UTAR VEHICLES TRACKING APP

WONG JIA YING

UNIVERSITI TUNKU ABDUL RAHMAN

UTAR VEHICLES TRACKING APP

WONG JIA YING

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2021

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature :

Name : Wong Jia Ying

ID No. : 1801193

Date : 6-5-2021

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"UTAR VEHICLES TRACKING APP"** was prepared by **WONG JIA YING** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ____________________

Supervisor : ____________________

Date : ____________________

Signature : ____________________

Co-Supervisor : ____________________

Date : ____________________

ACKNOWLEDGEMENTS

I would like to express my gratitude to everyone who had contributed to the successful completion of this project. I would like to thank my supervisor, Dr. Mohammad Babrdel Bonab for his invaluable advice, guidance and his enormous patience in guiding me throughout the project.

In addition, I would also like to express my appreciation to my family for their support and encouragement in completing the project.

Lastly, I would like to thank my friends who had given me support, feedbacks and advice throughout the project.

ABSTRACT

UTAR provides transport services for students and staffs to allow them to go to the university for classes or work and other places more conveniently. However, there are several problems in existing transport services such as lack of real time information to enable students and staffs to have better knowledge on current transport services and ease of getting the latest routes and schedules information. Hence, this project aimed to provide a platform for students and staffs to track UTAR vehicles on real time and obtain routes and schedules information more conveniently. Other than that, this project sought to provide a platform for UTAR transport section to manage transport services more efficiently and effectively. The project objective is to develop a vehicle tracking application for UTAR students and staffs and UTAR transport section. The development methodology adopted in this project was evolutionary prototyping methodology. The final outcome of this project is 3 applications, which are the driver's mobile application for sending location updates, the students and staffs' mobile application and the transport section administrative staff's web application. At the end of this project, the project objectives planned were achieved and all 3 applications were successfully developed.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

UTAR	Universiti Tunku Abdul Rahman
API	Application Programming Interface
SDK	Software Development Kit
OEM	Original Equipment Manufacturer

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

UTAR is in sub-urban area. In order to go to university for classes or work as well as to other places such as rail interchange station, a number of UTAR students and staffs travel by buses. Moreover, most students came from other states or country and do not have own transport. Buses become the mode of transport for them to travel to university and other destinations. Hence, transport service in UTAR is important to improve the mobility of UTAR students and staffs. Transport service in UTAR is offered with UTAR vehicles such as buses and vans. Punctuality and reliability of UTAR vehicles are crucial so that students and staffs that rely on the transport service can trust that they will not be late for classes or work as well as other activities.

The outcome of this project is a mobile application that can enable UTAR students and staffs to track UTAR vehicles and obtain routes information conveniently at the tip of their hands. UTAR transport section can also track UTAR vehicles and manage the driver and schedule information through the web application version.

1.2 Project Background

Buses are among the most used public transport in Malaysia. People take bus to workplaces, schools, universities, shopping centres and railway interchange stations. However, public transport such as buses are becoming less attractive due to several reasons. Punctuality and reliability of bus service are among the reasons people are unsatisfied with the service. Besides, the accessibility of public buses is limited and may not be in the preferred area for university students and staffs. Therefore, many universities provide transportation service to ease the transportation problems of university students and staffs.

Students and staffs can obtain information such as schedules and routes available from the university such as get it on bulletin boards and university websites. However, there is lack of real time information such as arrival time based on current traffic conditions and current location of the vehicle provided by university for students and staffs. The uncertainties of the current location of the vehicle make them to feel insecure while waiting (Sharif, et al., 2018). Students and staffs have to wait at the bus stop earlier and if there is delay, they have to wait even longer, not knowing when it will arrive. Therefore, long waiting time and lack of knowledge about exact arrival time are among the reasons of unsatisfaction with university transport service (Chit, et al., 2017).

Vehicle tracking system is increasingly popular nowadays to enable users to be more informed and to monitor transport services. Accurate location of the vehicle according to respective routes can be displayed and retrieved by the users of the system. Users can track the real time location of the vehicle as well as view on routes available and time schedules of the routes. This makes their traveling more convenient. According to Watkins, et al. (2011), real time information gives users a stronger sense of control. The access to real time information makes them feel more at ease. As students and staffs rely on university transport to travel, the importance of real time information regarding arrival and schedules is significant to allow them to plan their time better and avoid being late.

1.3 Problem Statement

There a few issues regarding to UTAR transport service making it not so efficient. This includes lack of real time vehicle arrival information and ease of getting route schedules.

1.3.1 Lack of real time vehicle arrival information

1.3.1.1 Increase waiting time

According to studies made by Watkins, et al., (2011) and Rahman, Wirasinghe and Kattan (2013), if passengers have access to real time information regarding bus arrival, their waiting time is shorter. This shows that the time they go to the bus stop is more near to the actual time the bus arrives. As a result, they can have more time to prepare. The time spent waiting at the bus stop is reduced.

At present, students and staffs can only know the scheduled time of UTAR vehicles for every route from the schedules provided on the UTAR website. They are unable to get the exact or the accurate arrival time of the vehicle that is according to current traffic conditions. Hence, they are advised to wait at the bus stop earlier than the scheduled arrival time to avoid missing the vehicle. As a result, they have lesser time for preparation or have to wake up earlier to get ready so that they can be reach the bus stop in advance.

1.3.1.2 Undesirable bus stop condition

In addition, bus stops usually only equipped with simple facilities such as a bench and a shuttle for cover. Some even have only a signboard indicating it is a bus stop without a proper shuttle. Tropical country like Malaysia has high temperatures and rainfall. Weather conditions affect students and staffs waiting at the bus stops. For example, bad weather conditions such as heavy rain can cause them waiting at the bus stops to be drenched whereas on hot and sunny days, they will feel hot and uncomfortable waiting at the bus stop for long time. Besides, there is also safety risk waiting in the bus stop. They might become a target of robbery if waiting at the bus stop alone especially early in the morning where there are less people around.

1.3.1.3 Frustration

Waiting cause people to get frustrated. Especially when they are waiting to go to the university for classes or work as well as having important date that they should be on time, they can get anxious during the waiting time. People tend to feel insecure and worried when there are uncertainties. There might also be situations where the vehicle arrives late due to traffic congestion or other issues. Travel satisfaction will reduce due to delays and excessive waiting time (Lunke, 2020). Despite of the pre-determined schedule provided on UTAR's website, they do not know exactly how long more they must wait for the vehicle to arrive. There is no way for students and staffs to know where the vehicle currently is. As a result, they get more frustrated. This also result in consequences such as students and staffs unable to attend to classes or work on time.

1.3.2 Ease of getting latest route schedules

In addition, it is not convenient enough for students and staffs to get the latest route schedules. The transport management section will come up with new schedules at the beginning of every semester. Students and staffs have to go to the official UTAR website to download a copy of the new version of route and schedule information which is less convenient. There might also be differences in schedules on orientation week and teaching week which makes it more troublesome as they have to download the schedules several times. The announcements are made on the UTAR bus news Facebook page, portal and email. However, they may not check them so frequently hence may be not informed of the latest schedule. This affect their planning if they do not know about the changes. They have to find for other alternatives last-minute to get to destination on time.

1.4 Project Objectives

As stated in the problem statement, there are few issues regarding the current transport services in UTAR. In order to make the transport service more efficient and allows students and staffs to plan their travel better, this project sought to implement a vehicle tracking application for UTAR students and staffs to improve their travel experience and satisfaction with UTAR transport. The objectives to be achieved are as follow:

- To analyse the requirements through reviewing existing applications.
- To develop a vehicle tracking application for UTAR students and staffs and UTAR transport section.
- To evaluate the application functionalities through unit testing, integration testing and user acceptance testing.

1.5 Proposed Solution

In order to solve the problems with UTAR transportation, UTAR Vehicles Tracking App was proposed. It is a real time vehicle tracking application that is based on mobile and web platform. The target users of the mobile application are UTAR drivers, students and staffs while the web application is for UTAR transport section administrative staffs. The application was based on client server architecture and used socket programming concept for real time location tracking.

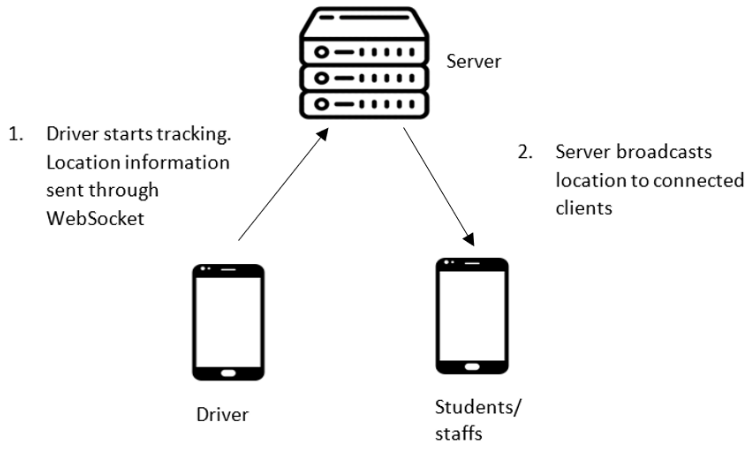

Server
1. Driver starts tracking. Location information sent through WebSocket
2. Server broadcasts location to connected clients
Driver
Students/ staffs

Figure 1.1: Overview of Solution

The workflow of tracking the vehicle location starts with UTAR drivers log in to the application, select the schedule they will be travelling then start tracking. The drivers can send message updates on the current status of the trip. For example, on schedule, traffic jam or vehicle breakdown message. The location information and messages will be sent to the socket server and the socket server will broadcast the information to all connected clients. On the other hand, UTAR students and staffs log in to the application to get the real time vehicle location, driver and vehicle information of the selected route. The estimated remaining time for the vehicle to arrive to the respective bus stop on the route will be calculated and displayed. The location of the vehicle will be shown on a map through the use of Google Maps API. Transport section administrative staffs can use the web application version to monitor whether the transport services are in a timely manner.

1.6 Project Approach

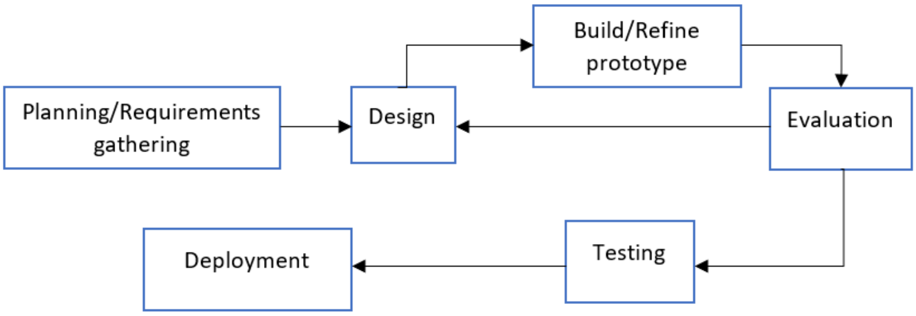


Figure 1.2: Evolutionary Prototype Methodology

Evolutionary prototyping methodology was chosen as the development methodology. First, the requirements were gathered from the project supervisor and also extracted from existing applications. At design phase, the design of the system architecture was produced. After designing, a working prototype with limited functionality was built first. The prototype then undergone evaluation to gather feedback. This allows better understanding of requirements so that the system produced fulfils users' needs. Users also get better understanding on the system as they are involved in early stage. After that, the prototype was refined and evaluated again. The process of design, refining and evaluation of prototype will be repeated until all requirements are being met and users are satisfied with the prototype. The prototype was then developed as the final product. This helps to save time and effort as the prototype can be used for final product development instead of having to develop it from scratch again. In testing phase, the final product was being tested so to ensure the application works as intended and defects and bugs are fixed. After all tests were passed, the final product was completed. The final product of this project is 3 version of UTAR vehicle tracking application which are the driver's mobile application version, students and staffs' mobile application version and UTAR transport section web application version.

1.7 Scope of the Project

This section discusses about the platform, target users, location, assumptions, modules covered and modules not covered for the project.

1.7.1 Platform

The development of UTAR Vehicles Tracking App was on two platforms, mobile and web platform. Mobile platform is based on Android. The mobile application will be compatible on Android mobile devices with Android version 5 and above only.

1.7.2 Target Users

1.7.2.1 Mobile platform

- **UTAR students and staffs**

  To track real time vehicle location and information and get latest route schedule.
- **UTAR vehicle drivers**

  To provide the vehicle location for tracking.

1.7.2.2 Web platform

- **UTAR transport section administrative staffs**

  To view travelling record, manage route schedules and driver information .

1.7.3 Location

The application is only for the use of UTAR Sungai Long campus. Only UTAR Sungai Long campus vehicle routes and schedules were included.

1.7.4 Assumptions

This project assumed that the user of the mobile application has mobile internet connectivity such as mobile data and Global Positioning System (GPS) on their devices.

1.7.5 Modules Covered

The modules below will be covered to achieve the project objectives.

1.7.5.1 Mobile platform

- **Track real time vehicle location**

  Drivers can select scheduled trip and start tracking through this module. Message regarding vehicle status can be updated by the driver to students and staffs. When the trip ends, drivers can stop the tracking. Students and staffs can view the real time location of UTAR vehicles of respective routes on a map displayed.

- **Display real time vehicle information**

  This module allows students and staffs to get information such as driver's name, vehicle plate number, estimated time of arrival and message update from driver.

- **Display route information**

  This module allows students and staffs to view a list of routes available. The stops and route map of respective route will be displayed when the route was being selected.

- **Display schedule**

  This module allows students and staffs to check on latest schedules according to route and date selected.

1.7.5.2 Web platform

- **View travel record**

  Transport section administrative staffs can view the information of travel record such as route name, vehicle plate number, driver name, scheduled start time, actual start time and status.

- **Manage schedule**

  Transport section administrative staffs can add and modify schedule information of a route such as arrival time at each stop on the route, schedule period, assign drivers and vehicles to each trip of the route accordingly.

- **Manage drivers**

  Transport section administrative staffs can add new drivers and modify existing driver information.

1.7.6 Modules Not Covered

Certain modules will not be covered in this project. This included route tracking module for drivers' application and module for administrative staffs to manage routes information.

Route tracking module is to track the route taken by the driver for UTAR transport section to monitor whether the driver skips certain stops on the route. Manage routes module is for UTAR transport section administrative staffs to add new routes and modify existing routes information. Hence, the display schedule and manage schedule module in the application will be based on existing route schedules only. These modules will not be included in the scope for this project due to limited time in implementation and development of the application.

1.8 Conclusion

This chapter discussed about the problems, objectives, solution, development approach and features included in the applications developed of this project. In summary, UTAR Vehicles Tracking App will be developed on mobile and web platform to solve the current transport service problems stated. The mobile platform consists of the drivers' application and students and staffs' application while the web platform consists of the administrative staffs' application.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter will be reviewing existing similar applications in section 2.2, software development methodologies in section 2.3 and mobile applications development framework in section 2.4.

2.2 Review on existing applications

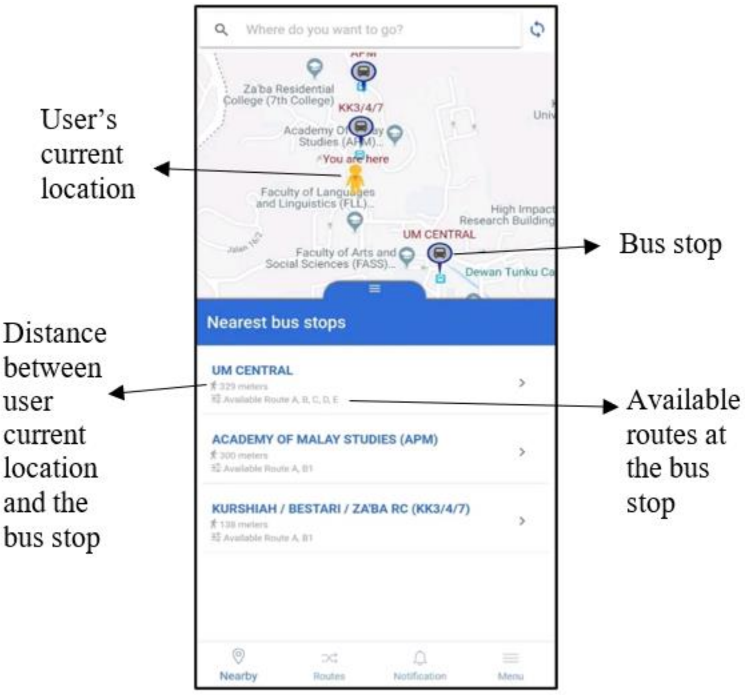
There are numerous universities in local and overseas that have successfully implemented vehicle tracking application. Some of the applications of those universities are included in this review. These universities are Universiti Malaya, National University of Singapore and Nanyang Technological University. A review was done for these similar applications to find out the features and to ensure better understanding on user requirements for development of the proposed application. Besides, this review also analysed the strengths and weaknesses in these applications by doing a comparison between them.

2.2.1 Overview of features available

2.2.1.1 BusMalaya

BusMalaya is a mobile application to track campus bus of Universiti Malaya in Malaysia. It is available on both Android and iOS platform. The features available for this application are display bus stop details, plan journey, display route information, track real time location of the bus, notifications and feedback.

- **Nearest bus stops and bus stop details**


User's current location
Where do you want to go?
APM
Za'ba Residential College (7th College)
KK3/4/7
Academy Of Malay Studies (APM)...
You are here
Faculty of Languages and Linguistics (FLL)...
Univ
High Impact Research Building
UM CENTRAL
Faculty of Arts and Social Sciences (FASS)...
Jalan 16/2
Dewan Tunku Ca
Bus stop
Distance between user current location and the bus stop
Available routes at the bus stop
Nearby
Routes
Notification
Menu

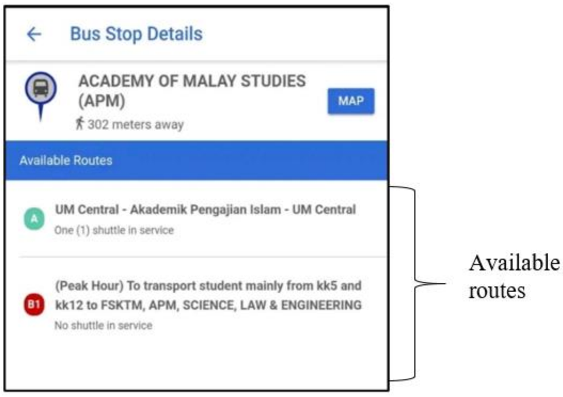
Nearest bus stops

UM CENTRAL
329 meters
Available Route A, B, C, D, E

ACADEMY OF MALAY STUDIES (APM)
300 meters
Available Route A, B1

KURSHIAH / BESTARI / ZA'BA RC (KK3/4/7)
138 meters
Available Route A, B1

Figure 2.1: BusMalaya Nearest Bus Stop Screen



Bus Stop Details

ACADEMY OF MALAY STUDIES (APM)
302 meters away

Available Routes

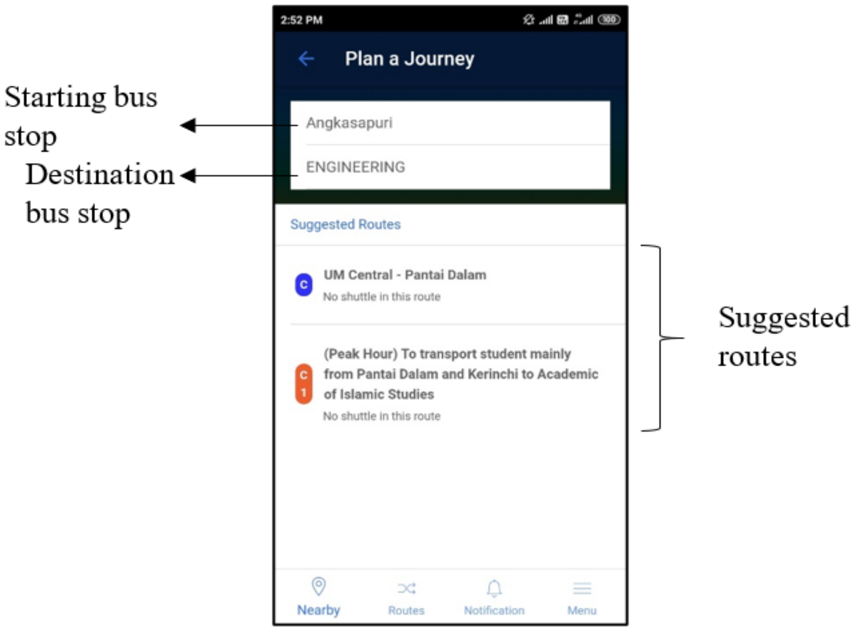
UM Central - Akademik Pengajian Islam - UM Central
One (1) shuttle in service

(Peak Hour) To transport student mainly from kk5 and kk12 to FSKTM, APM, SCIENCE, LAW & ENGINEERING
No shuttle in service

Figure 2.2: BusMalaya Bus Stop Details Screen

After opening the BusMalaya mobile application, the map will show the user's current location. User also get to see the list of nearest bus stops to current location. The bus stop's walking distance and routes available is also shown.

