| 2000leohokr@gmail.com | ✉ | Aug 26, 2022 | Aug 26, 2022 | EmlmoSZQZDgJFvUmJdoG0cqVa... |
| tanfatt@1utar.my | ✉ | Aug 26, 2022 | Sep 8, 2022 | G8FnICPFD2Qf7W8Je8GmXVFNLy1 |
| changlihao@1utar.my | ✉ | Aug 26, 2022 | Sep 8, 2022 | JIBEBh4FA4NkeHENxk4ojcmJsnV2 |

Figure 6.2 Firebase Authentication Console

| | |
|---|---|
| Test Case No: 1 | |
| Database: Firestore Database | |
| Condition: User sign up a new account. | |
| Data | email address, username, verified, imageURL, phoneNum, homePlace, schoolPlace |
| Description | Upload email address, username, verified, imageURL, phoneNum, homePlace, schoolPlace to Firestore database to store user's details. |
| Expected Result | Register successfully Email address, username, verified, imageURL, phoneNum, homePlace, schoolPlace uploaded to database |
| Status(P/F) | (Pass) |

Table 6.12 Database Testing for Firestore Database (sign up)

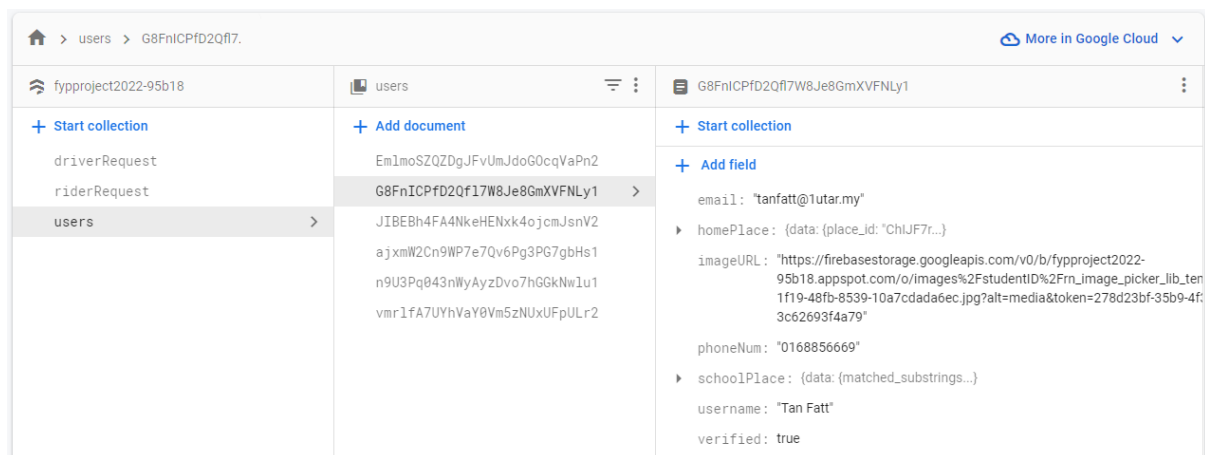


Figure 6.3 Firestore Database Console (users)

| | |
|--|---|
| Test Case No: 2 | |
| Database: Firestore Database | |
| Condition: User post rider request. | |
| Data | <i>userID, originName, originAddress, destinationName, destinationAddress, originLatitude, originLongitude, destinationLatitude, destinationLongitude, dateRider, timeRider, immRideState, schRideState, regRideState, regFrom, regUntill</i> |
| Description | Upload <i>userID, originName, originAddress, destinationName, destinationAddress, originLatitude, originLongitude, destinationLatitude, destinationLongitude, dateRider, timeRider, immRideState, schRideState, regRideState, regFrom, regUntill</i> to Firestore database to store user's rider request details. |
| Expected Result | Request successfully <i>userID, originName, originAddress, destinationName, destinationAddress, originLatitude, originLongitude, destinationLatitude, destinationLongitude, dateRider, timeRider, immRideState, schRideState, regRideState, regFrom, regUntill</i> uploaded to database |
| Status(P/F) | (Pass) |

Table 6.13 Database Testing for Firestore Database (post rider request)