- **Plan journey**



Angkasapuri

ENGINEERING

Suggested Routes

C UM Central - Pantai Dalam
No shuttle in this route

C1 (Peak Hour) To transport student mainly from Pantai Dalam and Kerinchi to Academic of Islamic Studies
No shuttle in this route

Starting bus stop

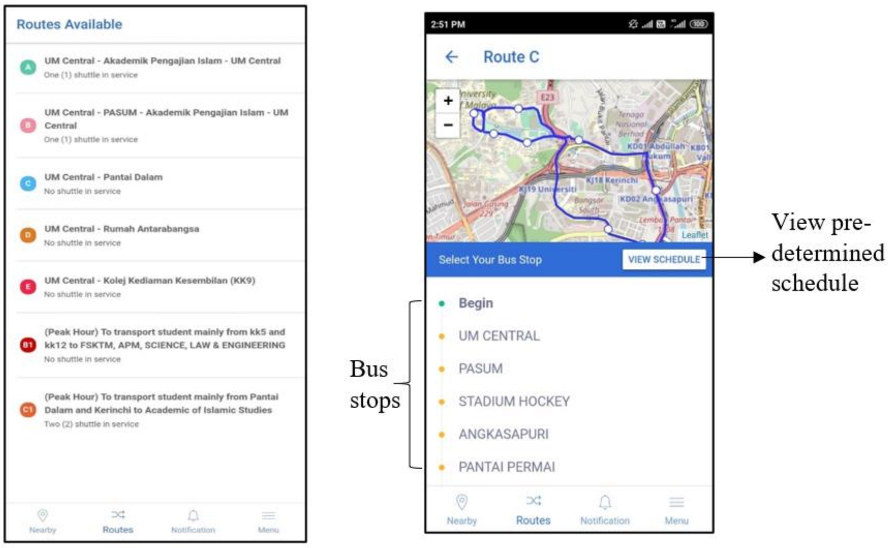
Destination bus stop

Suggested routes

Figure 2.3: BusMalaya Plan Journey Screen

For plan journey feature, user can search which route they can take to go to their destination. User can select their starting bus stop and destination bus stop. A list of suggested routes to the destination will be provided.

- **Routes information**



Routes Available

A UM Central - Akademik Pengajian Islam - UM Central
One (1) shuttle in service

B UM Central - PASUM - Akademik Pengajian Islam - UM Central
One (1) shuttle in service

C UM Central - Pantai Dalam
No shuttle in service

D UM Central - Rumah Antarabangsa
No shuttle in service

E UM Central - Kolej Kediaman Kesembilan (KK9)
No shuttle in service

B1 (Peak Hour) To transport student mainly from kk5 and kk12 to FSKTM, APM, SCIENCE, LAW & ENGINEERING
No shuttle in service

C1 (Peak Hour) To transport student mainly from Pantai Dalam and Kerinchi to Academic of Islamic Studies
Two (2) shuttle in service

Select Your Bus Stop

- Begin
- UM CENTRAL
- PASUM
- STADIUM HOCKEY
- ANGKASAPURI
- PANTAI PERMAI

Bus stops

View pre-determined schedule

Figure 2.4: BusMalaya Routes Information Screens

A list of routes information is also available for the user. User can select the route to view the route map and bus stops along the route. User can get the bus information by selecting the bus stop. User can also view the pre-determined schedule from here. An image of the scheduled timetable of the route will be shown. However, the image shown is blur and cannot be seen clearly especially the notes at the right side of the image.

- **Track bus location and remaining time to arrival**

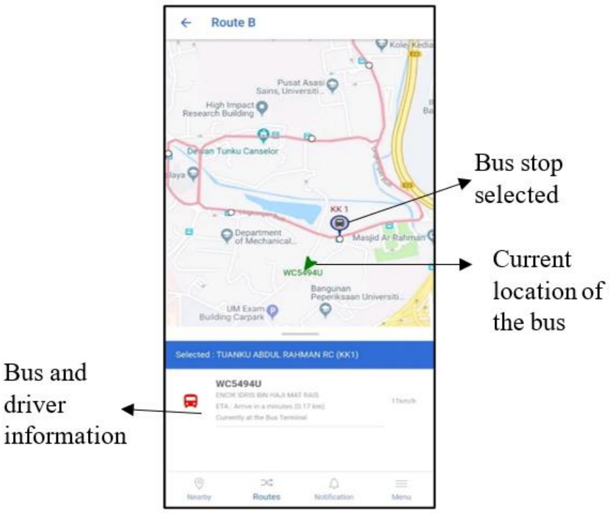

Route B
Pusat Asasi Sains, Universiti...
High Impact Research Building
Dewan Tunku Canselor
KK 1
Masjid Ar Rahman
Department of Mechanical...
WC5494U
Bangunan Peperiksaan Universiti...
UM Exam Building Carpark
Bus stop selected
Current location of the bus
Bus and driver information
Selected : TUANKU ABDUL RAHMAN RC (KK1)
WC5494U
ENCIK IDRIS BIN HAJI MAT RAIS
ETA : Arrive in a minutes (0.17 km)
Currently at the Bus Terminal
11km/h
Nearby
Routes
Notification
Menu

Figure 2.5: BusMalaya Track Bus Location Screen

When a bus stop is selected, the bus information such as bus plate number, driver’s name, estimated time to arrival, distance, speed and current location will be shown on the screen. The map showing the location of the bus and the route is also provided.

- **Notifications and feedback**

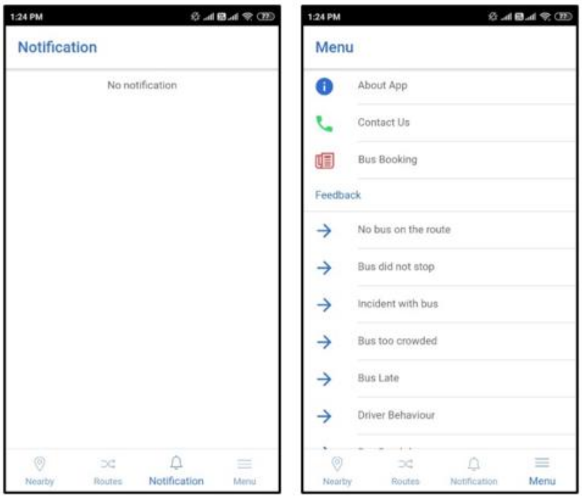


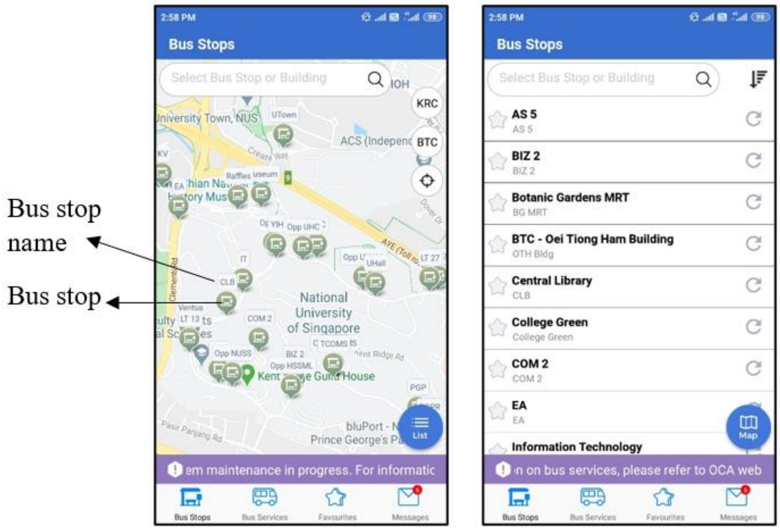
Figure 2.6: BusMalaya Notification and Feedback Screen

The application also allows user to see the latest notifications about the bus service. Users can also view contact information of the management, book buses and provide feedback regarding the bus service.

2.2.1.2 NUS NextBus

NUS NextBus is a mobile application to track campus bus of National University of Singapore. It is available on both Android and iOS platform. The features available are display bus stop details, routes information, real time location tracking, notifications and feedback.

- **Bus stop detail**



em maintenance in progress. For informatic

AS 5
AS 5

BIZ 2
BIZ 2

Botanic Gardens MRT
BG MRT

BTC - Oei Tiong Ham Building
OTH Bldg

Central Library
CLB

College Green
College Green

COM 2
COM 2

EA
EA

Information Technology

n on bus services, please refer to OCA web

Figure 2.7: NUSNextBus Bus Stop Detail Screen

User can view a map with the location of bus stops marked. The name of bus stops is also displayed. This makes it easy to identify the bus stop. Users can also choose to view the bus stops as a list instead of map. Bus stop detail such as routes available will be displayed if user select the bus stop.

- **Routes information**

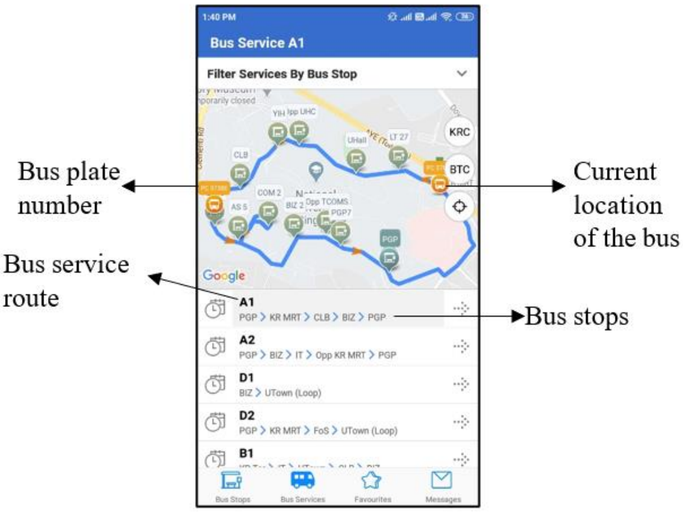

Bus Service A1
Filter Services By Bus Stop
Bus plate number
Bus service route
Current location of the bus
Bus stops
KRC
BTC
A1
PGP > KR MRT > CLB > BIZ > PGP
A2
PGP > BIZ > IT > Opp KR MRT > PGP
D1
BIZ > UTown (Loop)
D2
PGP > KR MRT > FoS > UTown (Loop)
B1
Bus Stops
Bus Services
Favourites
Messages

Figure 2.8: NUSNextBus Routes Information Screen

The application also has feature that show bus service routes instead of bus stops. The bus stops available along the route is also provided. On the map displayed, the selected bus service route will be highlighted. The current location of the bus is displayed with an orange bus icon and plate number is also shown. Through this, user can identify the bus correctly and avoid boarding the wrong bus.

- **Track bus location and remaining time to arrival**

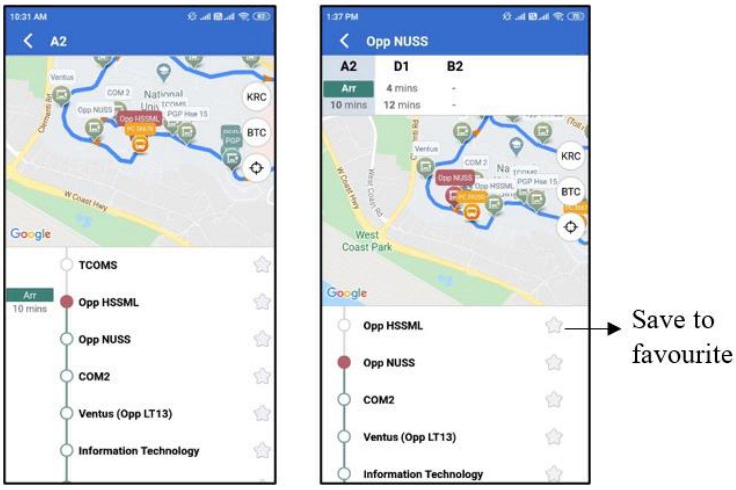
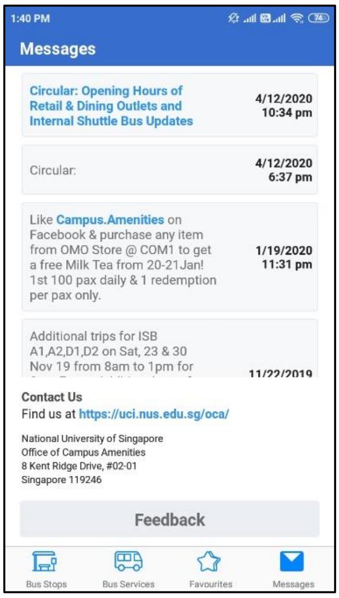


Figure 2.9: NUSNextBus Track Bus Location Screen

The application displays the next two estimated time to arrival to the user according to bus service route. The immediate arrival will be highlighted with green background. There are two presentations to display the estimated time to arrival. Besides, the application also has a feature to save bus stop as favourite.

- **Notification and feedback**



1:40 PM

Messages

Circular: Opening Hours of Retail & Dining Outlets and Internal Shuttle Bus Updates 4/12/2020 10:34 pm

Circular: 4/12/2020 6:37 pm

Like Campus.Amenities on Facebook & purchase any item from OMO Store @ COM1 to get a free Milk Tea from 20-21Jan! 1st 100 pax daily & 1 redemption per pax only. 1/19/2020 11:31 pm

Additional trips for ISB A1,A2,D1,D2 on Sat, 23 & 30 Nov 19 from 8am to 1pm for 11/22/2019

Contact Us
Find us at https://uci.nus.edu.sg/oca/

National University of Singapore
Office of Campus Amenities
8 Kent Ridge Drive, #02-01
Singapore 119246

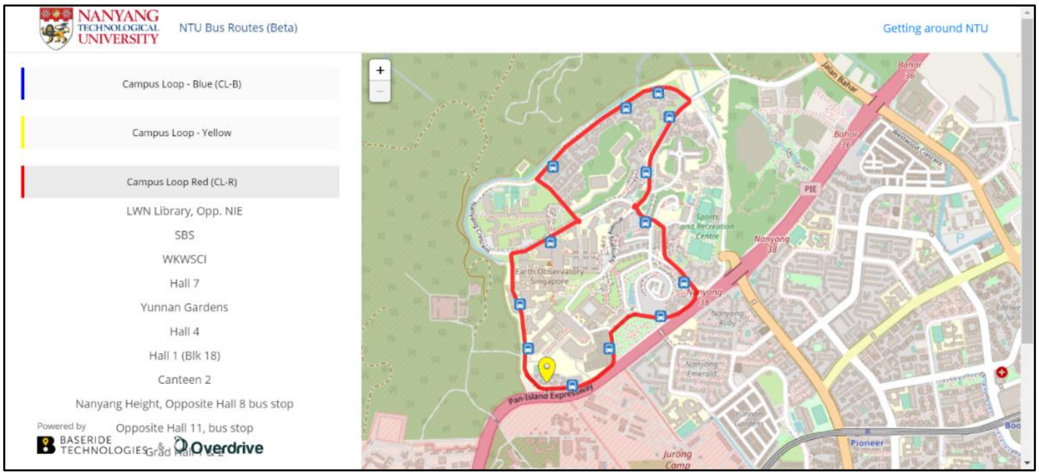
Figure 2.10: NUSNextBus Notification and Feedback Screen

Moreover, notification messages about updates on shuttle bus service is provided in the application. User can also give feedback on the services through the application.

2.2.1.3 NTU Bus Routes

NTU Bus Routes is a web application that is used to track campus bus of Nanyang Technological University Singapore. There is no mobile application for NTU Bus Routes currently. Features available include display route information and bus location tracking.

- **Route information**



NANYANG TECHNOLOGICAL UNIVERSITY
NTU Bus Routes (Beta)
Getting around NTU

Campus Loop - Blue (CL-B)

Campus Loop - Yellow

Campus Loop Red (CL-R)

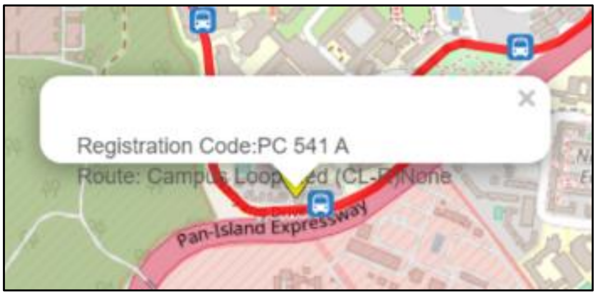
LWN Library, Opp. NIE
SBS
WKWSCI
Hall 7
Yunnan Gardens
Hall 4
Hall 1 (Blk 18)
Canteen 2
Nanyang Height, Opposite Hall 8 bus stop
Opposite Hall 11, bus stop
Grad Hall 1 & 2

Powered by BASERIDE TECHNOLOGIES & Overdrive
Jalan Bahar
PIE
Sports and Recreation Centre
Earth Observatory Singapore
Nanyang
Pan-Island Expressway
Pioneer
Jurong Camp

Figure 2.11: NTU Bus Routes Route Information Screen

A list of routes and a map is displayed to the user. When user clicks a route, a list of bus stops along the routes is displayed. The selected route will be highlighted in the map. The bus stops are displayed as a blue bus icon while the location of the bus is displayed as a yellow pin icon.

- **Track bus location and remaining time to arrival**

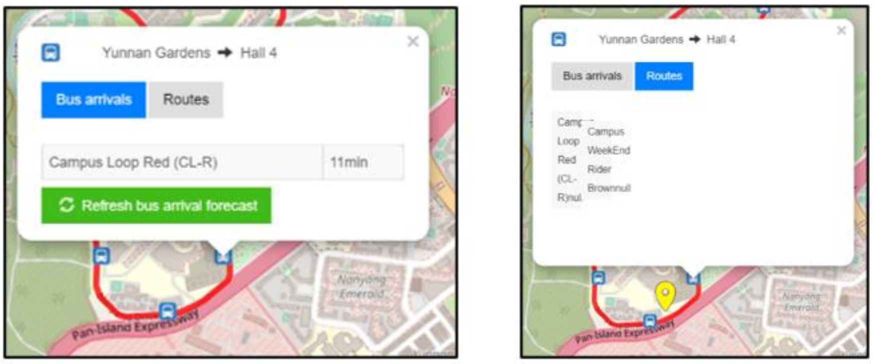


Registration Code:PC 541 A

Route: Campus Loop Red (CL-R)None

Figure 2.12: NTU Bus Routes Track Bus Location Screen

When clicking the yellow pin icon, bus information such as registration code and the current route operating is displayed to the user. This can help user to identify the correct bus and avoid boarding the wrong bus. However, the drawback is the bus information displayed is dislocated. It is out of the white box hence making it difficult to see the information.



Yunnan Gardens → Hall 4

Campus Loop Red (CL-R)	11min

Yunnan Gardens → Hall 4

Campus Loop Red (CL-R)null

Campus WeekEnd Rider Brownnull

Figure 2.13: NTU Bus Routes Bus Arrival Time and Routes Available

Besides, user can select the bus stop icon on the map to show the bus arrival information and routes available. The disadvantage is the estimated time to arrival is not updated automatically. User need to click the refresh button to see the latest estimated time. However, the routes information shown is also difficult to read.

2.2.2 Strengths and weaknesses of existing applications

2.2.2.1 BusMalaya

- **Strengths**
    - User can plan their journey in advance through the plan journey feature. User can search the routes to take to travel to destination automatically through the application instead of manually review through all available routes to find for the suitable route.
    - Driver's name is provided. User can get to know the driver for the bus. The travelling speed of the bus is also calculated and displayed. This can also improve the accountability of the driver as user can provide feedback specifically for the driver. For example, in situations where the driver is speeding or if user finds out driver's misbehaviour user can report the driver to the management.
- **Weakness**
    - The application uses images of pre-determined route schedule. If there are changes to the schedule, it is more difficult to update and is not reflected. Besides, the images provided are in low resolution. User might not be able to see it clearly.

2.2.2.2 NUS NextBus

- **Strengths**
    - User can save the frequently use bus stop to favourites. This can ease the process of finding them each time user wants to check on their information.
    - The nearest two estimated time to arrival of bus is provided to the user. User not only can check on the next bus arrival but also the bus arrival after that. When the bus is approaching to the bus stop, the remaining time will change to Arr (Arriving) highlighted in green. This can draw user's attention.

- **Weakness**
    - There is feature for user to check the route schedule. User cannot check on the timetable of the routes available to plan their journey in advance. There is only estimated time to next arrival available for user.

2.2.2.3 NTU Bus Routes

- **Strength**
    - The routes displayed on the map is differentiated with different colours. This allows user to see each route clearly on the map.
- **Weaknesses**
    - The application does not show any information to the user when the bus service is not available on the route. For example, when user clicks on the route, the route with bus stops is only highlighted on the map without a message to tell user that the bus service is currently unavailable. This will make user to be confused wondering where the bus is.
    - The bus and route information on the map is not clear. The information is either dislocated out of the pop-up box or cramped together making it difficult for user to see.
    - User has to refresh from time to time to get the latest estimated time of arrival and bus location.
    - There is no mobile application version so making it less convenient to access using mobile phone. User has to use phone browser to view the web page.

2.2.3 Comparison between existing and proposed application

Table 2.1: Comparison of Features

Applications **Features**	**BusMalaya**	**NUS NextBus**	**NTU Bus Routes**	**Proposed application**
Platform	Mobile (Android & iOS)	Mobile (Android & iOS)	Web	Mobile (Android)
Routes information	✓	✓	✓	✓
Time Schedule (e.g. pre-determined route timetable)	✓	✕	✕	✓
Plan journey	✓	✕	✕	✕
Vehicle location tracking	✓	✓	✓	✓
Estimated time to arrival	✓	✓	✓	✓
Vehicle information (e.g. plate number)	✓	✓	✓	✓
Driver information (e.g. driver's name)	✓	✕	✕	✓
Notification	✓	✓	✕	✕
User authentication	✕	✕	✕	✓
Driver's message on current vehicle status (e.g. on schedule, bus delay, traffic jam, accident, vehicle breakdown)	✕	✕	✕	✓

2.2.4 Summary

In summary of the review, the features in common of all the applications are display route information, vehicle location tracking, estimated time of arrival, vehicle and driver information. BusMalaya is considered the most all rounded as it has almost all the features available in NUS NextBus and NUS Bus Routes. It also has an additional feature to help user to plan their journey. However, NUS NextBus outperforms the other two applications for vehicle location tracking feature. The bus stop name and bus plate number are displayed on the map making it clearer for user to identify compared to BusMalaya especially when there are multiple buses operating on the route at the same period. The next two estimated time of arrival is also provided instead of only the next arrival. The strengths of these applications will be considered for the development of the proposed application. The proposed application will also include these features collected from the existing application except for plan journey feature in BusMalaya and notification feature due to time constraint.