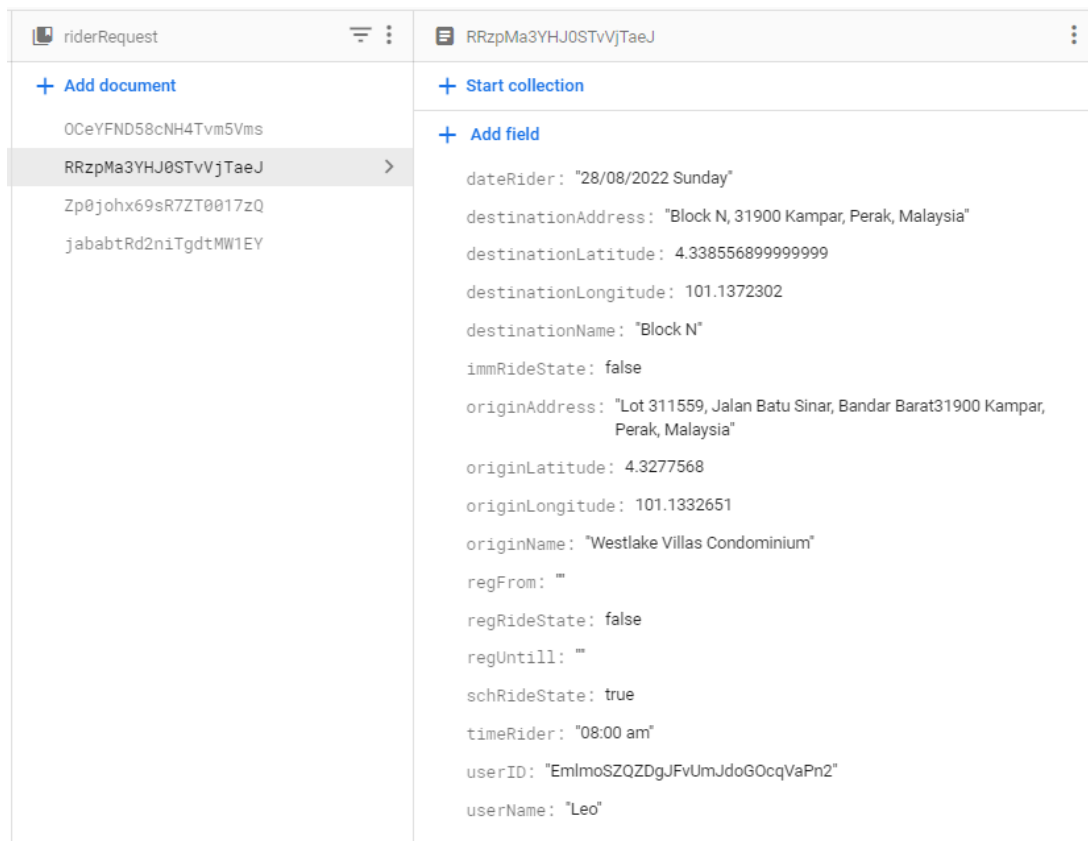


Figure 6.4 Firestore Database Console (riderRequest)

| | |
|---|---|
| Test Case No: 3 | |
| Database: Firestore Database | |
| Condition: User post driver request. | |
| Data | <i>userID, userName, originName, originAddress, destinationName, destinationAddress, originLatitude, originLongitude, destinationLatitude, destinationLongitude, dateDriver, timeDriver, immRideState, schRideState, regRideState, regFrom, regUntill</i> |
| Description | Upload <i>userID, userName, originName, originAddress, destinationName, destinationAddress, originLatitude, originLongitude, destinationLatitude, destinationLongitude, dateDriver, timeDriver, immRideState, schRideState, regRideState, regFrom, regUntill</i> to Firestore database to store user's rider request details. |
| Expected Result | Request successfully <i>userID, userName, originName, originAddress, destinationName, destinationAddress, originLatitude, originLongitude, destinationLatitude, destinationLongitude, dateDriver, timeDriver, immRideState, schRideState, regRideState, regFrom, regUntill</i> uploaded to database |
| Status(P/F) | (Pass) |

Table 6.14 Database Testing for Firestore Database (post driver request)

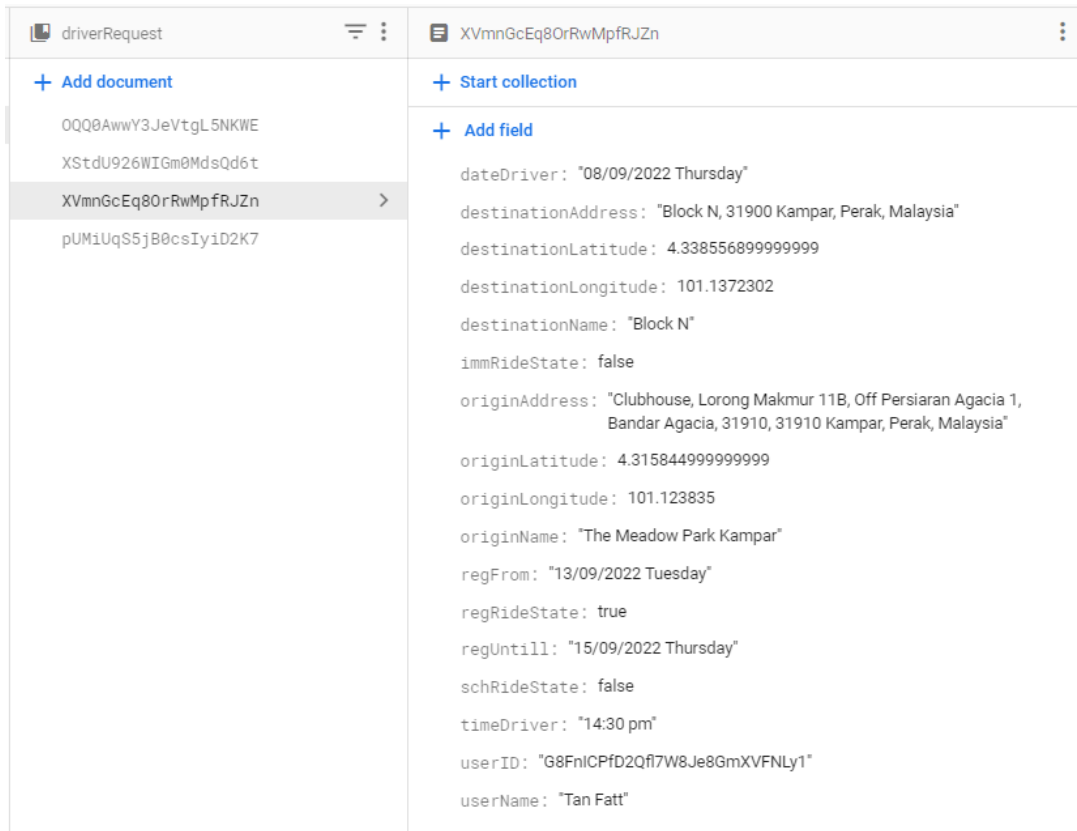


Figure 6.5 Firestore Database Console (driverRequest)