There are two additional features that is not available in these existing applications that will be included in the proposed application. The features are user authentication and driver’s message updates. The proposed application will implement user authentication to ensure only UTAR drivers, students and staffs can access the application. Besides, driver can provide updates on current vehicle status such as traffic jam, vehicle breakdown, on schedule and so on to allow students and staffs to be more informed of current situation while waiting for the vehicle.

2.3 Review on Software Development Methodologies

Software development lifecycle (SDLC) is the process to build a piece of software. It is also known as software development process. It includes planning and defining the project and requirements, software system design, building and implementation, testing the software and lastly deployment of the software (Tutorialspoint, n.d.). Software development methodology is the framework to implement the SDLC. There are many types of methodologies to date. The methodologies that will be covered in this review are waterfall, prototyping and agile methodology.

2.3.1 Waterfall methodology

Waterfall methodology is a sequential approach in software development and the oldest methodology in software development (Malik and Nigam, 2017). Sommerville (2011) concludes that waterfall model methodology has 5 phases which is requirements gathering, system design, implementation, testing and lastly operation and maintenance as shown in Figure 2.14.

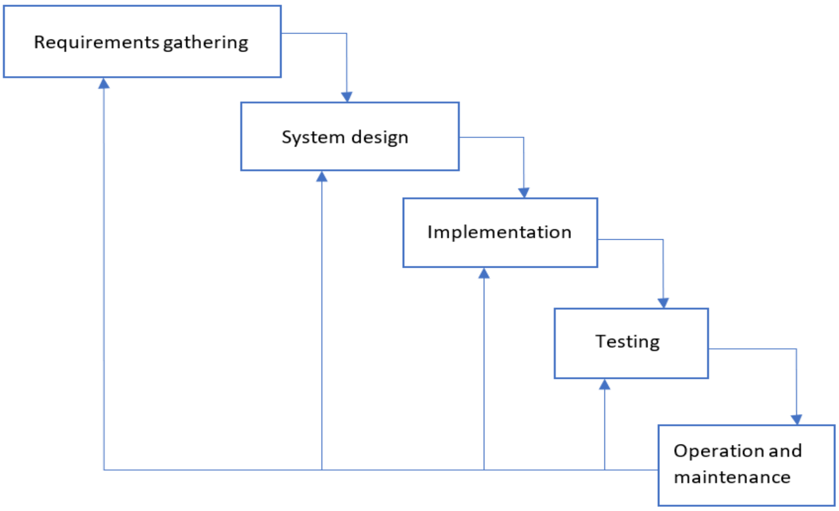


Figure 2.14: Waterfall Methodology

In waterfall methodology, each phase should be completed before proceeds to the next phase (Software Testing Help, 2020). There is lesser interaction with customer in waterfall methodology compared with other methodologies as the product will only be showed to customer at the end of development. There are high uncertainties for complex and long projects as it is hard to go back to previous phase if there are any changes. Therefore, requirements should be clearly defined and unambiguous before the start of development for projects using waterfall methodology. However, it is hard to have all requirements clearly specified from the start. As waterfall methodology is sequential, it can be easily understood and simple to use. However, as it is relatively rigid, it is more difficult and costly to go back to previous stages in case of changes. In addition, testing phase only occurs near the end of development. According to Ghahrai (2016), as waterfall methodology involves high uncertainties and demands that all requirements should be clearly defined in advance, it is more suitable for small projects and projects that have stable requirements.

2.3.2 Prototyping

According to Pressman and Maxim (2015), prototype methodology is considered as an evolutionary model. Hence, prototyping is an iterative process in which the software evolves in each iteration closer to the final complete software. The development team produce a prototype, gather feedback from customer and continuously refine the prototype. Pressman and Maxim (2015) concluded that there are few iterative phases in prototype methodology. This includes designing the prototype or refining the design, building the prototype and refining the prototype after customer evaluation. The process of communicating with customer and refining the prototype iterates until customer requirements are clear and customer is happy with the final prototype before getting into implementation phase. After that, the development of the product will be based on the final prototype approved by the customer. There are different types of prototype methodologies such as throwaway prototype and evolutionary prototype.

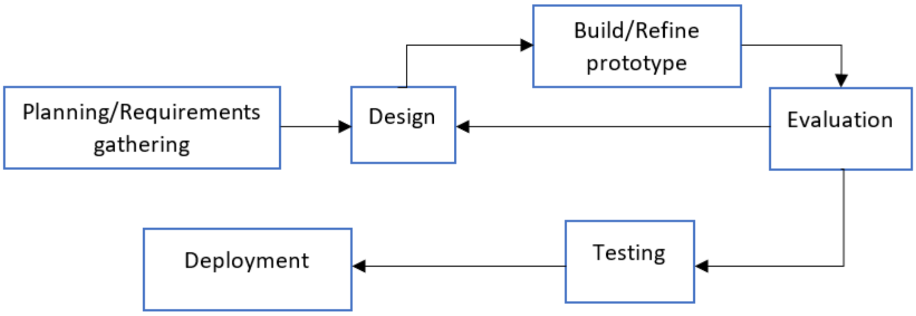


Figure 2.15: Prototype Methodology

In prototyping methodology, there is more interaction with the customer. This helps to clarify the ambiguity in requirements and functionality. Better understanding can be achieved. Developers can have better understanding of customer needs while customer can have better understanding on the system to be produced (Guru99, n.d.). Besides, feedback given from customer is quicker in early stages as every time the prototype is produced it is being evaluated by the customer. This can reduce the risk of developing a product which does not meet customer requirements (Pal, 2018). However, prototype methodology may result in more complexity in the project as customer might add new requirements not specified previously (Saeed et al., 2019). The project scope may be expanded and extended. Prototyping methodology is suitable when there is ambiguity in requirements and when the system involves a lot of user interactions.

2.3.3 Agile methodology

Agile methodology emphasizes on continuous delivery and improvements of the software in iterations (Guru99, n.d.). It focuses on rapid delivery of software without sacrificing quality and at the same time satisfying customer requirements.

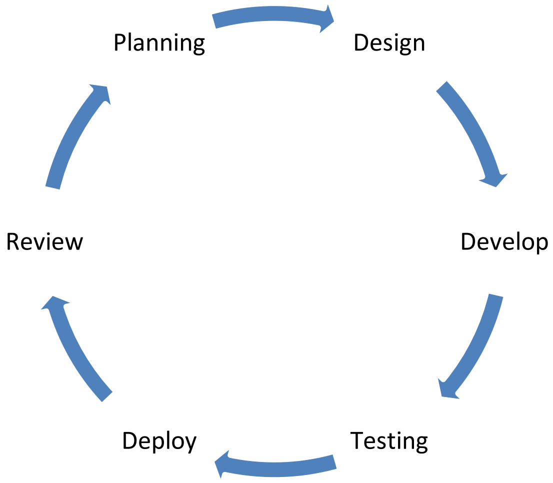


Figure 2.16: Agile Methodology

The phases in agile methodology are planning, designing, developing, testing, deployment and review. The cycle is iterative. Each iteration usually lasts for one to four weeks (Meschankina, 2019). Each iteration is equivalent to a small software project which produces an incremental release at the end.

Every phase in the agile development cycle repeats in each iteration. Additional functionalities or improvements to be made after customer review can be added in next release. The software product evolves and becomes a better version through each iteration. The iteration ends when the software is fully developed and all requirements or functionalities are fulfilled. There are many types of agile methodologies. For example, SCRUM, Feature Driven Development, eXtreme Programming and Kanban (Anand and Dinakaran, 2016). All types of the methodologies have differences in processes but have similar agile core values such as adaptability to changes in requirements and emphasize frequent interaction and communication between development team and also with customer. The advantages of adopting agile methodology includes early testing. As each iteration gone through testing phase, bugs and defects of the software can be detected and fixed earlier. Besides, as frequent feedback from customer is encouraged thus customer can have more opportunities to see the working software and check if it satisfies their needs (Shrivastava and Srivastava, 2020). Agile methodology is more suitable with small project instead of large projects as estimation of time needed for developing large project is harder using this methodology.

2.3.4 Comparison of software development methodologies

Table 2.2: Comparison of Methodologies

Methodology	**Strengths**	**Weaknesses**
Waterfall	• Easy to understand and use • Good and proper documentation • Easier to manage	• All requirements must be clearly specified before development • Difficult to go backwards, hard to incorporate changes to requirements afterwards • Testing is done at late stage

		• Little involvement of customer or end user
Prototyping	• Can adapt to changes in requirements • Early customer feedback • Better understanding on customer needs • Identify missing functionalities • Errors can be identified earlier	• May experience too much change requests • Increase cost of development • May be time consuming • Scope of project might be expanded
Agile	• Can adapt to changes in requirements • Early testing • Deliver the project quickly • Frequent customer feedback	• Lack of proper documentation • Not suitable for large project

2.3.5 Summary

After reviewing the different methodologies, waterfall methodology is not chosen as the requirements must be clearly specified in advance and are difficult to be changed during development. Agile methodology is not chosen because the period of each iteration has to be done in a very fast pace and must be very familiar with the technology used. Prototype methodology is the most suitable as the features of the application can be developed in stages and the time period of each iteration is not as constraint as in agile methodology. Besides, requirements can be amended throughout the development.

2.4 Review on mobile application development frameworks

2.4.1 React Native

React Native is a framework for building native mobile applications using JavaScript (Novick, 2017). Applications developed in React Native can be run on both Android and iOS platform without having to write the code in native programming languages such Java, Kotlin or Swift. The application developed looks and feels similar to applications built by native languages (Skuza, Mroczkowska and Włodarczyk, 2019). React Native is developed based on the ReactJS which is a library also developed by Facebook. React Native is considered as one of the best choices for developing mobile applications. React Native applications can be built on Windows, Mac and Linux operating system. Expo can be used to work with React Native application development to ease some tasks in the development. The target platform of the application must be at least iOS 10.0 and Android 4.1. React Native is backed by large community. According to Octoverse (2018), it ranked as second top open source project with 10k contributors. The main repository for React Native is GitHub. Examples of popular applications such as Instagram, Facebook, Pinterest are built using React Native. React Native uses components, states, props and JSX for development.

Table 2.3: React Native Advantages and Disadvantages

Advantages	**Disadvantages**
Written in JavaScript which is widely used	Implementation of some features still need knowledge on native mobile application programming languages
Cross-platform mobile application development with single codebase	The framework is still considered quite new.
Saves cost as application can be built once and deployed on various platform	Not suitable for computation and memory intensive applications
Large community support	
Hot reload feature saves compilation and development time	

Code and component reusability	
Third-party plugins are supported	

2.4.2 Flutter

Flutter is developed by Google in 2017 (Singh, 2020). Flutter uses Dart as development programming language. By using Flutter, the mobile application developed can be delivered in multiple platforms such as Android and iOS. Flutter’s application development is based on widgets. The user interface is developed by combining the widgets together. It is inspired by ReactJS library. Operating system that supports Flutter’s application development includes Windows, Mac and Linux. Flutter is relatively new compared to other frameworks but it had gained popularity over the years. According to a survey by Stack Overflow (2019), 75.4% of respondents had chosen Flutter as the most loved frameworks. Applications such as Google Ads, Xianyu by Alibaba and Hamilton by Broadway.

Table 2.4: Flutter Advantages and Disadvantages

Advantages	**Disadvantages**
Low development cost, once written can be deployed on multiple platforms	Developers have to learn new language
Cross-platform mobile application development with single codebase	Very new technology and still considered immature
Code reusability	Community support is still growing
High performance as it does not rely on device’s OEM widgets	Dart programming language is less popular
Hot reload results in less development time	Application occupies more space
	Limited third-party libraries

2.4.3 Comparison between mobile application frameworks

Table 2.5: Comparison between React Native and Flutter

Framework Aspect	React Native	Flutter
Developed by	Facebook	Google
Programming language	JavaScript	Dart
Released in	2015	2017
Maturity	More matured compared to Flutter	Relatively immature and new
Third-party library plug-in and support	Wide range	Smaller
Application performance	Lower	Higher
Community support	Larger	Smaller
Documentation	Unclear compared to Flutter	Clear documentation
Application size	Smaller	Larger

2.4.4 Summary

React Native will be chosen as the framework used for the development of proposed application after comparison made between React Native and Flutter. Flutter is not chosen as it is relatively new and limited familiarity to Dart programming language. Besides, the community support is lesser than React Native. There is also lack of third-party libraries or plug-ins support for location tracking which will be needed in the development.

2.5 Conclusion

Literature review was done on the existing applications, software development methodologies and mobile application frameworks. The existing applications reviewed were used as reference to develop the features to be included in UTAR Vehicles Tracking App. The advantages and disadvantages of various development methodologies and mobile application frameworks were analysed and compared to choose the most suitable option to be adopted in UTAR Vehicles Tracking App.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

The chosen methodology will be discussed in section 3.2 and the development tools will be discussed in section 3.3. Section 3.4 will be focusing on the project plan.

3.2 Chosen methodology

After studies and reviews done on available software development methodologies in Chapter 2, prototyping methodology was chosen. The specific prototype methodology chosen was evolutionary prototype methodology.

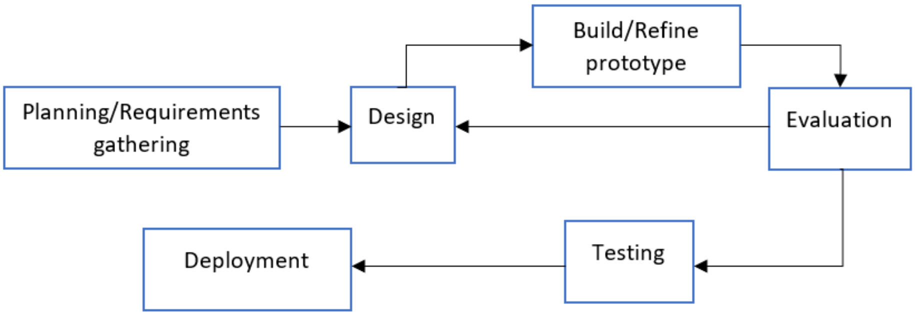


Figure 3.1: Evolutionary Prototype Methodology

Evolutionary prototype methodology was chosen as it allows user involvement in the development process. After requirements were gathered, prototype was produced for supervisor’s evaluation to confirm about the requirements and functions of the application to be developed. Feedbacks gathered after the review was incorporated in the prototype and sent for evaluation again for further improvements. Bugs can be discovered and fixed earlier. Besides, the features of the proposed application were implemented in various iteration instead of developing all required features at one time. As few features were being developed at one time, more focus can be given while implementing each of the feature. Through prototyping, the system produce can better meet the user expectations and satisfaction.

The first phase in this methodology was requirements gathering or planning. The project objectives to be achieved were defined at this phase. The problems statement and the scope of the project, the assumptions made, and target users was identified in this phase. Requirements were collected through reviewing similar existing applications and from the project supervisor. Through reviewing existing applications, general features that are essential in the application and additional features that are useful but have not been provided were identified. The overall workflow of the application was also being identified. The general and additional features collected was reviewed to determine whether they are suitable to be included in the proposed application. After that, functional and non-functional requirements will be specified.

After requirements gathering and initial specification, design phase was started. Use case modelling was carried out. The use case diagram and use case description for each function to be implemented were developed.

The project then entered the prototype development iteration. This project had three iterations of prototype. The first prototype built was only focusing on the frontend development to demonstrate the user interface and the user flow of the application. The first prototype was presented to project supervisor for evaluation. Feedbacks gathered were used to make improvements in the prototype. After that, the development entered second iteration. The second and third iteration of prototype development were focusing on both frontend and backend development. The architecture design and database design were created. The functions or requirements of the proposed application were implemented in the prototype. The functions to be developed were assigned accordingly to each iteration. Each iteration involved implementation of few functions. After the development of second and third prototype ended, it was sent for evaluation. Feedbacks were collected. After evaluation, the prototype was improved according to the feedbacks. When the prototype in third iteration was approved, it was being developed into the complete application.

The project entered testing phase after the iteration of prototype development ends. The application undergone unit testing, integration testing and user acceptance testing. Tools that were used are Jest and Postman. Jest will be used to test the mobile application whereas Postman will be used for testing route endpoints for API. User acceptance testing test results were collected and recorded through Google Form. After all testing passed, the development of the application is complete and ready to deliver to users.

3.3 Development Tools

UTAR Vehicles Tracking App had a mobile application for UTAR drivers, students and staffs and also a web application for UTAR transport section administrative staffs. Hence, the development tools involved mobile and web application development tools.

3.3.1 JavaScript

JavaScript is a scripting language for client and server application development. By using JavaScript, developers can build more interactive web pages. JavaScript is chosen as the programming language because it is easy to learn and has strong community support. Besides, it is ranked as top technologies for software development in Pluralsight Technology Index (Pluralsight, n.d.), indicating it is a popular language among developers and will be one of the future trends. It was the main programming language being used to develop the mobile version of proposed application in React Native.

3.3.2 React Native

React Native was used to develop UTAR Vehicles Tracking App. It uses JavaScript and JSX as the programming language. React Native was being chosen as it supports a lot of third-party plugins and is backed by a large community support. The hot reload feature also helps to save development time of the application. Besides, React Native is free and open source.

3.3.3 Google Maps Platform

Google Maps Platform provide APIs and SDKs for developers to include the Google Maps in their applications. Maps SDK for Android was used to display the map in the mobile application for UTAR students and staffs. Other than that, Directions API was used to illustrate the route on the map and for retrieval of travel duration according to traffic condition and distance.

3.3.4 Socket.IO

Socket.IO enables real-time two-way communication between client and server (Socket.IO, n.d.). The communication is based on events. Socket.IO was used for sending and receiving of location information updates between the driver's and students and staffs' mobile application.

3.3.5 Node.js and Express

Node.js provides JavaScript runtime environment that runs on Chrome's V8 JavaScript engine (Node.js, n.d.). React Native used Node.js to manage the open-source packages and as development server for debugging and live updates (Ortiz, 2018). On the other hand, Express is a Node.js web application framework that can be used to build API quick and easy (Express, n.d.). It is also minimal and flexible. It was used as the HTTP server and to create the API to connect the mobile application to the database. Besides, the Socket.IO server was also implemented in Express server.

3.3.6 Laravel

Laravel is a web application framework that is based on PHP and uses MVC design pattern. It is open-source, robust and easy to understand. Laravel increases the scalability of applications and provides a wide range of features that ease the development of web applications. Laravel was used as the framework for developing the web application for UTAR transport section administrative staffs.

3.3.7 MySQL

MySQL is among the most popular database used in developing software applications. MySQL is an open-source relational database based on SQL. MySQL was chosen as it is reliable and easy to set up and use. It also offers high performance and scalability. Apart from that, it can be easily connected to React Native application and Laravel application.

3.3.8 Visual Studio Code

The editor chosen to write all source codes for the proposed application was Visual Studio Code. It is free and support multiple programming languages and provides support of connecting with Git. Besides, there is also a large number of extensions available for use that allows support for different programming languages.

3.3.9 Android Studio

Android Studio is developed by Google and can be used for mobile application development for Android devices (Google Developers, n.d.). Android Studio was used to provide an emulator to preview the application being developed.

3.4 Project plan

The work breakdown structure and Gantt chart were included in the Appendices (APPENDIX A: Work Breakdown Structure, APPENDIX B: Gantt Chart).

3.5 Conclusion

Project planning activities carried out include the development methodology chosen and development tools used, work breakdown structure and Gantt chart to describe the tasks and its duration as guidance for the execution of this project so that it could be completed on time.

CHAPTER 4

PROJECT SPECIFICATION

4.1 Introduction

This chapter includes section 4.2 requirements specification and section 4.3 use case modelling of the proposed application.

4.2 Requirements Specification

The requirements of the proposed application were identified through sampling and reviewing of similar existing applications and from the project supervisor. The findings from the observation are included in Chapter 2 Literature Review Section 2.2 Review on existing applications.

The functional requirements were specified according to the roles of target users of the proposed application. The roles include UTAR drivers, UTAR students and staffs and UTAR transport section administrative staffs.

4.2.1 Functional Requirements

4.2.1.1 UTAR drivers

1. UTAR Vehicles Tracking App shall be able to get the location of the vehicle.
2. UTAR Vehicles Tracking App shall allow user to provide real time update on current vehicle status.
3. UTAR Vehicles Tracking App shall be able to send the location and message updates through web sockets.

4.2.1.2 UTAR students and staffs

1. **Track real time vehicle location**
    a. UTAR Vehicles Tracking App shall be able to display the list of operating routes.
    b. UTAR Vehicles Tracking App shall be able to refresh list of operating routes after user pull down the screen.
    c. UTAR Vehicles Tracking App shall be able to display map of the selected route.

   d. UTAR Vehicles Tracking App shall be able to display real time location of the vehicle on the map.
   e. UTAR Vehicles Tracking App shall be able to display the location of stops on the route map.

2. **Display real time vehicle information**
   a. UTAR Vehicles Tracking App shall allow users to select route and stop to display real time vehicle information.
   b. UTAR Vehicles Tracking App shall be able to display the estimated time of arrival of the vehicle.
   c. UTAR Vehicles Tracking App shall be able to display the vehicle plate number.
   d. UTAR Vehicles Tracking App shall be able to display the driver's name.
   e. UTAR Vehicles Tracking App shall display the real time message update of current vehicle status from the driver.
3. **Display route information**
   a. UTAR Vehicles Tracking App shall be able to display the routes available.
   b. UTAR Vehicles Tracking App shall be able to display the stops on the route according to correct sequence.
   c. UTAR Vehicles Tracking App shall be able to display the stops on the route map.
4. **Display schedule**
   a. UTAR Vehicles Tracking App shall be able to display the time schedule of the stops on the route according to the route and date selected.

4.2.1.3 UTAR Transport Section administrative staffs

1. **View travel record**
   a. UTAR Vehicles Tracking App shall be able to display the driver and vehicle information of each route trip.
   b. UTAR Vehicles Tracking App shall be able to display the scheduled and actual start time of each route trip.

   c. UTAR Vehicles Tracking App shall be able to display the status of each route trip.

2. **Manage schedule**
   a. UTAR Vehicles Tracking App shall allow user to view schedule of selected route.
   b. UTAR Vehicles Tracking App shall be able to add new schedule of selected route.
   c. UTAR Vehicles Tracking App shall be able to update schedule of selected route.
   d. UTAR Vehicles Tracking App shall be able to delete schedule of selected route.
3. **Manage drivers**
   a. UTAR Vehicles Tracking App shall allow user to view drivers' information.
   b. UTAR Vehicles Tracking App shall be able to add new driver.
   c. UTAR Vehicles Tracking App shall be able to update driver.
   d. UTAR Vehicles Tracking App shall be able to delete driver.

4.2.2 Non-functional Requirements

1. UTAR Vehicles Tracking App shall be able to be accessed by multiple users concurrently.
2. UTAR Vehicles Tracking App shall be able to prevent unauthorised user login.
3. UTAR Vehicles Tracking App shall have consistent and easy-to-use interface.
4. UTAR Vehicles Tracking App shall be able to provide the accurate vehicle location.

4.3 Use Case Modelling

4.3.1 Use Case Diagram

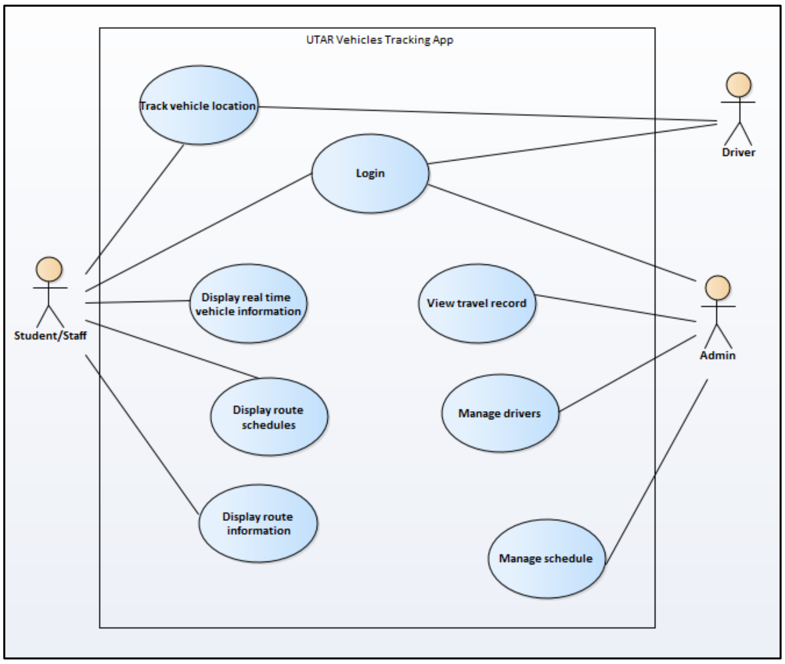


Figure 4.1: Use Case Diagram

4.3.2 Use Case Descriptions

4.3.2.1 Login

Table 4.1: Use Case Description – Login

Use Case Name: Login	Actor: Driver, Student/Staff, Admin
Description: The login process of user to the application.	
Normal Flow of Events: 1. The application displays the login screen. 2. User enters the id and password. 3. User presses Login button. 4. The application checks user credentials and redirects to home screen.	

Alternate Flow: 2.1. Admin login 2.1.1. User enters email and password. 3.1. Incorrect id or password 3.1.1. The id and password entered by user was incorrect. 3.1.2. The application display error message. 3.2. Did not fill in login credentials 3.2.1. The id and password were not entered. 3.2.2. The application display error message.

4.3.2.2 Track vehicle location

Table 4.2: Use Case Description - Track Vehicle Location (Driver)

Use Case Name: Track vehicle location	Actor: Driver
Description: The process to start/stop vehicle location tracking and update real time vehicle status.	
Normal Flow of Events: 1. User selects the scheduled trip to be operated. 2. User selects Start Tracking button. 3. The application starts sending real time vehicle location updates. 4. The application display message updates screen. 5. User selects status messages from options displayed. 6. The application sends message update. 7. User selects Stop Tracking button. 8. The application stops sending real time vehicle location updates. 9. The application redirects to previous screen.	
Alternate Flow: -	

Table 4.3: Use Case Description - Track Vehicle Location (Student/Staff)

Use Case Name: Track vehicle location	Actor: Student/staff

Description: The process for student/staff to track vehicle location.
Normal Flow of Events: 1. User selects Track Vehicle button on the menu bar. 2. User selects the route. 3. The application displays the real time vehicle location on the map. 4. The application displays the stops on the map. 5. The application updates vehicle location on the map when it changes.
Alternate Flow: 5.1 No operating route trip 5.1.1 The application display message that there is no operating route trip currently.

4.3.2.3 Display real time vehicle information

Table 4.4: Use Case Description - Display Real Time Vehicle Location

Use Case Name: Display real time vehicle information	Actor: Student/staff
Description: The process for student/staff to view real time vehicle information.	
Normal Flow of Events: 1. User selects Track Vehicle button on the menu bar. 2. User selects the route. 3. The application displays the route trip information such as vehicle plate number, driver name and driver's message update. 4. The application displays the stops of the route in correct sequence. 5. User selects a stop. 6. The application displays the selected stop and estimated time to arrival of the stop.	
Alternate Flow: -	

4.3.2.4 Display route schedule

Table 4.5: Use Case Description - Display Route Schedule

Use Case Name: Display route schedule	Actor: Student/staff
Description: The process for student/staff to view pre-determined route schedule.	
Normal Flow of Events: 1. User selects Schedule button on the menu bar. 2. User chooses a route. 3. User chooses a date. 4. The application displays the time schedule.	
Alternate Flow: 4.1 No transport service available for the selected route on the selected day 4.1.1 The application display message that no schedule found.	

4.3.2.5 Display route information

Table 4.6: Use Case Description - Display Route Information

Use Case Name: Display route information	Actor: Student/staff
Description: The process for student/staff to view route information.	
Normal Flow of Events: 1. User selects Route button on the menu bar. 2. User selects the route. 3. The application displays a map. 4. The application indicates the route, stops and stop sequence.	
Alternate Flow: -	

4.3.2.6 View travel record

Table 4.7: Use Case Description - View Travel Record

Use Case Name: View travel record	Actor: Admin
Description: The process for administrative staff to view travel records.	
Normal Flow of Events: 1. User selects Travel Record button from menu bar. 2. The application displays the route name, driver name, vehicle plate number, date, scheduled start time, actual start time and status.	
Alternate Flow: -	

4.3.2.7 Manage schedule

Table 4.8: Use Case Description - Manage Schedule

Use Case Name: Manage schedule	Actor: Admin
Description: The process for administrative staff to view, add, update, delete schedule information such as scheduled arrival time for each bus stop, schedule period, driver and vehicle assigned.	
Normal Flow of Events: View schedule 1. User selects Route Schedule button from menu bar. 2. The system displays the routes available. 3. User selects the route. 4. The application displays schedules for the route. Add schedule 1. User selects Route Schedule button from menu bar. 2. The application displays the routes available. 3. User selects the route. 4. The application displays schedules for the route. 5. User selects Add button. 6. The application displays the form on the page.	

7. User can add new schedule information.
8. User selects Save to save the information.
9. The application saves and updates the schedule information.

Update schedule

1. User selects Route Schedule button from menu bar.
2. The application displays the routes available.
3. User selects the route.
4. The application displays schedule for the route.
5. User selects View button on the schedule record.
6. The application displays the schedule details of selected record on next page.
7. User selects Edit button.
8. The application displays the form on the page.
9. User can edit the information.
10. User selects Save to save the updated information.
11. The application saves and updates the schedule information.
12. The application redirects to previous page.

Delete schedule

1. User selects Schedule button from menu bar.
2. The application displays the routes available.
3. User selects the route.
4. The application displays schedule for the route.
5. User selects View button on the schedule record.
6. The application displays the schedule details of selected record on next page.
7. User selects Delete button.
8. The application deletes the schedule record.
9. The application redirects to previous page.

Alternate Flow: -

Add schedule

7.1 User selects Cancel button.

7.1.1 The application dismisses the modal form displayed.

Update schedule 9.1 User selects Cancel button. 9.1.1 The application dismisses the modal form displayed.

4.3.2.8 Manage drivers

Table 4.9: Use Case Description - Manage Drivers

Use Case Name: Manage drivers	Actor: Admin
Description: The process for administrative staff to view, update, delete driver information.	
Normal Flow of Events: View driver 1. User selects Drivers button from menu bar. 2. The application displays the drivers’ information. Add driver 1. User selects Driver button from menu bar. 2. The application displays drivers’ information. 3. User selects Add button. 4. The application displays the form on the page. 5. User can add new driver information. 6. User selects Save to save the information. 7. The application saves and updates the driver information. Update driver 1. User selects Drivers button from menu bar. 2. The application displays the drivers’ information. 3. User selects Edit button on the drivers record user wishes to update. 4. The application displays the selected driver’s information on the page. 5. User can edit the information. 6. User selects Save to save the updated information. 7. The application saves and updates the driver information.	

Delete driver 1. User selects Drivers button from menu bar. 2. The application displays the drivers' information. 3. User selects Delete button on the driver's record user wishes to delete. 4. The application deletes the driver's record.
Alternate Flow: - Add driver 5.1 User clicks Cancel button. 5.1.1 The application dismisses the modal form displayed. Update driver 5.1 User clicks Cancel button. 5.1.1 The application dismisses the modal form displayed.

4.4 Conclusion

The functional and non-functional requirements for the UTAR Vehicles Tracking App were specified in detail in this chapter. Besides, use case diagram was created to illustrate the use cases and users of UTAR Vehicles Tracking App as part of requirements specification phase activities. Use case description was provided for each use case to describe the interactions of the user with the application and the possible flow of events.

CHAPTER 5

DESIGN

5.1 Introduction

This chapter includes section 5.2 software architecture design, section 5.3 database design and section 5.4 user interface design of UTAR Vehicles Tracking App.

5.2 Software Architecture Design

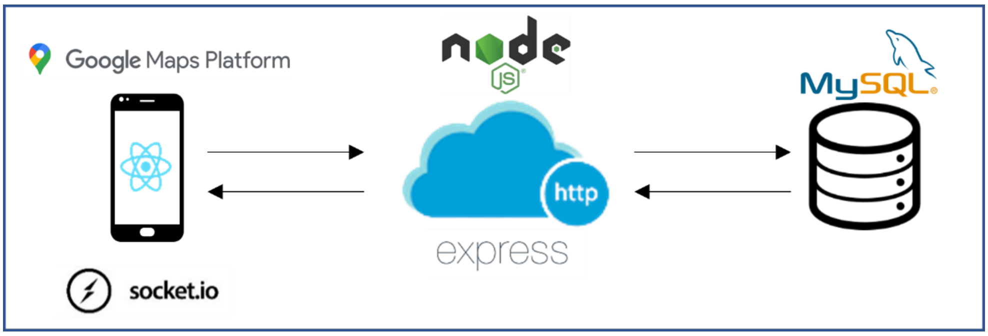


Figure 5.1: Client Server Architecture

Client-server architecture was used for the driver's and students and staffs' mobile application version. Information such as routes, drivers and schedule information are retrieved from database to the mobile application with the help of REST API. The requests were sent from the client which will be the mobile application to the server and through HTTP. The server then processed the requests, retrieved data from the database and sent responses back to the client. Besides, the location updates of the drivers were sent to the backend server through socket, then the server broadcasted information received to connected socket clients.

Node.js and Express was used to write the REST API for connecting the application to the backend database. Socket.IO was used for sending location and message updates between socket server and connected clients. React Native was used to develop the front end for the mobile application for UTAR drivers and UTAR students and staffs. Google Maps Platform API such as Maps SDK for Android and Directions API was used for the location tracking, displaying route map and retrieving estimated remaining time to arrival. MySQL database was used to store information such as routes, drivers and schedule information.

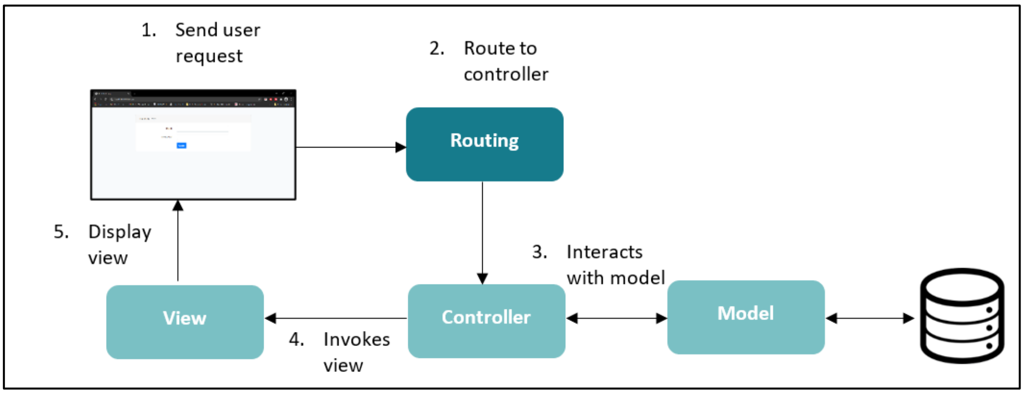


Figure 5.2: MVC Architecture

For the web application used by UTAR transport section, Laravel was used as the web application framework. Laravel uses a Model-View-Controller architecture. User input or request were sent from the view to the controller through routing. The controller performed application logics and accessed the model. The model accessed the database to retrieve or save data depending on user’s action. The model then returned data back to the controller and the controller invoked the view to display the results. The web application used the same MySQL database as the mobile application to store information.

5.3 Database Design

5.3.1 Entity Relationship Diagram

Figure 5.3 shows the entity relationship diagram of UTAR Vehicles Tracking App. The entity relationship diagram shows the relationship between each database tables in the database. A total of 9 database tables were created to store the application users, routes, schedules, travel records, vehicles and driver information.

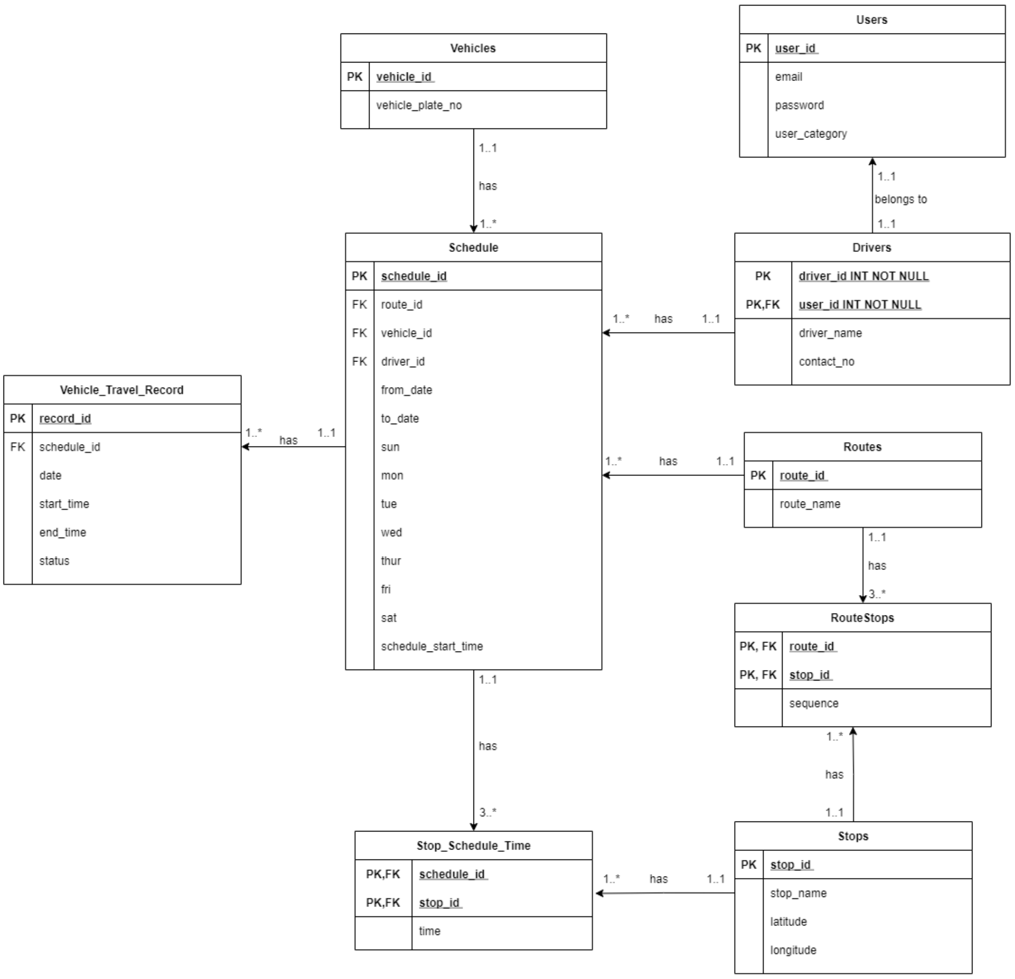


Figure 5.3: Entity Relationship Diagram

5.3.2 Data Dictionary

Table Name: users

Table 5.1: Users Table

Attribute	**Description**	**Data Type**	**PK/FK**	**FK Reference Table**
user_id	Unique ID for all users	Int	PK	-
email	User's email	Varchar	-	-
password	User's password	Varchar	-	-
user_category	User's category (student/staff/driver /admin)	Varchar	-	-

Table Name: drivers

Table 5.2: Drivers Table

Attribute	**Description**	**Data Type**	**PK/FK**	**FK Reference Table**
driver_id	Unique ID for all drivers	Int	PK	-
user_id	Driver's user ID	Int	FK	users
driver_name	Driver's name	Varchar	-	-
contact_no	Driver's contact number	Varchar	-	-

Table Name: vehicles

Table 5.3: Vehicles Table

Attribute	**Description**	**Data Type**	**PK/FK**	**FK Reference Table**
vehicle_id	Unique ID for all vehicles	Int	PK	-
vehicle_plate_no	Vehicle's plate number	Varchar	-	-

Table Name: routes

Table 5.4: Routes Table

Attribute	**Description**	**Data Type**	**PK/FK**	**FK Reference Table**
route_id	Unique ID	Int	PK	-

	for all routes			
route_name	Route’s name	Varchar	-	-

Table Name: stops

Table 5.5: Stops Table

Attribute	**Description**	**Data Type**	**PK/FK**	**FK Reference Table**
stop_id	Unique ID for all stops	Int	PK	-
stop_name	Stop’s name	Varchar	-	-
latitude	Latitude of stop’s location	Varchar	-	-
longitude	Longitude of stop’s location	Varchar	-	-

Table Name: routestops

Table 5.6: Routestops Table

Attribute	**Description**	**Data Type**	**PK/FK**	**FK Reference Table**
route_id	Unique ID for all routes	Int	PK, FK	routes
stop_id	Unique ID for all stops	Int	PK, FK	stops
sequence	Sequence of the stop for the route	Int	-	-

Table Name: schedules

Table 5.7: Schedules Table

Attribute	Description	Data Type	PK/FK	FK Reference Table
schedule_id	Unique ID for all schedules	Int	PK	-
route_id	Unique ID for all routes	Int	FK	routes
vehicle_id	Unique ID for all vehicles	Int	FK	vehicles
driver_id	Unique ID for all drivers	Int	FK	drivers
from_date	Starting date of the schedule	Date	-	-
to_date	End date of the schedule	Date	-	-
sun	Sunday	Boolean	-	-
mon	Monday	Boolean	-	-
tue	Tuesday	Boolean	-	-
wed	Wednesday	Boolean	-	-
thur	Thursday	Boolean	-	-
fri	Friday	Boolean	-	-
sat	Saturday	Boolean	-	-
schedule_start_time	Starting time of the schedule	Time	-	-

Table Name: stop_schedule_time

Table 5.8: Stop_Schedule_Time Table

Attribute	**Description**	**Data Type**	**PK/FK**	**FK Reference Table**
schedule_id	Unique ID for all schedules	Int	PK, FK	schedules
stop_id	Unique ID for all stops	Int	PK, FK	stops
time	Scheduled arrival time for the stop	Time	-	-

Table Name: vehicle_travel_record

Table 5.9: Vehicle_Travel_Record Table

Attribute	**Description**	**Data Type**	**PK/FK**	**FK Reference Table**
record_id	Unique ID for all travel records	Int	PK	-
schedule_id	Unique ID for all schedules	Int	FK	schedules
date	Date where the record is created	Date	-	-
start_time	Time where the driver starts tracking for the schedule	Time	-	-
end_time	Time where the driver stops tracking for the schedule	Time	-	-

status	Status of the schedule travelled (operating/ended)	Varchar	-	-

5.4 User Interface Design

5.4.1 UTAR Student and Staff Application

The mobile application for students and staffs consists of 5 interfaces. The interfaces include login page, routes page, route information page, track vehicle page, track vehicle page and schedules page.

The login page is shown in Figure 5.4. User shall enter the ID and password to login. After successful login, the page will be redirected to the routes page in default as shown in Figure 5.5.