| | |
|--|--|
| Test Case No: 4 | |
| Database: Firestore Database | |
| Condition: Rider request to ride along. | |
| Data | <i>raUserID, raUserName, raUserPhone, locationPin</i> |
| Description | Upload <i>raUserID, raUserName, raUserPhone, locationPin</i> to Firestore database to store rider request ride along data to the database. |
| Expected Result | Request successfully <i>raUserID, raUserName, raUserPhone, locationPin</i> uploaded to database |
| Status(P/F) | (Pass) |

Table 6.15 Database Testing for Firestore Database (request ride along)

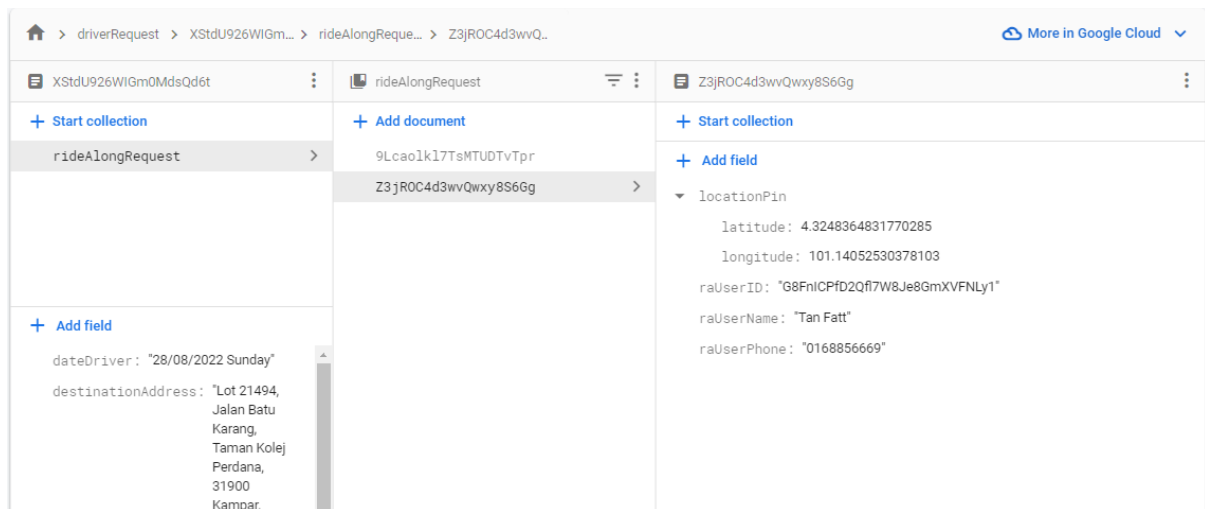


Figure 6.6 Firestore Database Console (rideAlongRequest)

| | |
|--|---|
| Test Case No: 5 | |
| Database: Firestore Database | |
| Condition: Driver request to pick up. | |
| Data | <i>raUserID, raUserName, raUserPhone</i> |
| Description | Upload <i>raUserID, raUserName, raUserPhone</i> to Firestore database to store rider request ride along data to the database. |
| Expected Result | Request successfully <i>raUserID, raUserName, raUserPhone</i> uploaded to database |
| Status(P/F) | (Pass) |

Table 6.16 Database Testing for Firestore Database (request pick up)

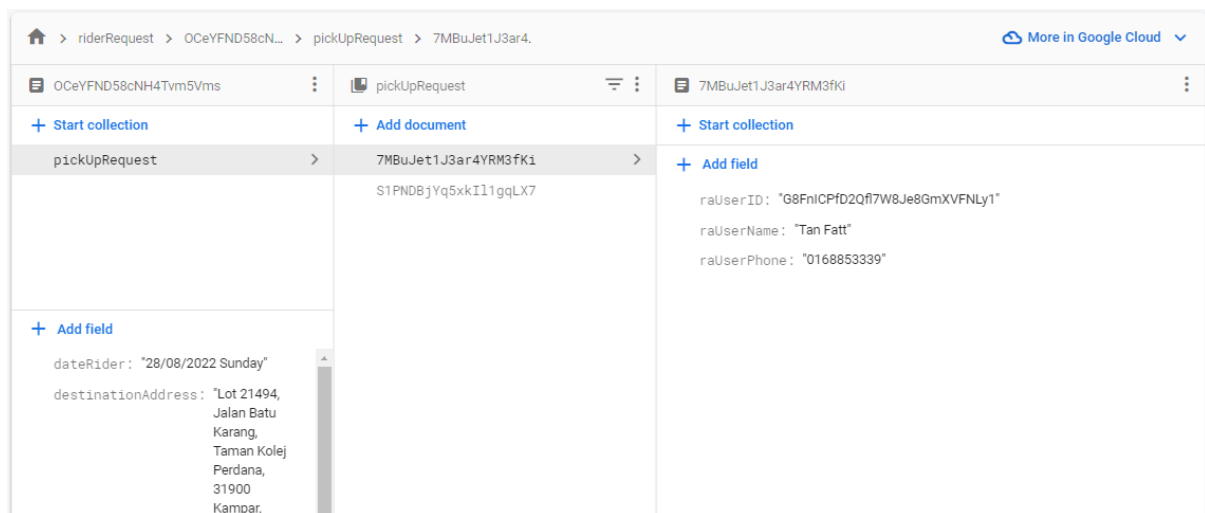


Figure 6.7 Firestore Database Console (pickUpRequest)

6.3 User Acceptance Testing (UAT)

Since this application is design for university students, I have invited 5 UTAR students to take part of the user acceptance test. Due to the application are designed for android device only, 5 of the participants who are also using android smart phone will install the APK file of this application.