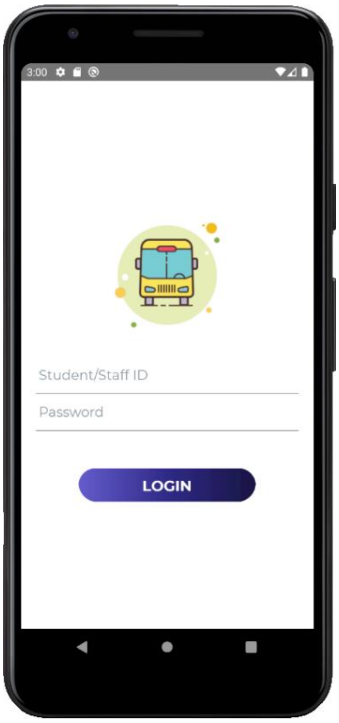
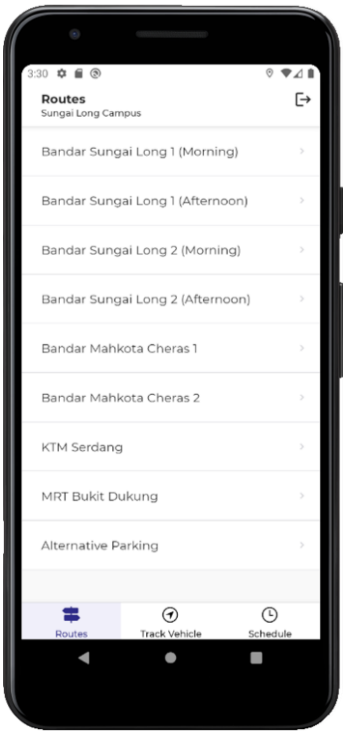


Figure 5.4: Login - UTAR Student and Staff Application

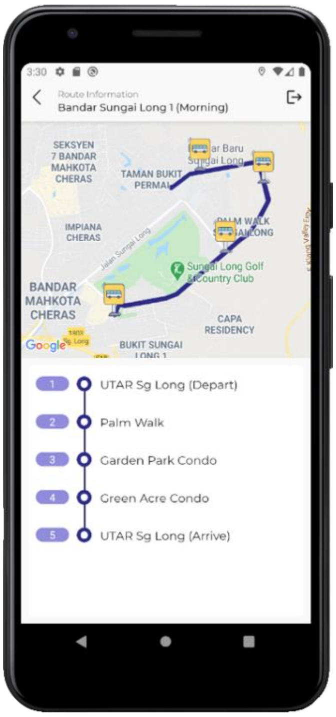


Routes
Sungai Long Campus

- Bandar Sungai Long 1 (Morning)
- Bandar Sungai Long 1 (Afternoon)
- Bandar Sungai Long 2 (Morning)
- Bandar Sungai Long 2 (Afternoon)
- Bandar Mahkota Cheras 1
- Bandar Mahkota Cheras 2
- KTM Serdang
- MRT Bukit Dukung
- Alternative Parking

Figure 5.5: Routes Available

User can select other options on the bottom tab menu as shown in Figure 5.5. When user selects a route display on the routes page, the route information will be displayed as shown in Figure 5.6.

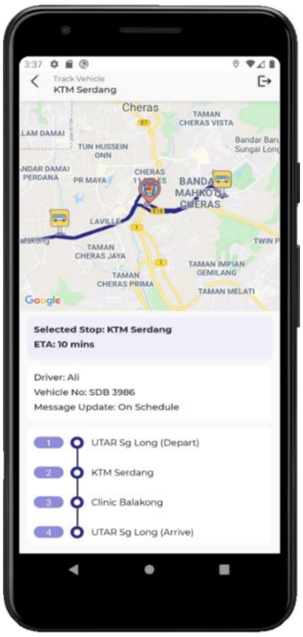


Route Information
Bandar Sungai Long 1 (Morning)

1 UTAR Sg Long (Depart)
2 Palm Walk
3 Garden Park Condo
4 Green Acre Condo
5 UTAR Sg Long (Arrive)

Figure 5.6: Route Information

Track vehicle page will be displayed when user selects Track Vehicle option from the bottom tab menu and selects an operating route. The route map, stops and vehicle location will be shown. User can view the schedule information such as driver name, vehicle plate number and stop sequence from this page. The message updates, selected stop and its ETA will also be displayed to the user upon selection.



Track Vehicle
KTM Serdang

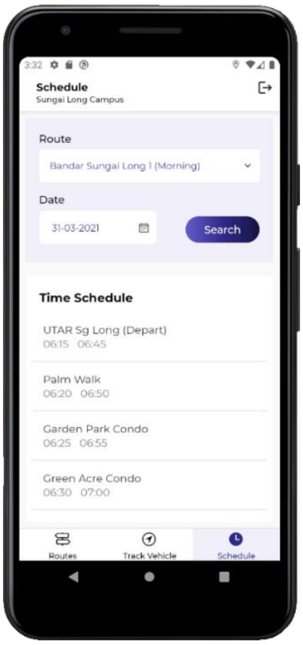
Selected Stop: KTM Serdang
ETA: 10 mins

Driver: Ali
Vehicle No: SDB 3986
Message Update: On Schedule

1 UTAR Sg Long (Depart)
2 KTM Serdang
3 Clinic Balakong
4 UTAR Sg Long (Arrive)

Figure 5.7: Track Vehicle

The interface for schedule information is shown in Figure 5.8. User shall select the route name and date to get the schedule.



Schedule
Sungai Long Campus

Route
Bandar Sungai Long 1 (Morning)

Date
31-03-2021

Time Schedule

UTAR Sg Long (Depart)
06:15 06:45

Palm Walk
06:20 06:50

Garden Park Condo
06:25 06:55

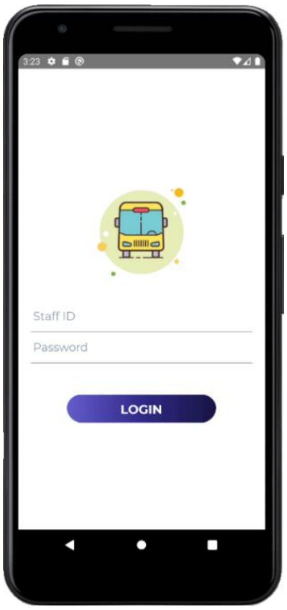
Green Acre Condo
06:30 07:00

Figure 5.8: Schedule Information

5.4.2 UTAR Driver Application

The mobile application for the drivers consists of 4 interfaces. The interfaces include schedules assigned page, start tracking page and message updates and stop tracking page.

The login page is shown in Figure 5.9. The page consists of 2 input fields, user shall enter the ID and password to login. After successful login, the page will be redirected to the schedules assigned page in default as shown in Figure 5.10.



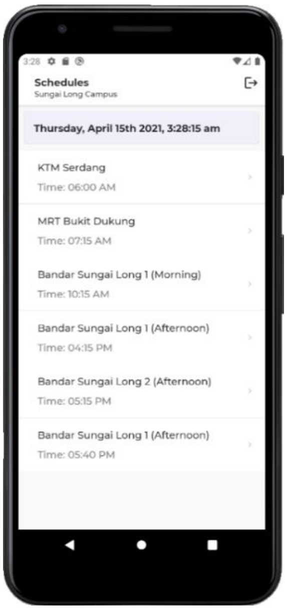
3:23

Staff ID

Password

LOGIN

Figure 5.9: Login - UTAR Driver Application



3:28

Schedules
Sungai Long Campus

Thursday, April 15th 2021, 3:28:15 am

KTM Serdang
Time: 06:00 AM

MRT Bukit Dukung
Time: 07:15 AM

Bandar Sungai Long 1 (Morning)
Time: 10:15 AM

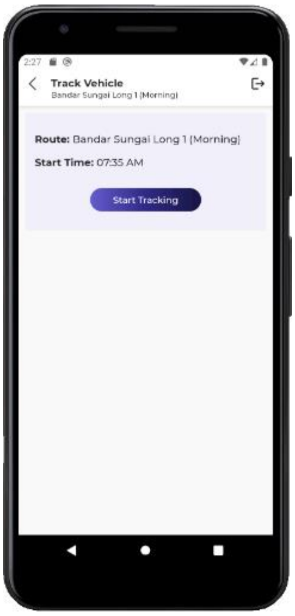
Bandar Sungai Long 1 (Afternoon)
Time: 04:15 PM

Bandar Sungai Long 2 (Afternoon)
Time: 05:15 PM

Bandar Sungai Long 1 (Afternoon)
Time: 05:40 PM

Figure 5.10: Schedules Assigned

The application will redirect to the start tracking page as shown in Figure 5.11 after user selected a schedule. After selecting Start Tracking button, the application will redirect to the page shown in Figure 5.12. User can send message updates and stop tracking from this page.



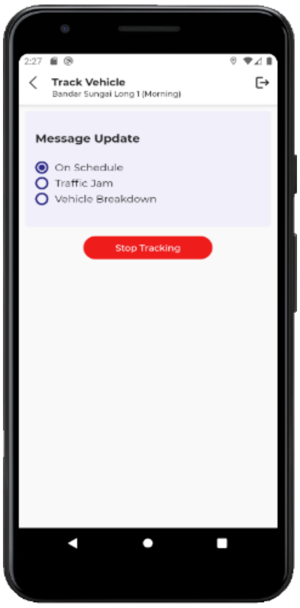
2:27

Track Vehicle
Bandar Sungai Long 1 (Morning)

Route: Bandar Sungai Long 1 (Morning)

Start Time: 07:35 AM

Figure 5.11: Start Tracking



2:27

Track Vehicle
Bandar Sungai Long 1 (Morning)

Message Update

● On Schedule
○ Traffic Jam
○ Vehicle Breakdown

Figure 5.12: Message Updates and Stop Tracking

5.4.3 UTAR Transport Section Administrative Staff Application

The web application for the administrative staffs consists of 11 interfaces. The interfaces include login page, homepage, travel record page, routes page, route schedules information page and driver information page. Modals are used to show the pop-up forms for adding or updating schedules and driver information.

Figure 5.13 shows the login page. User needs to enter email and password to login to the application. After successful login, the application will redirect to homepage shown in Figure 5.14. User can select the options displayed on the homepage or the side menu.

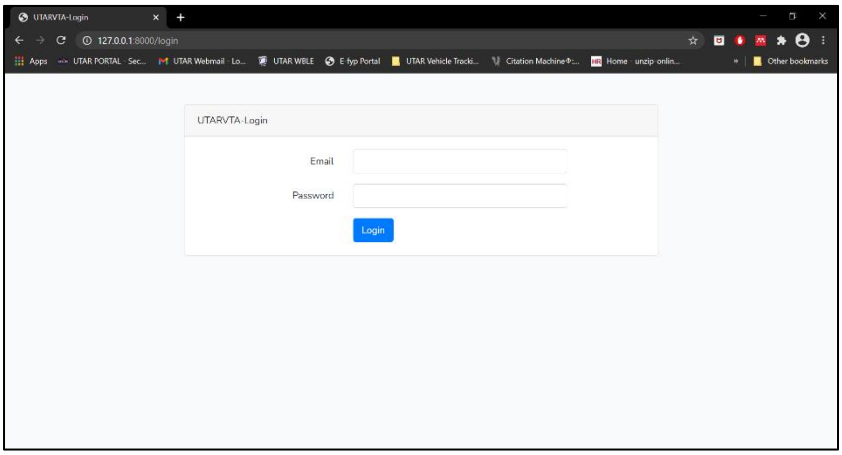


Figure 5.13: Login - UTAR Transport Section Application

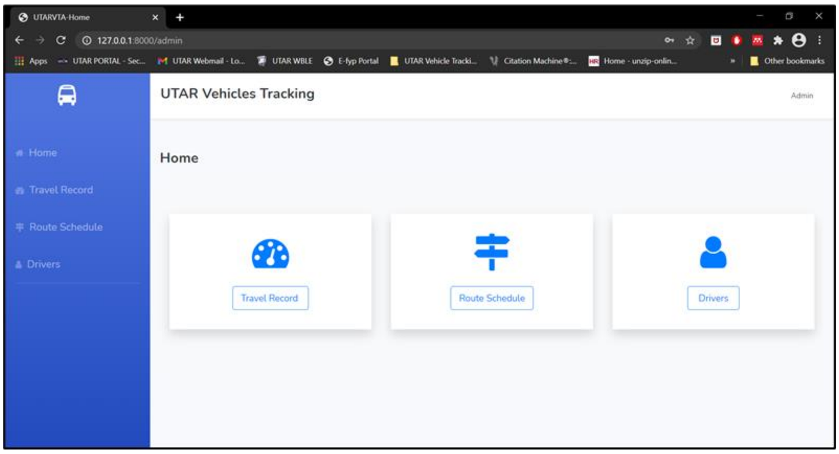
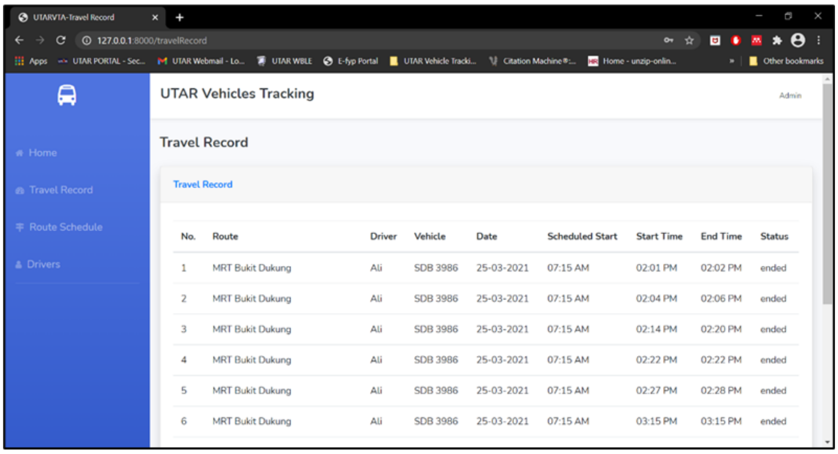


Figure 5.14: Homepage

When user selects Travel Record option, the travel records will be shown as in Figure 5.15.



UTAR Vehicles Tracking

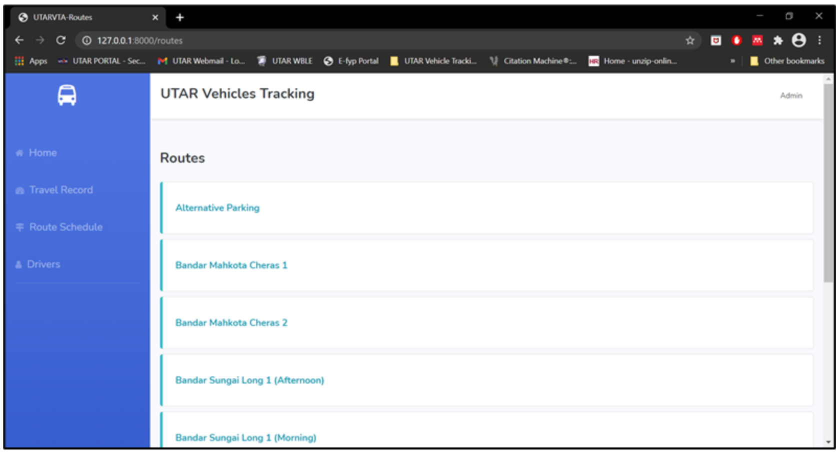
Travel Record

Travel Record

No.	Route	Driver	Vehicle	Date	Scheduled Start	Start Time	End Time	Status
1	MRT Bukit Dukung	Ali	SDB 3986	25-03-2021	07:15 AM	02:01 PM	02:02 PM	ended
2	MRT Bukit Dukung	Ali	SDB 3986	25-03-2021	07:15 AM	02:04 PM	02:06 PM	ended
3	MRT Bukit Dukung	Ali	SDB 3986	25-03-2021	07:15 AM	02:14 PM	02:20 PM	ended
4	MRT Bukit Dukung	Ali	SDB 3986	25-03-2021	07:15 AM	02:22 PM	02:22 PM	ended
5	MRT Bukit Dukung	Ali	SDB 3986	25-03-2021	07:15 AM	02:27 PM	02:28 PM	ended
6	MRT Bukit Dukung	Ali	SDB 3986	25-03-2021	07:15 AM	03:15 PM	03:15 PM	ended

Figure 5.15: Travel Record

Figure 5.16 shows the list of routes available. User can select a route from the list to view the route and schedule information page as shown in Figure 5.17.



UTAR Vehicles Tracking

Routes

Alternative Parking

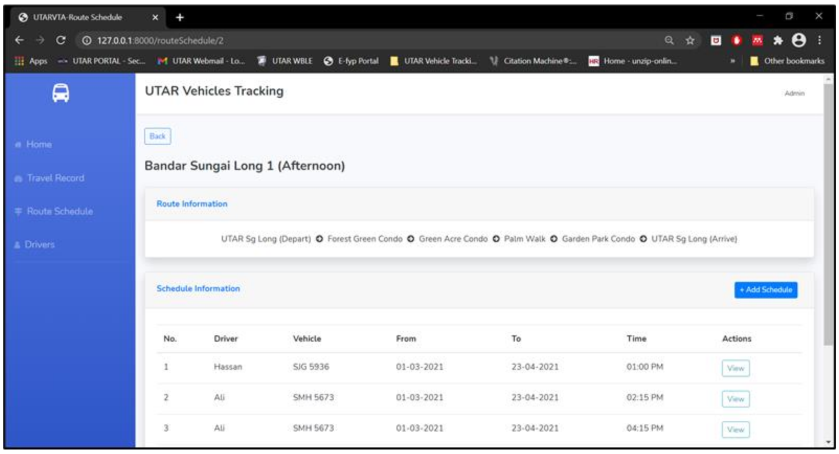
Bandar Mahkota Cheras 1

Bandar Mahkota Cheras 2

Bandar Sungai Long 1 (Afternoon)

Bandar Sungai Long 1 (Morning)

Figure 5.16: Routes



UTAR Vehicles Tracking

Bandar Sungai Long 1 (Afternoon)

Route Information

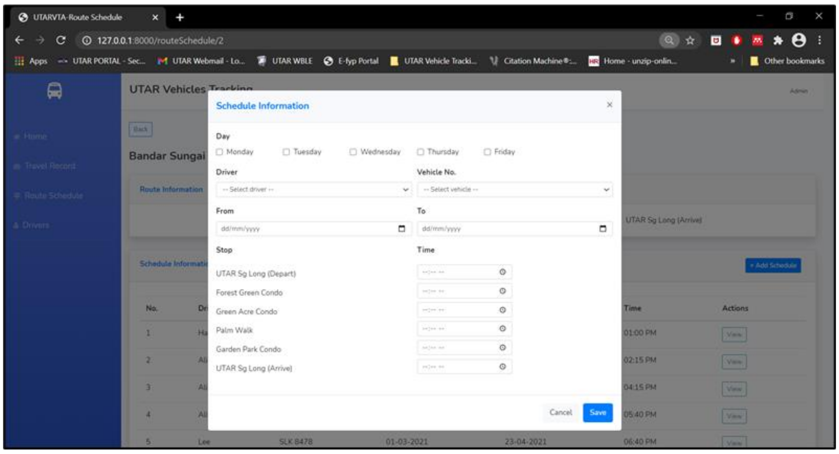
UTAR Sg Long (Depart) ● Forest Green Condo ● Green Acre Condo ● Palm Walk ● Garden Park Condo ● UTAR Sg Long (Arrive)

Schedule Information

No.	Driver	Vehicle	From	To	Time	Actions
1	Hassan	SJG 5936	01-03-2021	23-04-2021	01:00 PM	View
2	Ali	SMH 5673	01-03-2021	23-04-2021	02:15 PM	View
3	Ali	SMH 5673	01-03-2021	23-04-2021	04:15 PM	View

Figure 5.17: Route Schedule Information

User can add, update and view schedule details from the page shown in Figure 5.17. Figure 5.18 shows the modal form for adding new schedule, Figure 5.19 shows the page for displaying schedule details and Figure 5.20 shows the modal form for updating schedule details.