6.3.1 UAT Test Scenario

I have prepared 7 activities scenario for the participants which simulate how the students will interact with the application. Participants can also explore the application themselves.

| | |
|-----------------------------|---|
| Scenario | <ol style="list-style-type: none"> 1. You would like to register an account. Details required: Username Email Address Password Phone Number Student ID 2. You would like to login into the application. 3. You would like to log out from the application. |
| Result | |
| Comment (if any) | |

Table 6.17 UAT Test scenario (Sign Up, Login and Logout)

| | |
|-------------------------|--|
| Scenario | <ol style="list-style-type: none"> 1. You would like to post a ride request. <ol style="list-style-type: none"> a. Home Screen b. Enter origin and destination c. Select rider or driver profile d. Select carpool service (immediate, schedule, regular) e. Enter date f. Enter time g. Read on the reminder h. Submit request 2. You would like to check for your ride request posting. |
| Result | |
| Comment (if any) | |

Table 6.18 UAT Test scenario (Post Ride Request and Check Rides Requested)

| | |
|-------------------------|--|
| Scenario | <ol style="list-style-type: none"> 1. You are rider, you want to request for a ride from driver. <ol style="list-style-type: none"> a. Drawer Menu b. Search Rides c. Check on driver request posting d. Select a carpool service (immediate, schedule, regular) e. Select a ride request post f. Check on ride request details (origin, destination, date, time) g. Click on Request to Ride Along h. Click on Select Location i. Long press and drag the location pin to your place j. Click Done k. Read on the reminder l. Submit Request 2. Wait for the driver to contact you for further ride arrangement. |
| Result | |
| Comment (if any) | |

Table 6.19 UAT Test scenario (Request Ride Along)

| | |
|-----------------------------|--|
| Scenario | <ol style="list-style-type: none"> 1. You are driver, you want to request to pick up riders. <ol style="list-style-type: none"> a. Drawer Menu b. Search Rides c. Check on rider request posting d. Select a carpool service (immediate, schedule, regular) e. Select a ride request post f. Check on ride request details (origin, destination, date, time) g. Click on Request to Pick Up h. Read on the reminder i. Submit Request 2. Wait for the rider to contact you for further ride arrangement. |
| Result | |
| Comment (if any) | |

Table 6.20 UAT Test scenario (Request Pick Up)

| | |
|-----------------------------|--|
| Scenario | <ol style="list-style-type: none"> 1. You forgot your password. 2. You would like to reset your account password. 3. You check on your email for the password reset link. 4. You try to log in again with your new password. |
| Result | |
| Comment (if any) | |

Table 6.21 UAT Test scenario (Forgot Password)

| | |
|-------------------------|--|
| Scenario | <ol style="list-style-type: none"> 1. You changed your phone number recently. 2. You would like to edit your account profile details. 3. You would like to edit your phone number on profile details. |
| Result | |
| Comment (if any) | |

Table 6.22 UAT Test scenario (Manage User Profile)

| | |
|-------------------------|--|
| Scenario | <ol style="list-style-type: none"> 1. You have been using the application regularly. 2. You would like to add your home address to the account. 3. You would like to add your school address to the account. 4. You post a ride request with you home address. |
| Result | |
| Comment (if any) | |

Table 6.23 UAT Test scenario (Set Home and School address)

6.3.2 UAT Test Result

The test will be release for 2 days, as a result, all the participants are able to understand all the user interface and functions of the application. At the end, all the participants will require to give a rating from 1 to 5 on each of the functionality of the application.

| Function/Feature | Number of Test Conducted | Average Rating (1-5) | Comments Gathered (if any) |
|---------------------------|--------------------------|----------------------|---|
| User Auth | 5 | 4 | |
| Post Ride Request | 5 | 4.33 | |
| Manage User Profile | 5 | 3.5 | 1. Hope can edit more personal details. |
| Check My Rides | 5 | 4.5 | |
| Search Rides | 5 | 4 | |
| Request Ride Along | 5 | 4.5 | 1. The request function is interesting and easy to use. 2. The reminder is a nice touch. |
| Request Pick Up | 5 | 4 | |
| Set home & school address | 5 | 4 | |

Table 6.24 UAT Test Result

6.4 Usability Testing

After conducted the UAT test, all the participants are also requested to fill up a simple survey form that include 10 questions to evaluate their experience with the application.

6.4.1 Survey Form

| | 1 Strongly Disagree | 2 Disagree | 3 OK | 4 Agree | 5 Strongly Agree |
|---|---------------------------|---------------|---------|------------|------------------------|
| 1. You would like to use DinoPool every day. | | | | | |
| 2. You think DinoPool is easy to use. | | | | | |
| 3. You think DinoPool is safe to use. | | | | | |
| 4. You think the user interface is vivid and interesting. | | | | | |
| 5. You think the map view is easy to understand. | | | | | |
| 6. You think location input is easy to use. | | | | | |
| 7. You think the carpool service offered can fulfil everyone needs. | | | | | |
| 8. You think upload student id is necessary. | | | | | |
| 9. You think the application is responsive. | | | | | |
| 10. You feel excited to share ride with strangers. | | | | | |
| Comments (if applicable): | | | | | |

Table 6.25 Usability Test (Survey Form)

6.4.2 Survey Result

Overall, from the survey we know that most of the participants are happy with the functionality provided from the application.

| Question | Participant | | | | | Average |
|----------|-------------|-----|-----|-----|-----|---------|
| | 1st | 2nd | 3rd | 4th | 5th | |
| 1 | 4 | 2 | 3 | 3 | 4 | 3.2 |
| 2 | 4 | 3 | 3 | 3 | 4 | 3.4 |
| 3 | 3 | 4 | 4 | 3 | 4 | 3.6 |
| 4 | 3 | 2 | 3 | 3 | 3 | 2.8 |
| 5 | 4 | 3 | 4 | 4 | 3 | 3.6 |
| 6 | 3 | 3 | 3 | 3 | 4 | 3.2 |
| 7 | 3 | 4 | 4 | 3 | 3 | 3.4 |
| 8 | 3 | 3 | 4 | 2 | 3 | 3.6 |
| 9 | 3 | 3 | 4 | 4 | 4 | 3.6 |
| 10 | 3 | 3 | 3 | 3 | 4 | 3.2 |
| Total | 33 | 30 | 35 | 31 | 36 | 33.60 |
| Percent% | 66% | 60% | 70% | 62% | 72% | 67% |

Table 6.25 Usability Test (Survey Result)

Chapter 7

Conclusion and Recommendation

In this chapter we present the conclusion and recommendation for future work in this project.

7.1 Conclusion

To conclude, this project will be development-based, and a carpooling application will be developed to solve the existing problems. The project has been in progress since the October 2021 trimester, beginning with the creation of the project proposal and continuing for approximately a year. The deliverable of this project is to help university students by providing them with an alternative way to commute during their daily routine. We also aim to provide a safe and secure environment for university students to enjoy carpooling services by identifying all users' identification before they can request a service. In addition, we hope to bring the benefits of carpooling to all university students, such as providing an affordable ride based on shared costs, reducing one's carbon footprint, and providing a carpooling platform that allows users to connect with others during the ride, which also contributes to positive mental health.

It is important to have a good planning on a development project, therefore agile methodology has been chosen for this project as it is more suitable for rapid development and changing requirement. An iterative process will be used to review the application when a prototype has been developed, it allows to develop the application in a cyclical way, building small parts in an evolving way, instead of starting with a complete understanding of the requirements.

This project will be using a non-cost method, which means that the application will not implement any payment features for the users, either drivers or riders in the application. The special feature of this project is that users need to be verified to use the carpooling service, and the administrator will review the uploaded student IDs and decide which user needs to have a verified account. The application will be developed using the React Native framework, as NPM includes many ready-made libraries that will make the development process faster. This will be an online application, which always needs an internet connection

Chapter 7 Conclusion and Recommendation

to receive or send data to the cloud database. The application will also require using user's geolocation which the application will ask to access user's device location service. The main delivery platform for this project will be the Android platform, as it has more users and greater flexibility in application selection. In terms of functionality, the application will provide several key features such as user authentication, managing user profiles, posting ride requests, checking the status of requested rides, etc. The name of this application will be DinoPool, and its cute application icon will attract the user's attention while providing more simplicity in the user interface. The user will be further divided into two different roles, namely: rider or driver. Both roles will have the same interface, but with slightly different functions. The rider can request a ride on the driver's route, while the driver can request to pick up the passenger at the passenger's origin. The app will offer three carpooling services, namely instant, scheduled, and regular; each service will differ in terms of time requirements and methods. While solo travel is still popular in Malaysia, we hope to promote the concept of carpooling to millennials, who are college students who are already practically immersed in technology and are more willing to interact with it.

7.2 Recommendation for future work

Due to the limited development time, some of the features proposed have not been able to be implemented yet on this project, such as the notification and rating system, which I think is crucial to have in a carpooling application. I hope to be able to implement the features in the future to provide better user experience for the target audience.

Besides, although the application has been proposed to cater to most of the needs of commute for university students and has proven to satisfy most of the users, there is still some improvement that needs to be done in the future to provide better features. There are some features which I think can be included in future work:

1. Reminder to ask the user to turn on location services.
2. Provide in-app messaging features.
3. Stop the application screen from rotating automatically.
4. Provide refer-a-friends features to reach more target audiences.