Schedule Information

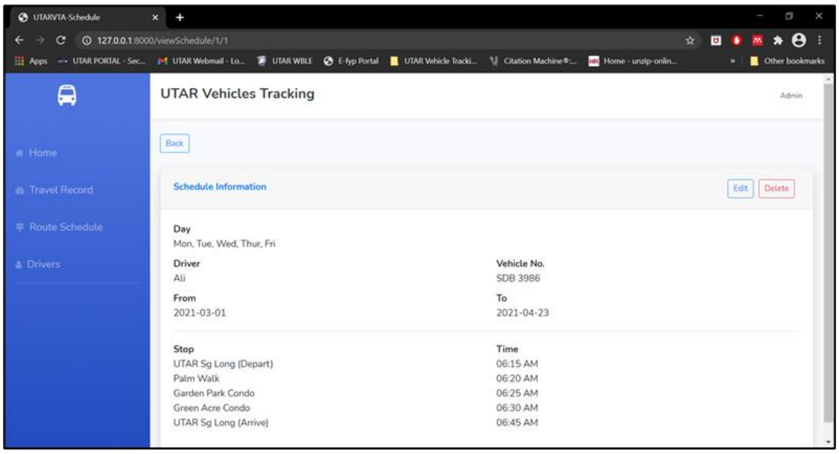
Day
☐ Monday ☐ Tuesday ☐ Wednesday ☐ Thursday ☐ Friday

Driver	Vehicle No.
-- Select driver --	-- Select vehicle --

From	To
dd/mm/yyyy	dd/mm/yyyy

Stop	Time
UTAR Sg Long (Depart)	--:-- --
Forest Green Condo	--:-- --
Green Acre Condo	--:-- --
Palm Walk	--:-- --
Garden Park Condo	--:-- --
UTAR Sg Long (Arrive)	--:-- --

Figure 5.18: Add Schedule



UTAR Vehicles Tracking

Schedule Information

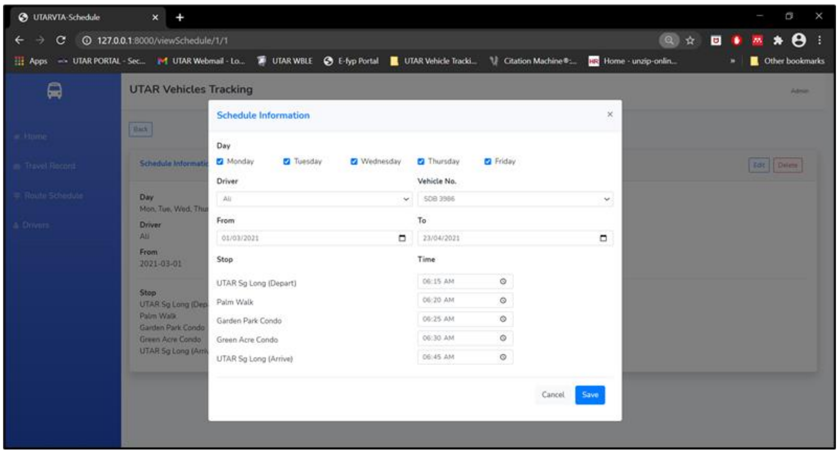
Day
Mon, Tue, Wed, Thur, Fri

Driver	Vehicle No.
Ali	SDB 3986

From	To
2021-03-01	2021-04-23

Stop	Time
UTAR Sg Long (Depart)	06:15 AM
Palm Walk	06:20 AM
Garden Park Condo	06:25 AM
Green Acre Condo	06:30 AM
UTAR Sg Long (Arrive)	06:45 AM

Figure 5.19: View Schedule Details



Schedule Information

Day
☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday ☑ Friday

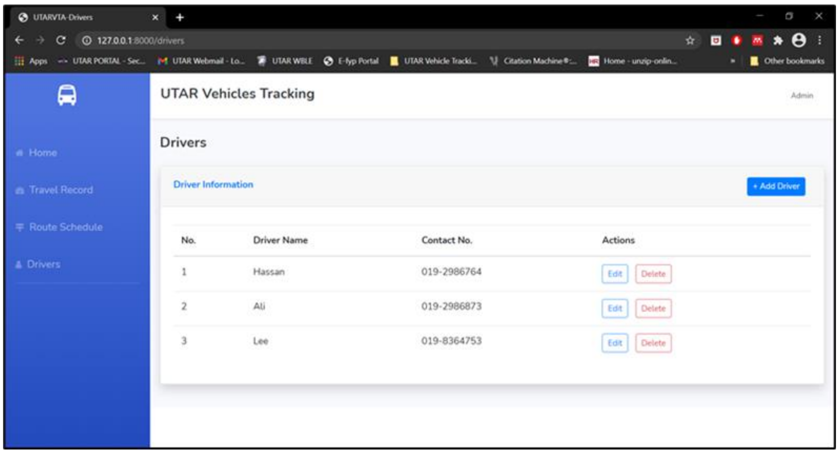
Driver	Vehicle No.
Ali	SDB 3986

From	To
01/03/2021	23/04/2021

Stop	Time
UTAR Sg Long (Depart)	06:15 AM
Palm Walk	06:20 AM
Garden Park Condo	06:25 AM
Green Acre Condo	06:30 AM
UTAR Sg Long (Arrive)	06:45 AM

Figure 5.20: Update Schedule

When user select Drivers option, the drivers page will be displayed as shown in Figure 5.21. User can add, update and view driver details from the page. Figure 5.22 shows the page for adding new driver while Figure 5.23 shows the modal form for updating driver details.



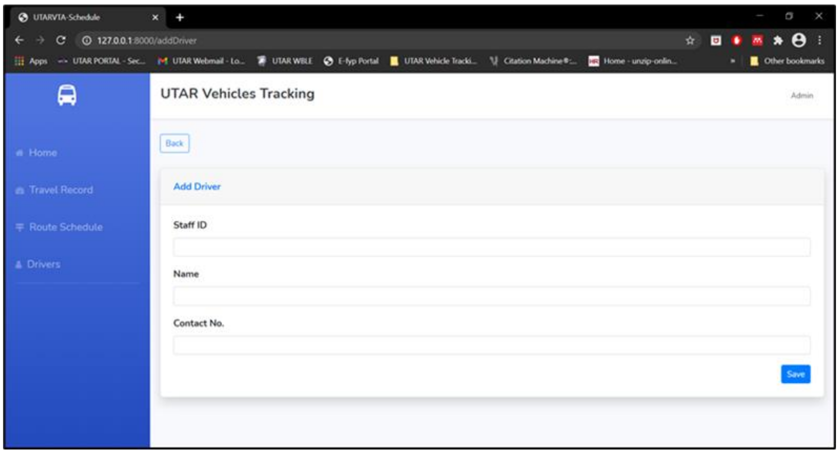
UTAR Vehicles Tracking

Drivers

Driver Information

No.	Driver Name	Contact No.	Actions
1	Hassan	019-2986764	Edit Delete
2	Ali	019-2986873	Edit Delete
3	Lee	019-8364753	Edit Delete

Figure 5.21: Drivers



UTAR Vehicles Tracking

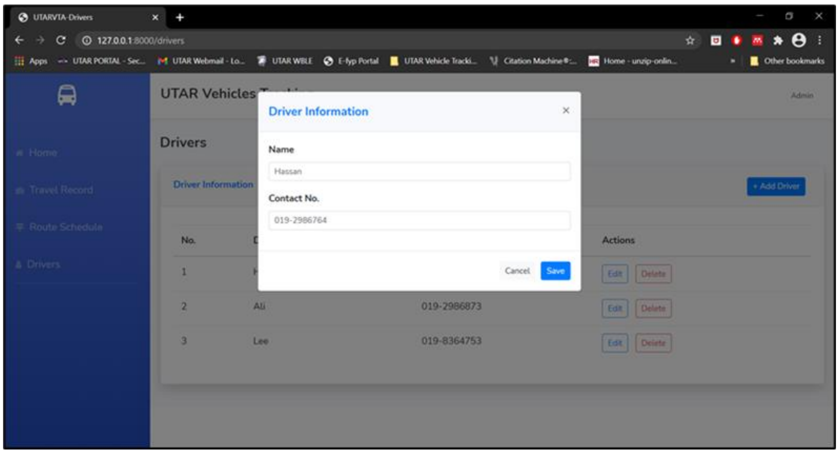
Add Driver

Staff ID

Name

Contact No.

Figure 5.22: Add Driver



Driver Information

Name: Hassan

Contact No.: 019-2986764

Figure 5.23: Update Driver

5.5 Conclusion

The work done for design phase includes software architecture design, database design and user interface design. The software architecture design for UTAR Vehicles Tracking App was illustrated. The database design was shown as entity relationship diagram and data dictionary was provided for each database table. The user interfaces for the students and staffs’ application, drivers’ application and transport section application were developed and the flow of interfaces were described.

CHAPTER 6

IMPLEMENTATION

6.1 Introduction

This chapter describes the detail implementation of UTAR Vehicles Tracking App. Section 6.2, 6.3 and 6.4 describe about the application flow for real time location tracking, real time message update and estimated time of arrival information implementation. Section 6.5 describes Socket.IO implementation, section 6.6 describes Google Maps Platform API implementation and section 6.7 describe route endpoints design for UTAR Vehicles Tracking App.

6.2 Real Time Location Tracking

Location tracking in UTAR Vehicles Tracking App was implemented using React Native's Geolocation API and Socket.IO. As the mobile application was on Android platform, the Geolocation API used android.location API to access driver's location. The flow of real time location tracking in Track real time vehicle location module are as follows:

1. When the driver selects start tracking and if it is the first time for the driver to use the driver's application, the application asks permission to access driver's location.
2. After permission is being granted, the driver's application watches for the driver's location changes using Geolocation.watchPosition. The application requests GPS position of the driver's location every 10 meters moved.
3. The driver's application sends real time location updates to the server through sockets when the location changes. The server then broadcasts the information to students and staffs' application.
4. Students and staffs' application listens to the location updates event and updates the marker position that represents the driver's location accordingly.

5. The driver’s application then compares the distance between current position’s coordinates and the next stop’s coordinates. If the distance is less than 20 meters, the stop’s status will be changed to “passed”. 20 meters is being chosen as it provides the most accuracy compared with other distances tested. The application then sends stop status updates and increments the index to the next stop. The application continues to watch location changes and repeats the step 2 to 5.
6. Meanwhile, the driver can send message updates to students and staffs. This will be further explained in section 6.3.
7. When the driver’s select stop tracking, the application stops observing driver’s location changes and close the socket connection.

6.3 Real Time Message Updates

While travelling, the driver can choose to send message updates such as “On Schedule”, “Traffic Jam” and “Vehicle Breakdown” to the students and staffs according to current situation. The default message will be “On Schedule”. By having this feature, students and staffs will have better knowledge on real time vehicle status. The flow of real time message updates in Display real time vehicle information module are as follows:

1. After the driver selected start tracking, the driver’s application will show a screen with several options of message as mentioned above for the driver to send message updates to inform students and staffs on current situation.
2. After the driver selected an option, the message will be sent to the server through socket. The server then broadcasts the information to students and staffs’ application.
3. Students and staffs’ application listens to the message updates event and update the message accordingly when receives new updates.

6.4 Estimated Time of Arrival Information

Estimated time of arrival (ETA) information is important in vehicle tracking applications for the users to have an idea on how long they need to wait before the vehicle arrives. For UTAR Vehicles Tracking App, the ETA information was implemented in Display real time vehicle information module for the mobile application. The ETA information was retrieved with the help of Google Directions API. The flow of getting ETA information in Display real time vehicle information module are as follows:

1. Student and staffs' application will request for stop status from the driver's application through socket. The application then stores the initial stop status and listens for event of updating stop status.
2. After a stop is being selected, the driver's current position coordinates will be used as the origin. The selected stop location coordinates will be used as the destination.
3. The application then checks the status of each stops between the origin and the destination. If the status is "otw", the stop will be added as waypoints.
4. The application sends request to the Directions API to get the duration to travel from the origin to the destination along all waypoints. The duration returned from the API response will be converted to minutes and displayed as the ETA.
5. The application will send request to the API every minute with updated driver's position and waypoints. The ETA displayed for the selected stop will also be refreshed every minute.

6.5 Socket.IO

Socket.IO was used for sending location and message updates from the driver's side to the students and staffs for Track real time vehicle location module. It consists of a Node.js server and Node.js client. The socket clients will try to establish connection to the socket using WebSocket connection. If it is not possible at that time, the connection will be established using HTTP long polling.

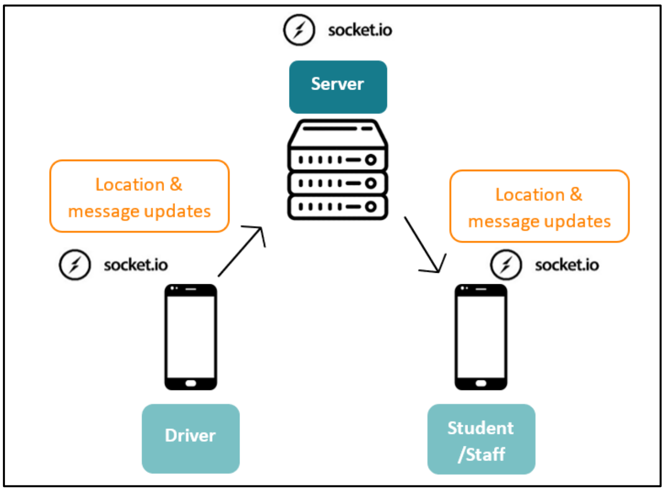


Figure 6.1: Socket.IO Implementation

When the driver starts the location tracking, the application will create a socket instance at the client side and connect to the server instance at the server side. Once connected, the connected client can emit events to and listen for events from the other clients and server.

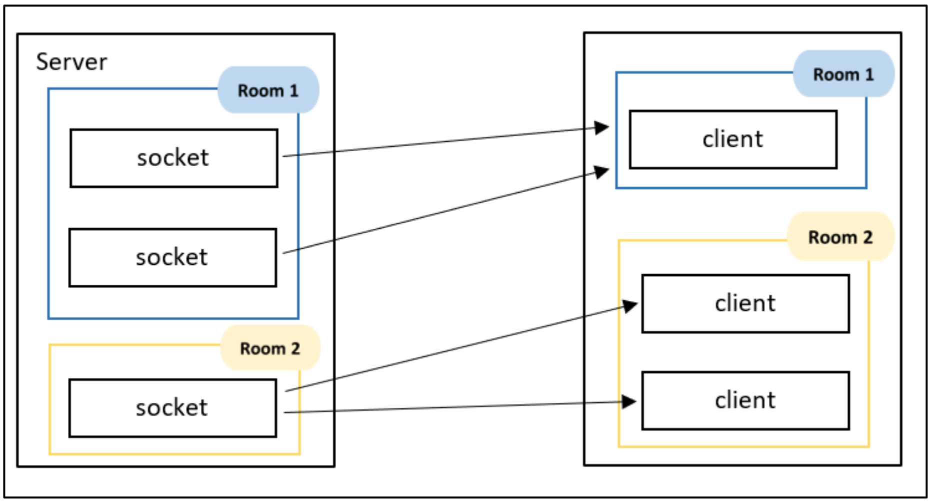
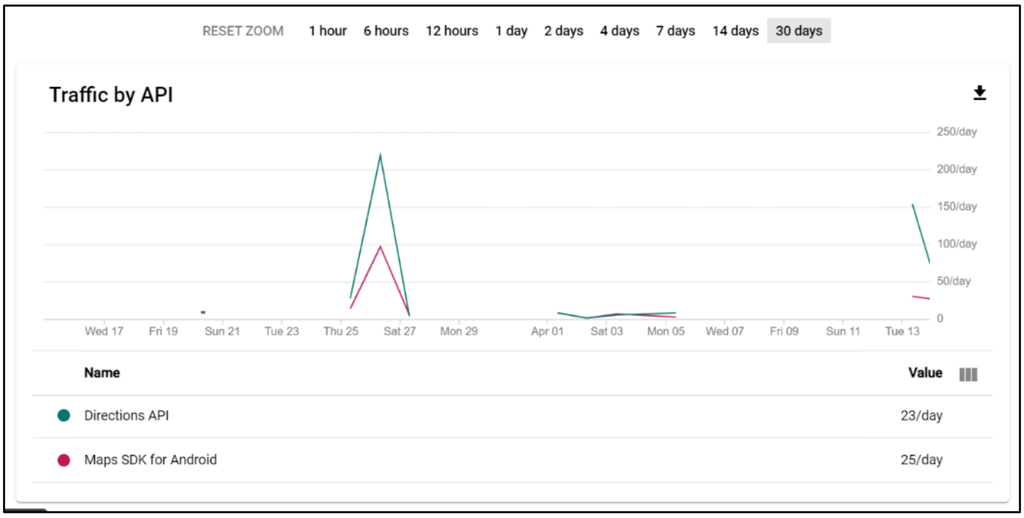


Figure 6.2: Room Concept of Socket Implementation

The socket implementation for UTAR Vehicles Tracking App used the concept of rooms. Each route was considered as different rooms. For example, when a driver starts location tracking for KTM Serdang route with route id 2 in the driver's mobile application, the socket client instance will be added to a room named "room-route id". Hence, the driver's socket instance will join room-2. If students or staffs select the same route for tracking in their mobile application, their socket client instance will be added to the same room as the driver's. When the driver's socket instance emits events such as location update event, the server will only broadcast the location update to clients in the same room. This means that only clients in room 2 can receive the updates. Hence, this allows location tracking for each route remain separated and do not conflict with each other although there are multiple event communication happening concurrently.

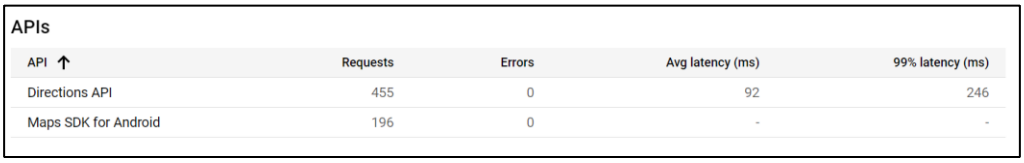
6.6 Google Maps Platform

Google Maps Platform consists of numerous API such as Maps, Routes and Places. The APIs used for the application are Map SDK for Android under Maps API category and Directions API under Routes category.



Name	Value
● Directions API	23/day
● Maps SDK for Android	25/day

Figure 6.3: Traffic by API



APIs

API ↑	Requests	Errors	Avg latency (ms)	99% latency (ms)
Directions API	455	0	92	246
Maps SDK for Android	196	0	-	-

Figure 6.4: Google Maps Platform Enabled APIs

Maps SDK for Android allows developers to include Google Map into the application. It was used in UTAR Vehicles Tracking App to display the Google Map for user to locate the bus stops and view the route map. Markers were added on the map to represent the bus stop location. The route was drawn using the Polyline functionality. Figure 6.5 shows an example of Google Map displayed on the mobile application using Map SDK for Android.

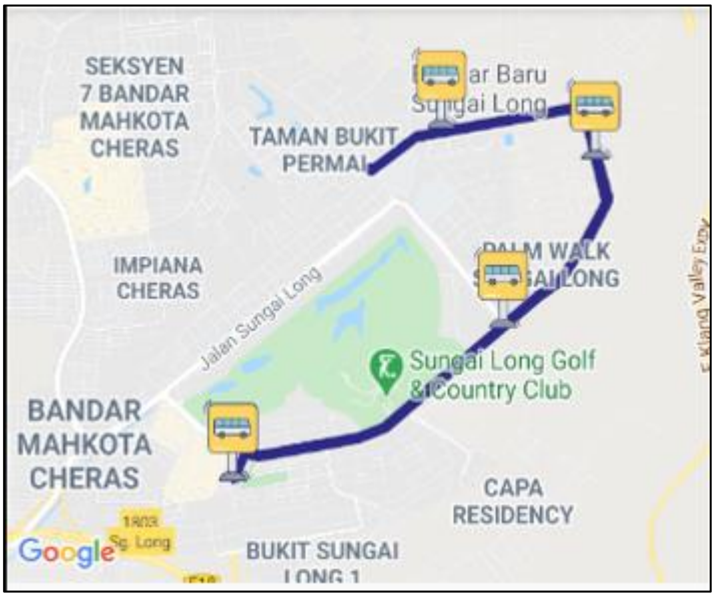


Figure 6.5: Example of Google Map Displayed

Google Directions API is a web service that return the directions from an origin to a destination through HTTP request. Waypoints along the route can also be added to the request. The Directions API was used in Display route information module and Track real time vehicle location module to get the directions to travel from the origin through all bus stops along the route as waypoints and finally the destination of the route. Besides, it was used to retrieve the estimated remaining time to arrival for user selected bus stops. The calculation of remaining time takes in to account the traffic conditions.

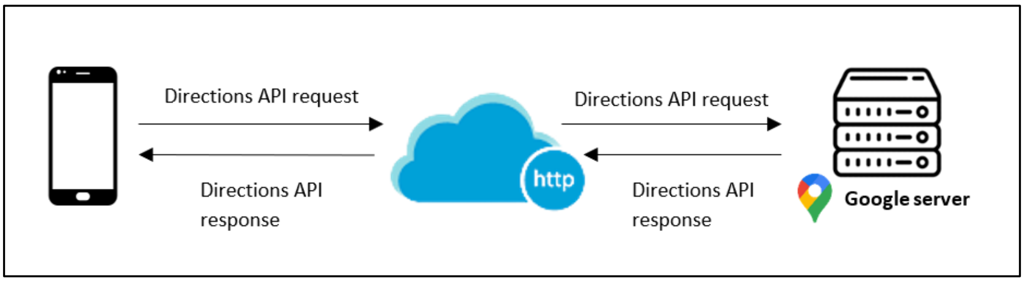
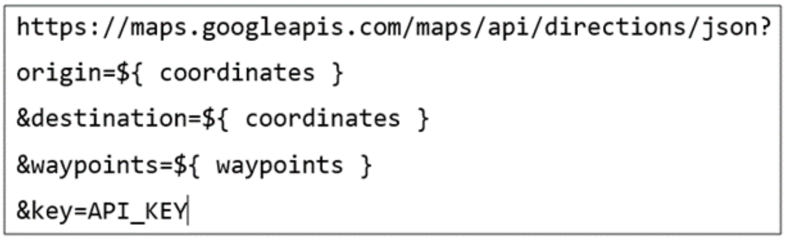


Figure 6.6: Google Directions API Request and Response



```
https://maps.googleapis.com/maps/api/directions/json?
origin=${ coordinates }
&destination=${ coordinates }
&waypoints=${ waypoints }
&key=API_KEY
```

Figure 6.7: Example of Directions API Request

Figure 6.7 shows an example of Directions API request. The result will be returned in JSON format. The result returned will include an encoded polyline specifying the latitude and longitude coordinates of locations along the route. It will then be decoded and used to draw the route to be displayed on the map. For Track real time vehicle location module, the request parameter will need to include an additional parameter which is “&departure_time=now” to get the duration in traffic of the travel for estimated remaining time to arrival.

6.7 Route Endpoints

Different routes were created for different actions to access the database for storing or retrieving information through HTTP request. The development of UTAR Vehicles Tracking App was using local server. Hence, the prefix of each route is “http://server_ip_address:port_number”. Methods used in the applications were GET and POST method.