References

- [1] R. Manzini and A. Pareschi, "A decision-support system for the car pooling problem," *Journal of Transportation Technologies*, vol. 2, pp. 85-101, 2012.
- [2] Calvo, Roberto Wolfler, Fabio de Luigi, Palle Haastrup and Vittorio Maniezzo, "A distributed geographic information system for the daily car pooling problem," *Computers & Operations Research*, vol. 31, pp. 2263-2278, 2004.
- [3] L. Jianling, Patrick Embry, P. M. Stephen, Kaveh Farokhi Sadabadi, Isaradatta Rasmidatta and Mark W. Burris, "Who chooses to carpool and why? Examination of Texas carpoolers," *Transportation Research Record*, vol. 1, pp. 110-117, 2007.
- [4] Dorinson, Diana, Deanna Gay, Paul Minett and Susan Shaheen, "Flexible carpooling: exploratory study," 2009.
- [5] Kladeftiras, George and Constantinos Antoniou, "Social networks' impact on carpooling systems performance: Privacy vs. efficiency," Vols. 15-1540, 2015.
- [6] Aïvodji, Ulrich Matchi, Sébastien Gambs, Marie-José Huguet and Marc-Olivier Killijian, "Privacy-preserving carpooling," *In Odysseus 2015-6th International Workshop on Freight Transportation and Logistics*, 2015.
- [7] J. Friginal, Sébastien Gambs, Jérémie Guiochet and Marc-Olivier Killijian, "Towards privacy-driven design of a dynamic carpooling system," *Pervasive and mobile computing*, vol. 14, pp. 71-82, 2014.
- [8] P. Minett, Albany Village and John Pearce, "Flexible carpooling: challenging the ride matching paradigm, saving energy by making it easier to share rides," *Energy Policy Journal*, vol. 21, pp. 134-159, 2008.
- [9] Ciasullo, Maria Vincenza, Orlando Troisi, Francesca Loia and Gennaro Maione, "Carpooling: travelers' perceptions from a big data analysis," *The TQM Journal*, vol. 30, 2018.
- [10] U. S. D. o. Transportation, "Bureau Of Transportation Statistics. Commute Mode by Percentage," in <<https://www.bts.gov/commute-mode>>, 2019.
- [11] Delhomme, Patricia and Alexandra Gheorghiu, "Comparing French carpoolers and non-carpoolers: Which factors contribute the most to carpooling?," *Transportation Research Part D: Transport and Environment*, vol. 42, pp. 1-15, 2016.
- [12] Bresciani, Chiara, Alberto Colorni, Francesca Costa, Alessandro Luè and Luca Studer, "Carpooling: facts and new trends," *In 2018 International Conference of Electrical and Electronic Technologies for Automotive*, pp. 1-4, 2018.
- [13] Cadyjacob, "17 Fascinating World War II Posters About Conservation," in <http://www.ridebuzz.org/wartime_conservation>, 2010.
- [14] Anthopoulos, Leonidas G. and Dimitrios N. Tzimos, "Carpooling Platforms as Smart City Projects: A Bibliometric Analysis and Systematic Literature Review," *Sustainability*, vol. 13, p. 10680, 2021.
- [15] J. O'Brien, "What is Carpooling," in <<https://www.liftango.com/blog/what-is-carpooling>>, 2020.

References

- [16] V. Nambiar, "Designing the Uber Pool vehicle occupancy experience – a UX case study," in < <https://varunxnambiar.medium.com/designing-the-uber-pool-vehicle-occupancy-experience-a-ux-case-study-2debffb5616a>>, 2019.
- [17] D. Etherington, "Lyft is testing a new rider experience with a small percentage of users," in < <https://techcrunch.com/2017/11/08/lyft-is-testing-a-new-rider-experience-with-a-small-percentage-of-users/>>, 2017.
- [18] B. Helling, "The Definitive Guide to Via Rideshare (For Passengers and Drivers)," in < <https://gigworker.com/via/>>, 2018.
- [19] APKPure, "RYDE Carpool, Taxi or Private Hire Ride Hailing," in < <https://apkpure.com/ryde-carpool-taxi-or-private-hire-ride-hailing/com.rydesharing.ryde>>, 2021.
- [20] Arnould, Gérald, Djamel Khadraoui, Marcelo Armendáriz, Juan C. Burguillo and Ana Peleteiro, "A transport based clearing system for dynamic carpooling business services," In *2011 11th International Conference on ITS Telecommunications*, pp. 527-533, 2011.
- [21] Xin, Wenhui, Shunying Zhu, Hong Wang, Yongfei Yan and Jining Xiong, "Notice of Retraction: Analyzing Early Market Potential and Strategies for Carpooling in China: A Case Study of Wuhan," In *2009 International Conference on Management and Service Science*, pp. 1-4, 2009.
- [22] Amey, Andrew, John Attanucci and Rabi Mishalani, "Real-time ridesharing: opportunities and challenges in using mobile phone technology to improve rideshare services," *Transportation Research Record*, vol. 2217, pp. 103-110, 2011.
- [23] L. Créno, "User experience of Dynamic Carpooling: How to encourage drivers and passengers?," In *Energy Consumption and Autonomous Driving*, Vols. Springer, Cham, pp. 71-81, 2016.
- [24] Massaro, Dominic W., Benjamin Chaney, Stephanie Bigler, Jessica Lancaster, Suresh Iyer, Mrunal Gawade, Michael Eccleston, Edith Gurrola and Angelica Lopez, "Just-In-Time Carpooling Without Elaborate Preplanning," In *Proceedings of the fifth international conference on web information systems and technologies*, pp. 219-224, 2009.
- [25] Grgurević, Ivan, Adam Stančić and Marko Slavulj, "Identifying relevant factors of applying technologies in dynamic carpooling," *Tehnički glasnik*, vol. 9, pp. 337-344, 2015.
- [26] Chan, Nelson D. and Susan A. Shaheen., "Ridesharing in North America: Past, present, and future," *Transport reviews*, vol. 32, pp. 93-112, 2012.
- [27] Masoud, Neda and R. Jayakrishnan., "A real-time algorithm to solve the peer-to-peer ride-matching problem in a flexible ridesharing system," *Transportation Research Part B: Methodological*, vol. 106, pp. 218-236, 2017.
- [28] E. Badger, "Slugging the People Transit, Miller-McCune, New York, NY," 2011.
- [29] Mote, Jonathon E. and Yuko Whitestone, "The social context of informal commuting: Slugs, strangers and structuration," *Transportation Research Part A: Policy and Practice*, vol. 45, pp. 258-268, 2011.
- [30] S. Beroldo, "Casual carpooling in the San Francisco Bay Area," *Travelation Quarterly*, vol. 44, pp. 133-150, 1990.

References

- [31] Minett and Paul, "The energy consumption impacts of flexible carpooling," *Flexible Carpooling: Exploratory Study*, pp. 48-63, 2009.
- [32] Shaheen, Susan A., Nelson D. Chan and Teresa Gaynor, "Casual carpooling in the San Francisco Bay Area: Understanding user characteristics, behaviors, and motivations," *Transport Policy*, vol. 51, pp. 165-173, 2016.

Appendix A: Final Year Project Biweekly Report

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

| | |
|--|--------------------------|
| Trimester, Year: JUNE, 2022 | Study week no.: 5 |
| Student Name & ID: Chang Li Hao, 19ACB05664 | |
| Supervisor: Mr Tan Chiang Kang | |
| Project Title: Carpooling App for university students | |

1. WORK DONE

Revising on project 1 report and getting started on project 2 development according to timeline.

Revised Chapter:

Chapter 1:

Project Objective

Project Scope and Direction

Add on Chapter:

Chapter 2:

Summary on Existing Applications

Comparison Between Android and IOS platform

Comparison Between Firebase and AWS Amplify

Comparison Between React Native and Flutter

2. WORK TO BE DONE

Create a work breakdown chart to better planning and organize work to be done.
Started to implement all the basic functionality and interface on the project.

3. PROBLEMS ENCOUNTERED

No

4. SELF EVALUATION OF THE PROGRESS

After revising on project 1 report, I have gained a better understanding on the previous work done and found some weak thesis that should be improve.



Supervisor's signature

Chang Li Hao

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

| | |
|--|--------------------------|
| Trimester, Year: JUNE, 2022 | Study week no.: 7 |
| Student Name & ID: Chang Li Hao, 19ACB05664 | |
| Supervisor: Mr Tan Chiang Kang | |
| Project Title: Carpooling App for university students | |

1. WORK DONE

Done some research and update on literature review to have a better view of technologies used.

Done implement basic functionality and interface on the project.

2. WORK TO BE DONE

Do some revise on Chapter 3 diagram and do some technical report writing.

Continue to implement functions according to the timeline.

Start to implement user authentication to the project.

3. PROBLEMS ENCOUNTERED

No

4. SELF EVALUATION OF THE PROGRESS

Getting to know more about node package manager and expose to more libraries available to be implement in the project.



Supervisor's signature

Chang Li Hao

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

| | |
|--|--------------------------|
| Trimester, Year: JUNE, 2022 | Study week no.: 9 |
| Student Name & ID: Chang Li Hao, 19ACB05664 | |
| Supervisor: Mr Tan Chiang Kang | |
| Project Title: Carpooling App for university students | |

1. WORK DONE

Done revise on use case diagram and update some of the functions to be implement.
Done implement all the user authentication interface and functions on the project.

2. WORK TO BE DONE

Continue to implement functions according to the timeline.
Continue to implement Firestore Database and Storage to the project.
Try to implement more functions from NPM's libraries that will improve user engagement.

3. PROBLEMS ENCOUNTERED

During the development, I have accidentally removed parameter away from one of the use Effect functions and created an infinite loop that keep reading data from Firebase, which has used up all the limited quota on free tier account.

4. SELF EVALUATION OF THE PROGRESS

Learnt a lesson on use Effect function which a parameter is needed to prevent infinite loop.



Supervisor's signature

Chang Li Hao

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

| | |
|--|---------------------------|
| Trimester, Year: JUNE, 2022 | Study week no.: 11 |
| Student Name & ID: Chang Li Hao, 19ACB05664 | |
| Supervisor: Mr Tan Chiang Kang | |
| Project Title: Carpooling App for university students | |

1. WORK DONE

Done implement Firestore Database and Storage to the project.
Done implement image picker and date time picker to the project.

2. WORK TO BE DONE

Started to wrap up the whole development process and proceed to physical device debugging.
Generate APK file and upload to Huawei App Gallery store which is free.
Do some technical report writing on project justification and design specification.

3. PROBLEMS ENCOUNTERED

The development process is slow and cannot keep up with the timeline set. Some of the functions have occupied lots of time to develop.

4. SELF EVALUATION OF THE PROGRESS

Due to the limited time restricted, I will be focusing on implement the necessary functions that will deliver good interface and functionality on the project.



Supervisor's signature

Chang Li Hao


Student's signature

Appendix B: Poster

CAR POOLING APP FOR UNIVERSITY STUDENTS

FINAL YEAR PROJECT

Developer: **Chang Li Hao** Supervisor: **Tan Chiang Kang**



INTRODUCTION

Proposed project will focus on developing a carpooling application specifically for university students. This project is to providing students with an alternative way to commute during their daily routine. We also hope to share the benefits of carpooling through this project with numerous Malaysia university students.