6.7.1 UTAR Student and Staff Application

Table 6.1: GET Route Endpoints - UTAR Student and Staff Application

Method: GET	
Route	**Description**
/routes	Get all routes
/routeInfo/{routeId}	Get stops information and stop sequence of the route
/schedules/{routeId}/{day}	Get scheduled arrival time for each stop of the route on the day
/scheduleInfo/{scheduleId}	Get schedule information such as driver and vehicle assigned to the schedule
/checkOperatingTrip	Get currently operating routes and schedule

Table 6.2: POST Route Endpoints - UTAR Student and Staff Application

Method: POST	
Route	**Description**
/user/login	Login user

6.7.2 UTAR Driver Application

Table 6.3: GET Route Endpoints - UTAR Driver Application

Method: GET	
Route	**Description**
/driverId	Get driver id of logged in driver
/driverSchedule/{driverId}/{day}	Get schedules assigned to the driver on the day
/schedules/{routeId}/{day}	Get scheduled arrival time for each stop of the route on the day
/scheduleInfo/{scheduleId}	Get schedule information such as driver and vehicle assigned to the selected schedule
/checkOperatingTrip	Get currently operating routes and schedule

Table 6.4: POST Route Endpoints - UTAR Driver Application

Method: POST	
Route	**Description**
/user/login	Login user
/record/{scheduleId}	Add new travel record of the schedule
/updateRecord/{scheduleId}	Update travel record of the schedule

6.7.3 UTAR Administrative Staff Application

Table 6.5: GET Route Endpoints - UTAR Administrative Staff Application

Method: GET	
Route	**Description**
/travelRecord	Get all travel records

/drivers	Get all drivers
/deleteDriver/{id}	Delete the driver’s information
/routes	Get all routes
/routeSchedule/{routeId}	Get all schedules of the route
/addSchedule/{routeId}	Get all stops of the route
/viewSchedule/{routeId}/{scheduleId}	Get the selected schedule information of the route
/deleteSchedule/{routeId}/{scheduleId}	Delete the selected schedule information of the route
logout	Logout user

Table 6.6: POST Route Endpoints - UTAR Administrative Staff Application

Method: POST	
Route	**Description**
/login	Login user
/addDriver	Add new driver
/editDriver	Update driver information
/addSchedule/{routeId}	Add new schedule of the route
/editSchedule	Update schedule information

6.8 Conclusion

The features planned to be included in UTAR Vehicles Tracking App were fully developed in implementation phase of the project. Implementation details and application flow of key features such as real time location tracking, real time message updates and estimated time of arrival and tools used such as Socket.IO and Google Maps Platform were described in this chapter.

CHAPTER 7

TESTING AND EVALUATION

7.1 Introduction

This chapter describes testing activities of the project to ensure all functions were implemented and work correctly. Section 7.2 describes the types of testing conducted, section 7.3 describes unit testing, section 7.4 describes integration testing, section 7.5 describes user acceptance testing and section 7.6 describes Lighthouse application audit.

7.2 Types of Testing

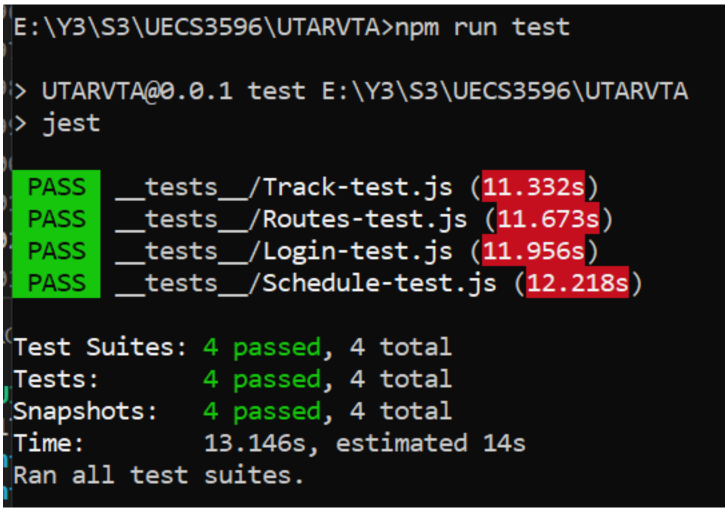
The types of testing conducted for UTAR Vehicles Tracking App were unit testing, integration testing and user acceptance testing. Unit testing tests individual components or modules to ensure each of them works as intended. Integration testing tests whether the software components interact and work correctly in combination. User acceptance testing test whether the users' requirements are being fulfilled and users are satisfied with the software developed.

Unit testing for UTAR Vehicles Tracking App focused on testing whether the screens are being rendered correctly by isolating its interactions with other screens and the database. Unit testing for driver's and students and staffs' mobile application were conducted using Jest. Jest is a testing framework for testing JavaScript applications. Unit testing for API routes endpoint used by the mobile applications were also conducted using Postman.

Integration testing for the applications developed focused on testing the application's interactions with database or third-party services such as Google Maps Platform API and navigations between different screens. Different test cases were designed to ensure that all functions of the applications work correctly. User acceptance testing tested user's satisfaction on the developed application and mainly focusing on usability aspects of the application.

7.3 Unit Testing

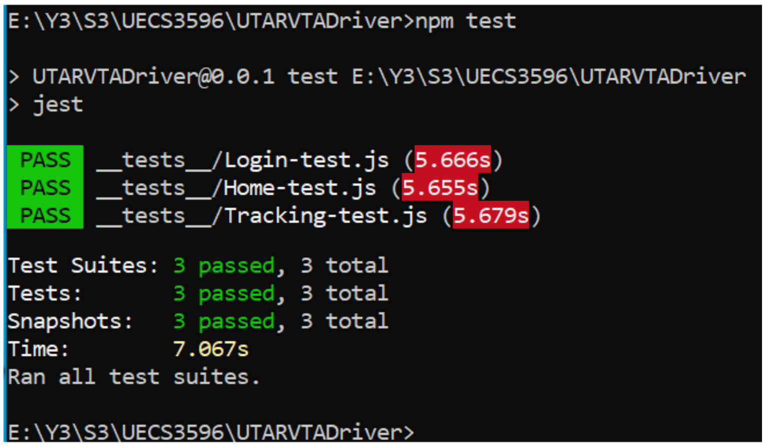
7.3.1 UTAR Students and Staffs Application



```
E:\Y3\S3\UECS3596\UTARVTA>npm run test

> UTARVTA@0.0.1 test E:\Y3\S3\UECS3596\UTARVTA
> jest

 PASS  __tests__/Track-test.js (11.332s)
 PASS  __tests__/Routes-test.js (11.673s)
 PASS  __tests__/Login-test.js (11.956s)
 PASS  __tests__/Schedule-test.js (12.218s)

Test Suites: 4 passed, 4 total
Tests:       4 passed, 4 total
Snapshots:   4 passed, 4 total
Time:        13.146s, estimated 14s
Ran all test suites.
```

Figure 7.1: Unit Test Results - UTAR Students and Staffs Application

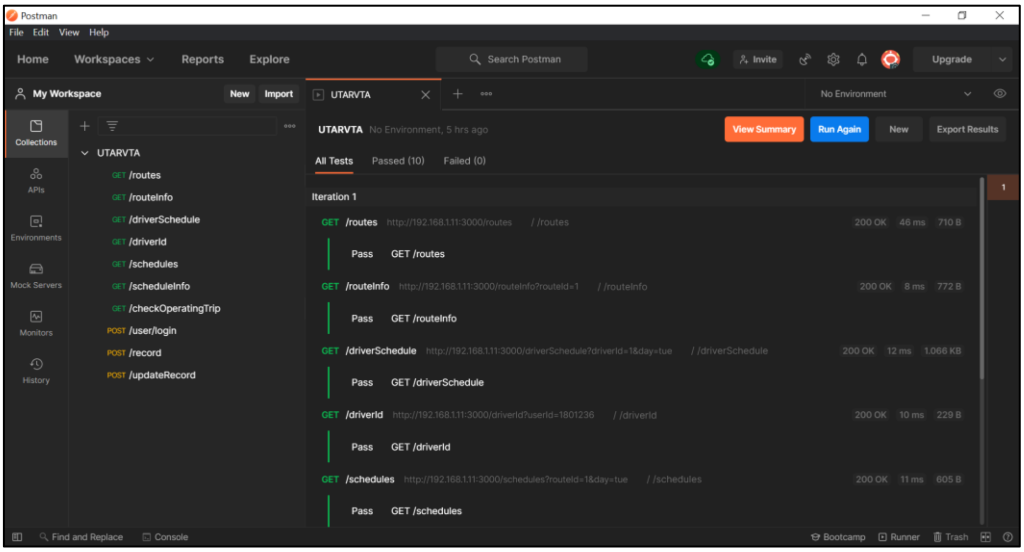
7.3.2 UTAR Driver Application



```
E:\Y3\S3\UECS3596\UTARVTADriver>npm test

> UTARVTADriver@0.0.1 test E:\Y3\S3\UECS3596\UTARVTADriver
> jest

 PASS  __tests__/Login-test.js (5.666s)
 PASS  __tests__/Home-test.js (5.655s)
 PASS  __tests__/Tracking-test.js (5.679s)

Test Suites: 3 passed, 3 total
Tests:       3 passed, 3 total
Snapshots:   3 passed, 3 total
Time:        7.067s
Ran all test suites.

E:\Y3\S3\UECS3596\UTARVTADriver>
```

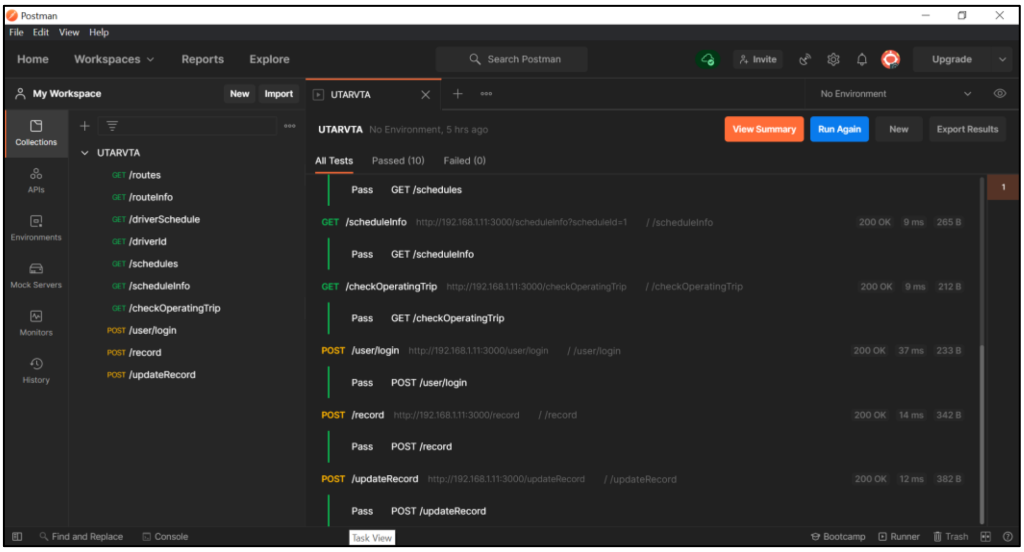
Figure 7.2: Unit Test Result - UTAR Driver Application

7.3.3 API Route Endpoints



```
UTARVTA No Environment, 5 hrs ago

Iteration 1
GET /routes  http://192.168.1.11:3000/routes  / /routes  200 OK  46 ms  710 B
    Pass    GET /routes
GET /routeInfo  http://192.168.1.11:3000/routeInfo?routeId=1  / /routeInfo  200 OK  8 ms  772 B
    Pass    GET /routeInfo
GET /driverSchedule  http://192.168.1.11:3000/driverSchedule?driverId=1&day=tue  / /driverSchedule  200 OK  12 ms  1.066 KB
    Pass    GET /driverSchedule
GET /driverId  http://192.168.1.11:3000/driverId?userId=1801236  / /driverId  200 OK  10 ms  229 B
    Pass    GET /driverId
GET /schedules  http://192.168.1.11:3000/schedules?routeId=1&day=tue  / /schedules  200 OK  11 ms  605 B
    Pass    GET /schedules
```

Figure 7.3: API Test Results 1



```
UTARVTA No Environment, 5 hrs ago

    Pass    GET /schedules
GET /scheduleInfo  http://192.168.1.11:3000/scheduleInfo?scheduleId=1  / /scheduleInfo  200 OK  9 ms  265 B
    Pass    GET /scheduleInfo
GET /checkOperatingTrip  http://192.168.1.11:3000/checkOperatingTrip  / /checkOperatingTrip  200 OK  9 ms  212 B
    Pass    GET /checkOperatingTrip
POST /user/login  http://192.168.1.11:3000/user/login  / /user/login  200 OK  37 ms  233 B
    Pass    POST /user/login
POST /record  http://192.168.1.11:3000/record  / /record  200 OK  14 ms  342 B
    Pass    POST /record
POST /updateRecord  http://192.168.1.11:3000/updateRecord  / /updateRecord  200 OK  12 ms  382 B
    Pass    POST /updateRecord
```

Figure 7.4: API Test Results 2

7.4 Integration Testing

Test cases and results for integration testing were included in the APPENDIX C: Integration Testing Test Cases.

7.5 User Acceptance Testing

As the main focus for UTAR Vehicles Tracking App project was on students and staffs' application, user acceptance testing was conducted for the application. 5 students were involved in testing the students and staffs' application. Due to the Covid-19 pandemic which resulted in harder reach for face-to-face testing with the users, the user acceptance testing was conducted online through Microsoft Teams by sharing screen to the users and allowing them to control screen to interact with the application. Users were required to fill in a survey form after testing the application. The survey form used the System Usability Scale template by John Brooke (usability.gov, 2013). Feedbacks gathered from the survey involve positive feedbacks such as the application was easy to use, user friendly and can be learned to use quickly. Some users responded that the estimated time of arrival was accurate. There were also negative feedbacks for improvements such as the map displayed does not cover the entire route hence users need to zoom in or out to view the stops or driver's location on the route. The survey form and responses were included in the APPENDIX D: User Acceptance Testing Survey Form and Responses.

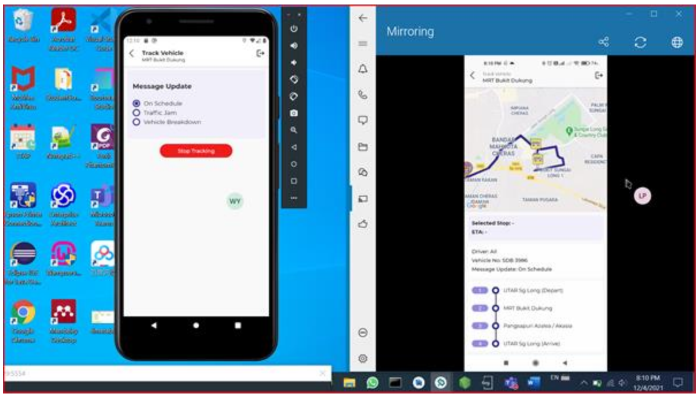


Figure 7.5: Online User Acceptance Testing

7.6 Lighthouse Application Audit

Lighthouse is an open-source web application audit tool developed by Google to access the non-functional attributes of the web application such as performance, accessibility and best practices. It was used to audit UTAR Vehicles Tracking App administrative staff's web application. The complete report was included in the APPENDIX E: Lighthouse Application Audit Report.

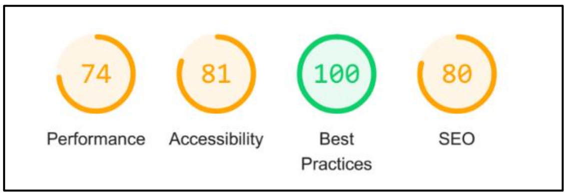


Figure 7.6: Lighthouse Audit Result

7.7 Conclusion

Unit testing and integration testing was done for students and staffs’ application, drivers’ application and transport section application to test whether the features developed work as intended. An application audit was conducted on the transport section application. User acceptance testing was also conducted for students and staffs’ application. All test results were recorded. All tests for the applications were passed at the end of testing phase.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

This chapter summarises the UTAR Vehicles Tracking App project, the limitations and recommendations for future work.

This project aimed to provide a platform for UTAR students and staffs to track UTAR vehicles, obtain latest routes and schedules information. Besides, this project provided a platform for UTAR transport section administrative staffs to manage schedule information, driver information and monitor travel records more efficiently and effectively.

At the end of this project, the project objectives planned at the beginning were achieved and UTAR Vehicles Tracking App was successfully developed. 3 applications which includes a mobile application for UTAR drivers, a mobile application for UTAR students and staffs, and a UTAR transport section administrative staff web application was developed in the UTAR Vehicles Tracking App project. The project objectives initially planned that were achieved include:

i. To analyse the requirements through reviewing existing applications.

ii. To develop a vehicle tracking application for UTAR students and staffs and UTAR transport section.

iii. To evaluate the application functionalities through unit testing, integration testing and user acceptance testing.

8.2 Limitations

There are several limitations faced during the implementation of this project. The limitations faced are as follows:

i. Several functions such as plan journey and monitor driver’s actual route travelled were not included due to time constraints.

ii. Track real time vehicle location function was tested through simulation due to author’s unavailability in Sungai Long area during Covid-19 CMCO period.

8.3 Recommendations for Future Work

Future works to improve UTAR Vehicles Tracking App were planned and considered. Future works include:

i. Plan journey feature for students and staffs to search for route to be taken and duration needed to reach their destination.

ii. Notification feature to allow students and staffs to get informed once there are changes in transport services.

iii. Monitor driver’s travel to allow UTAR transport section to monitor whether the driver travels according to the route planned.

iv. Track real time vehicle location function for the UTAR transport section web application to monitor operating trip on real time.

REFERENCES

Anand, R.V. and Dinakaran, M., 2016. Popular Agile Methods in Software Development: Review and Analysis. *International Journal of Scientific and Technical Advancements*, 2(4), pp.147–150.

Chit, S.M., Chaw, L.Y., Thong, C.L. and Lee, C.Y., 2017. A pilot study: Shuttle bus tracker app for campus users. *2017 International Conference on Research and Innovation in Information Systems (ICRIIS)*, [online] pp.1–6. Available through: Universiti Tunku Abdul Rahman Library website <http://library.utar.edu.my> [Accessed 4 July 2020].

Express, n.d. [online] Express. Available at: <https://expressjs.com/> [Accessed 13 February 2021].

Ghahrai, A., 2016. *Software Development Methodologies*. [online] DevQA. Available at: <https://devqa.io/software-development-methodologies/> [Accessed 22 August 2020].

Google Developers, n.d. *Meet Android Studio*. [online] Google Developers. Available at: <https://developer.android.com/studio/intro> [Accessed 2 September 2020].

Guru99, n.d. *Agile Methodology & Model: Guide for Software Development & Testing*. [online] Guru99. Available at: <https://www.guru99.com/agile-scrum-extreme-testing.html> [Accessed 17 August 2020].

Guru99, n.d. *Prototyping Model in Software Engineering: Methodology, Process, Approach*. [online] Guru99. Available at: <https://www.guru99.com/software-engineering-prototyping-model.html> [Accessed 22 August 2020].

Lunke, E.B., 2020. Commuters’ satisfaction with public transport. *Journal of Transport and Health*, [online] Available at: <https://www.researchgate.net/publication/340070970_Commuters’_satisfaction_with_public_transport> [Accessed 4 July 2020].

Malik, S. and Nigam, C., 2017. A Comparative study of Different types of Models in Software Development Life Cycle. *International Research Journal of Engineering and Technology*, [online] 04(10). Available at: <www.irjet.net> [Accessed 16 August 2020].

Node.js, n.d. [online] Node.js. Available at: <https://nodejs.org/en/> [Accessed 13 February 2021].

Octoverse, 2018. *Projects*. [online] The State of the Octoverse. Available at: <https://octoverse.github.com/2018/projects.html> [Accessed 18 August 2020].

Ortiz, C.E., 2018. *Dissecting React Native*. [online] IBM Developer. Available at: <https://developer.ibm.com/articles/dissecting-react-native/> [Accessed 13 February 2021].

Pal, S.K., 2018. *Software Engineering: Phases of Prototyping Model: Set - 2*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/software-engineering-phases-prototyping-model-set-2/> [Accessed 22 August 2020].

Pluralsight, 2020. *Pluralsight Technology Index*. [online] Pluralsight. Available at: <https://www.pluralsight.com/tech-index> [Accessed 22 August 2020].

Pressman, R.S. and Maxim, B.R., 2015. *Software Engineering : A PRACTITIONER'S APPROACH.* 8th ed. McGraw-Hill Education.

Rahman, M.M., Wirasinghe, S.C. and Kattan, L., 2013. Users' views on current and future real-time bus information systems. [online] Available through: Universiti Tunku Abdul Rahman Library website <http://library.utar.edu.my> [Accessed 10 July 2020].

Saeed, S., Jhanjhi, N.Z., Naqvi, M. and Humayun, M., 2019. Analysis of software development methodologies. *International Journal of Computing and Digital Systems,* 8(5), pp.445–460. Available through: Universiti Tunku Abdul Rahman Library website <http://library.utar.edu.my> [Accessed 16 August 2020].

Sharif, S.A., Suhaimi, M.S., Jamal, N.N., Riadz, I.K., Amran, I.F. and Jawawi, D.N.A., 2018. Real-Time Campus University Bus Tracking Mobile Application. *2018 Seventh ICT International Student Project Conference (ICT-ISPC)*, [online] pp.1–6. Available through: Universiti Tunku Abdul Rahman Library website <http://library.utar.edu.my> [Accessed 4 July 2020].

Shrivastava, S. and Srivastava, A.K., 2020. *Agile Methodology: A Beginner's Guide To Agile Method and Scrum.* [online] Software Testing Help. Available at: <https://www.softwaretestinghelp.com/agile-scrum-methodology-for-development-and-testing/> [Accessed 17 August 2020].

Singh, A., 2020. *Flutter Is The Future Of Mobile App Development-Know Why?* [online] Available at: <https://medium.com/flutter-community/why-the-flutter-is-the-future-of-mobile-development-9f6e5657a61d> [Accessed 28 August 2020].

Skuza, B., Włodarczyk, D. and Mroczkowska, A., 2019. *Flutter vs React Native – what to choose in 2020?* [online] Droids On Roids. Available at: <https://www.thedroidsonroids.com/blog/flutter-vs-react-native-what-to-choose-in-2020> [Accessed 18 August 2020].

Socket.IO, n.d. *Socket.IO*. [online] Available at: <https://socket.io/> [Accessed 13 February 2021].

Software Testing Help, 2020. *What is SDLC Waterfall Model?* [online] Software Testing Help. Available at: <https://www.softwaretestinghelp.com/what-is-sdlc- waterfall-model/> [Accessed 22 August 2020].

Sommerville, I., 2011. *Software engineering*. 9th ed. Boston: Pearson.

Stevenson, D., 2018. *What is Firebase? The complete story, abridged.* [online] Medium. Available at: <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0> [Accessed 2 September 2020].