RESULTS

Proposed project will be a carpooling apps that provide features such as user authentication, manage user profile, post ride request, check ride request and so on. There are three types of carpooling services provided, which can be categories into immediate, schedule and regular ride.

DISCUSSION


Carpooling allows transporting more people at one time resulting in fewer trips and cost-effective. It allow people to travel more effectively while creating less traffic and less pollutions. There is limited application that provide carpooling services in Malaysia. This project will provide a great option for students who don't have a vehicle to commute in during their university years in some cities where public transportation remains difficult in terms of availability and accessibility.

METHODS

Agile methodology will be used for this project, an iterative approach has been chosen for this project. iterative approach allows us to develop the product in a cyclical way, building small parts in an evolving way, instead of starting with a complete understanding of the requirements.

DINOPOOL


The name of this application will be DinoPool, with a cute app icon to attract user attention at the same time as providing more simplicity on the user interface




DinoPool

CONCLUSION

The deliverable of this project is to help university students by providing them with an alternative way to commute during their daily routine. We also aim to provide a safe and secure environment for university students while enjoying carpooling services by identifying all user's identification before they can request a service. In addition, we hope to bring the benefits of carpooling to all university students, such as providing an affordable ride based on shared costs, reducing one's carbon footprint, and providing a carpooling platform that allows users to connect with others during the ride, which also contributes to positive mental health.



By Chang Li Hao
Information Systems (Honours)
Business Information Systems



PLAGIARISM CHECK RESULT

FYP2

ORIGINALITY REPORT

| | | | |
|------------------|------------------|--------------|----------------|
| 5 % | 1 % | 4 % | 2 % |
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|----------|---|----------------|
| 1 | Maria Vincenza Ciasullo, Orlando Troisi, Francesca Loia, Gennaro Maione. "Carpooling: travelers' perceptions from a big data analysis", The TQM Journal, 2018 Publication | 3 % |
| 2 | Submitted to Taibah University Student Paper | 1 % |
| 3 | Navulur, Kumar. "List of Tables", Remote Sensing Applications Series, 2006. Publication | 1 % |
| 4 | pt.scribd.com Internet Source | <1 % |
| 5 | Roberto Tabet Aleixandre. "Development of an Installation to Emulate Altitude, Ambient Temperature, and Ambient Humidity on Thermal Engines. Application to the Study of the Impact over E6 Engine Performance", Universitat Politecnica de Valencia, 2022 Publication | <1 % |
| 6 | Submitted to University of Greenwich Student Paper | <1 % |

| | | |
|----|--|------|
| 7 | Submitted to Xiamen University Student Paper | <1 % |
| 8 | Carlos Sun, Praveen Edara. "Is Getting an Uber-Lyft from a Sidecar Different from Hailing a Taxi?", Transportation Research Record: Journal of the Transportation Research Board, 2015 Publication | <1 % |
| 9 | Matteo Mallus, Giuseppe Colistra, Luigi Atzori, Maurizio Murrone, Virginia Pilloni. "Dynamic Carpooling in Urban Areas: Design and Experimentation with a Multi-Objective Route Matching Algorithm", Sustainability, 2017 Publication | <1 % |
| 10 | Sara, . "List Of Tables", Site Assessment and Remediation Handbook Second Edition, 2003. Publication | <1 % |
| 11 | Blatchley, Barbara. "Statistics in Context", Oxford University Press Publication | <1 % |

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off

Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)

Form Number: FM-IAD-005 | Rev No.: 0 | Effective Date: | Page No.: 1 of 1



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

| | |
|-------------------------------------|--|
| Full Name(s) of Candidate(s) | Chang Li Hao |
| ID Number(s) | 19ACB05664 |
| Programme / Course | Business Information Systems |
| Title of Final Year Project | Carpooling App for university students |

| Similarity | Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR) |
|--|---|
| Overall similarity index: <u>5%</u> % Similarity by source Internet Sources: <u>1</u> % Publications: <u>4</u> % Student Papers: <u>2</u> % | |
| Number of individual sources listed of more than 3% similarity: <u>0</u> | |
| Parameters of originality required, and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i> | |

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

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

Signature of Supervisor

Name: Tan Chiang Kang

Date: 9/9/2022

Signature of Co-Supervisor

Name: _____

Date: _____



UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY

(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

| | |
|-----------------|--------------------|
| Student Id | 19ACB05664 |
| Student Name | Chang Li Hao |
| Supervisor Name | Mr Tan Chiang Kang |

| TICK (✓) | DOCUMENT ITEMS |
|-----------------|--|
| | Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item. |
| | Front Plastic Cover (for hardcopy) |
| ✓ | Title Page |
| ✓ | Signed Report Status Declaration Form |
| ✓ | Signed FYP Thesis Submission Form |
| ✓ | Signed form of the Declaration of Originality |
| ✓ | Acknowledgement |
| ✓ | Abstract |
| ✓ | Table of Contents |
| ✓ | List of Figures (if applicable) |
| ✓ | List of Tables (if applicable) |
| ✓ | List of Symbols (if applicable) |
| ✓ | List of Abbreviations (if applicable) |
| ✓ | Chapters / Content |
| ✓ | Bibliography (or References) |
| ✓ | All references in bibliography are cited in the thesis, especially in the chapter of literature review |
| ✓ | Appendices (if applicable) |
| ✓ | Weekly Log |

| | |
|---|---|
| ✓ | Poster |
| ✓ | Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005) |
| ✓ | I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report. |

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

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

Chang Li Hao

(Signature of Student)

Date: 8/9/2022