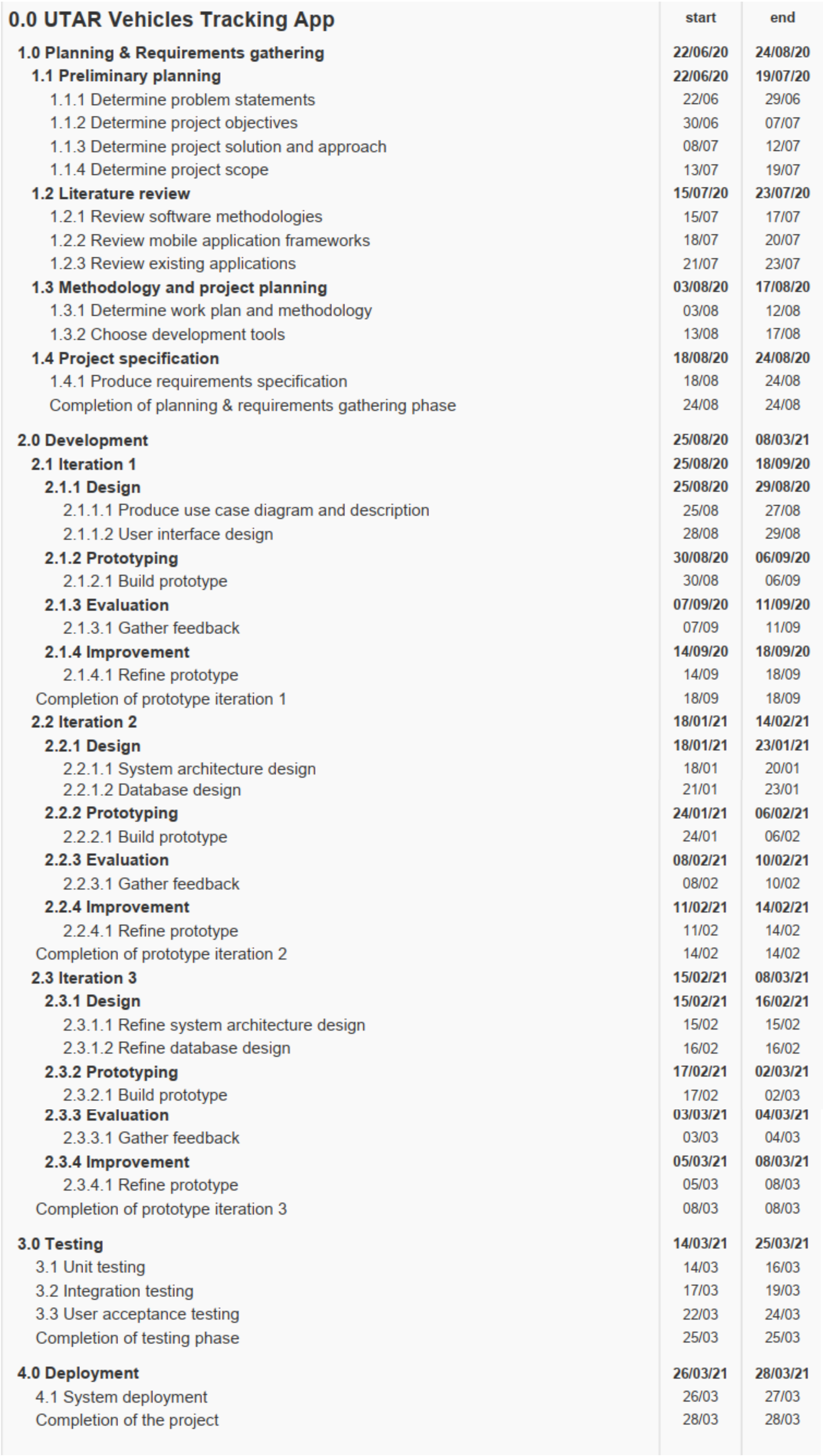
Tutorialspoint, n.d. *SDLC - Overview*. [online] Tutorialspoint. Available at: <https://www.tutorialspoint.com/sdlc/sdlc_overview.htm> [Accessed 22 August 2020].

usability.gov, 2013. *System Usability Scale (SUS)*. [online] Available at: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> [Accessed 15 April 2021].

Watkins, K.E., Ferris, B., Borning, A., Rutherford, G.S. and Layton, D., 2011. Where Is My Bus? Impact of mobile real-time information on the perceived and actual wait time of transit riders. *Transportation Research Part A: Policy and Practice*, [online] 45(8), pp.839–848. Available through: Universiti Tunku Abdul Rahman Library website <http://library.utar.edu.my> [Accessed 4 July 2020].

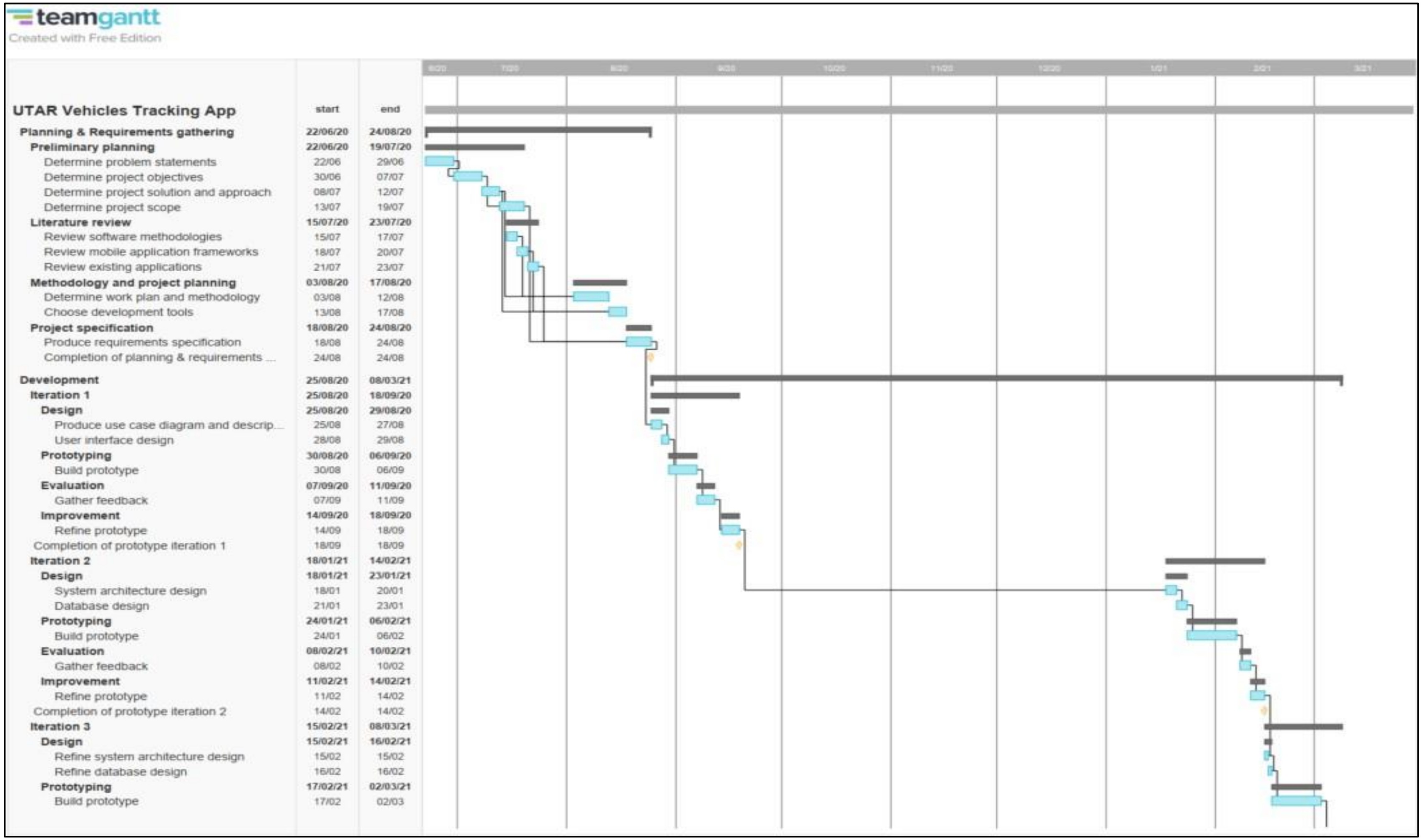
APPENDICES

APPENDIX A: Work Breakdown Structure

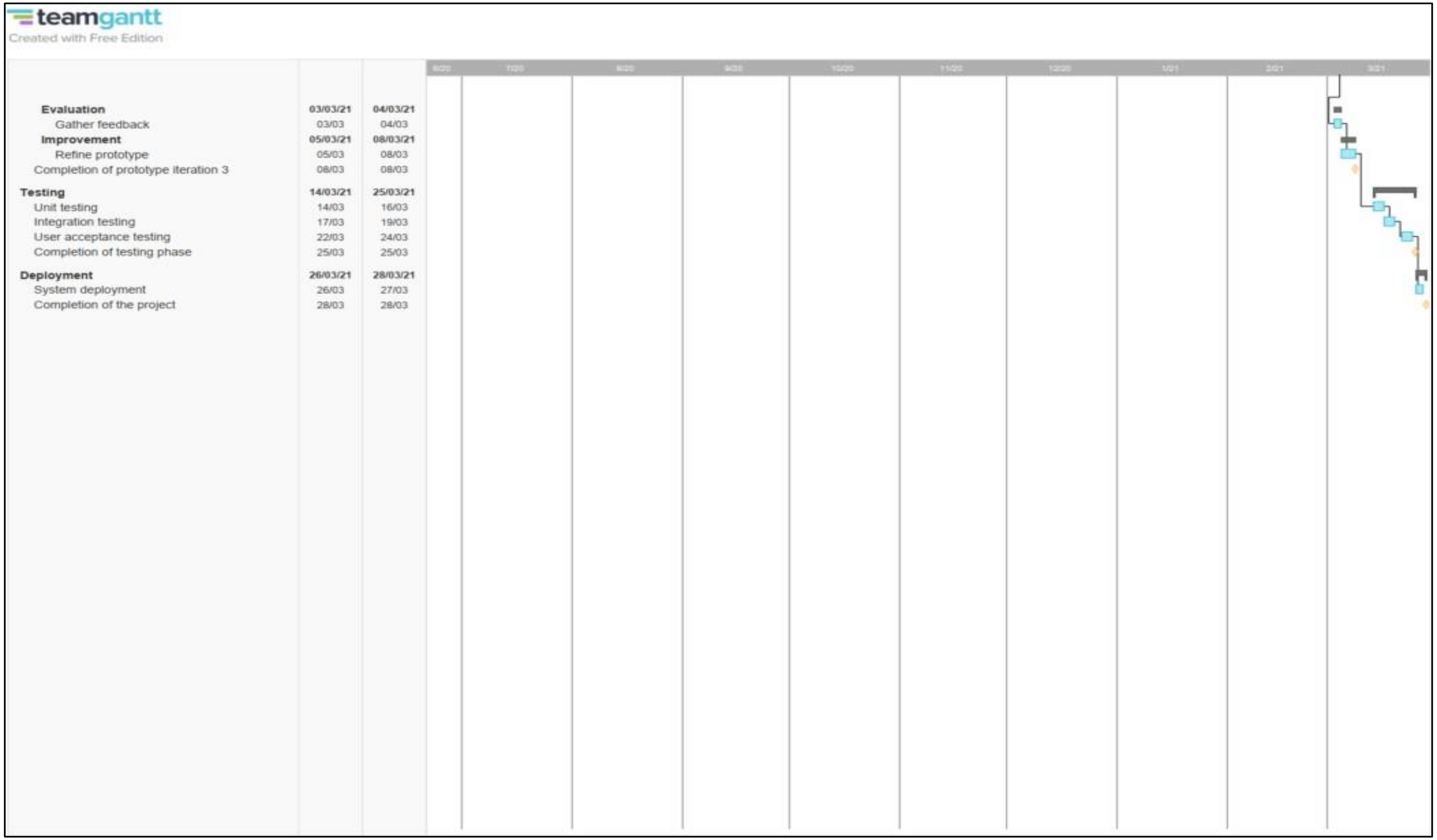


0.0 UTAR Vehicles Tracking App	start	end
1.0 Planning & Requirements gathering	**22/06/20**	**24/08/20**
1.1 Preliminary planning	**22/06/20**	**19/07/20**
1.1.1 Determine problem statements	22/06	29/06
1.1.2 Determine project objectives	30/06	07/07
1.1.3 Determine project solution and approach	08/07	12/07
1.1.4 Determine project scope	13/07	19/07
1.2 Literature review	**15/07/20**	**23/07/20**
1.2.1 Review software methodologies	15/07	17/07
1.2.2 Review mobile application frameworks	18/07	20/07
1.2.3 Review existing applications	21/07	23/07
1.3 Methodology and project planning	**03/08/20**	**17/08/20**
1.3.1 Determine work plan and methodology	03/08	12/08
1.3.2 Choose development tools	13/08	17/08
1.4 Project specification	**18/08/20**	**24/08/20**
1.4.1 Produce requirements specification	18/08	24/08
Completion of planning & requirements gathering phase	24/08	24/08
2.0 Development	**25/08/20**	**08/03/21**
2.1 Iteration 1	**25/08/20**	**18/09/20**
2.1.1 Design	**25/08/20**	**29/08/20**
2.1.1.1 Produce use case diagram and description	25/08	27/08
2.1.1.2 User interface design	28/08	29/08
2.1.2 Prototyping	**30/08/20**	**06/09/20**
2.1.2.1 Build prototype	30/08	06/09
2.1.3 Evaluation	**07/09/20**	**11/09/20**
2.1.3.1 Gather feedback	07/09	11/09
2.1.4 Improvement	**14/09/20**	**18/09/20**
2.1.4.1 Refine prototype	14/09	18/09
Completion of prototype iteration 1	18/09	18/09
2.2 Iteration 2	**18/01/21**	**14/02/21**
2.2.1 Design	**18/01/21**	**23/01/21**
2.2.1.1 System architecture design	18/01	20/01
2.2.1.2 Database design	21/01	23/01
2.2.2 Prototyping	**24/01/21**	**06/02/21**
2.2.2.1 Build prototype	24/01	06/02
2.2.3 Evaluation	**08/02/21**	**10/02/21**
2.2.3.1 Gather feedback	08/02	10/02
2.2.4 Improvement	**11/02/21**	**14/02/21**
2.2.4.1 Refine prototype	11/02	14/02
Completion of prototype iteration 2	14/02	14/02
2.3 Iteration 3	**15/02/21**	**08/03/21**
2.3.1 Design	**15/02/21**	**16/02/21**
2.3.1.1 Refine system architecture design	15/02	15/02
2.3.1.2 Refine database design	16/02	16/02
2.3.2 Prototyping	**17/02/21**	**02/03/21**
2.3.2.1 Build prototype	17/02	02/03
2.3.3 Evaluation	**03/03/21**	**04/03/21**
2.3.3.1 Gather feedback	03/03	04/03
2.3.4 Improvement	**05/03/21**	**08/03/21**
2.3.4.1 Refine prototype	05/03	08/03
Completion of prototype iteration 3	08/03	08/03
3.0 Testing	**14/03/21**	**25/03/21**
3.1 Unit testing	14/03	16/03
3.2 Integration testing	17/03	19/03
3.3 User acceptance testing	22/03	24/03
Completion of testing phase	25/03	25/03
4.0 Deployment	**26/03/21**	**28/03/21**
4.1 System deployment	26/03	27/03
Completion of the project	28/03	28/03

APPENDIX B: Gannt Chart



UTAR Vehicles Tracking App	start	end
Planning & Requirements gathering	**22/06/20**	**24/08/20**
Preliminary planning	**22/06/20**	**19/07/20**
Determine problem statements	22/06	29/06
Determine project objectives	30/06	07/07
Determine project solution and approach	08/07	12/07
Determine project scope	13/07	19/07
Literature review	**15/07/20**	**23/07/20**
Review software methodologies	15/07	17/07
Review mobile application frameworks	18/07	20/07
Review existing applications	21/07	23/07
Methodology and project planning	**03/08/20**	**17/08/20**
Determine work plan and methodology	03/08	12/08
Choose development tools	13/08	17/08
Project specification	**18/08/20**	**24/08/20**
Produce requirements specification	18/08	24/08
Completion of planning & requirements ...	24/08	24/08
Development	**25/08/20**	**08/03/21**
Iteration 1	**25/08/20**	**18/09/20**
Design	**25/08/20**	**29/08/20**
Produce use case diagram and descrip...	25/08	27/08
User interface design	28/08	29/08
Prototyping	**30/08/20**	**06/09/20**
Build prototype	30/08	06/09
Evaluation	**07/09/20**	**11/09/20**
Gather feedback	07/09	11/09
Improvement	**14/09/20**	**18/09/20**
Refine prototype	14/09	18/09
Completion of prototype iteration 1	18/09	18/09
Iteration 2	**18/01/21**	**14/02/21**
Design	**18/01/21**	**23/01/21**
System architecture design	18/01	20/01
Database design	21/01	23/01
Prototyping	**24/01/21**	**06/02/21**
Build prototype	24/01	06/02
Evaluation	**08/02/21**	**10/02/21**
Gather feedback	08/02	10/02
Improvement	**11/02/21**	**14/02/21**
Refine prototype	11/02	14/02
Completion of prototype iteration 2	14/02	14/02
Iteration 3	**15/02/21**	**08/03/21**
Design	**15/02/21**	**16/02/21**
Refine system architecture design	15/02	15/02
Refine database design	16/02	16/02
Prototyping	**17/02/21**	**02/03/21**
Build prototype	17/02	02/03



Evaluation	**03/03/21**	**04/03/21**
Gather feedback	03/03	04/03
Improvement	**05/03/21**	**08/03/21**
Refine prototype	05/03	08/03
Completion of prototype iteration 3	08/03	08/03
Testing	**14/03/21**	**25/03/21**
Unit testing	14/03	16/03
Integration testing	17/03	19/03
User acceptance testing	22/03	24/03
Completion of testing phase	25/03	25/03
Deployment	**26/03/21**	**28/03/21**
System deployment	26/03	27/03
Completion of the project	28/03	28/03

APPENDIX C: Integration Testing Test Cases

UTAR Student and Staff Application						
Test Case	**Test Title**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
1	Login	1. Enter ID and password. 2. Select Login button.	Valid user's credentials	- Routes screen will be shown after successful login.	Same as expected result	Pass
			Invalid user's credentials	- Alert box showing login failed will be shown.	Same as expected result	Pass
2	Track real time vehicle location	1. Select Track Vehicle on bottom tab menu. 2. Select route.	-	- Track Vehicle screen will be shown. - A map will be shown. - Route, stops and driver's location will be displayed on the map.	Same as expected result	Pass
3	Display real time vehicle information	1. Select Track Vehicle on bottom tab menu. 2. Select route. 3. Select a stop.	-	- Track Vehicle screen will be shown. - Driver's name, vehicle plate number, stops on the route will be shown. - Selected stop and ETA will be	Same as expected result	Pass

				shown when a stop is selected.		
4	Display route information	1. Select Routes on bottom tab menu. 2. Select a route.	-	- Routes screen will be shown. - Route Information screen will be shown after a route is selected. - A map will be shown. - Route and stops will be displayed on the map. - Stops' sequence will be shown.	Same as expected result	Pass
5	Display schedule	1. Select Schedule on bottom tab menu. 2. Select route and date. 3. Select Search.	Route name and date	- Schedule screen will be shown. - Time schedule for each stop will be shown.	Same as expected result	Pass

UTAR Driver Application						
Test Case	**Test Title**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
1	Login	1. Enter ID and password. 2. Select Login button.	Valid Driver's credentials	- Schedules screen will be shown. - Schedules assigned to the driver on that day will be shown.	Same as expected result	Pass
			Invalid Driver's credentials	- Alert box showing login failed will be shown.	Same as expected result	Pass
2	Track real time vehicle location	1. Select a schedule. 2. Select Start Tracking button. 3. Select Stop Tracking to stop tracking.	-	- Tracking screen will be shown. - Schedule information will be shown. - Message updates screen will be shown after Start Tracking button is selected. - Redirected to Schedules screen after Stop Tracking is selected.	Same as expected result	Pass

3	Display real time vehicle information	1. Select an option from message updates to send message.	-	- Snackbar showing "Message sent" will be shown after select an option from message updates.	Same as expected result	Pass

UTAR Administrative Staff Application						
Test Case	**Test Title**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
1	Login	1. Enter email and password. 2. Select Login button.	Valid admin's credentials	- Home page will be shown.	Same as expected result	Pass
			Invalid admin's credentials	- Alert message showing credentials does not match record will be shown.	Same as expected result	Pass
2	View travel record	1. Select Travel Record from Home page or side menu.	-	- Travel Record page will be shown.	Same as expected result	Pass
3	View schedule	1. Select Route Schedule from Home page or side menu. 2. Select a route.	-	- Routes page will be shown. - Route and schedule information will be shown after selecting a route.	Same as expected result	Pass

4	Add schedule	1. Select Route Schedule from Home page or side menu. 2. Select a route. 3. Select Add Schedule button. 4. Enter schedule details. 5. Select Save button.	Schedule details	- Modal form will be shown after selecting Add Schedule button. - Schedule information table will be updated with new schedule record.	Same as expected result	Pass
5	Update schedule	1. Select Route Schedule from Home page or side menu. 2. Select a route. 3. Select View button on the schedule record. 4. Select Edit button. 5. Enter schedule details. 6. Select Save button.	Schedule details	- Schedule details of the selected record will be shown after selecting View button. - Modal form will be shown after selecting Edit button. - Schedule information table will be updated with edited schedule record.	Same as expected result	Pass

6	Delete schedule	1. Select Route Schedule from Home page or side menu. 2. Select a route. 3. Select View button on the schedule record. 4. Select Delete button.	Schedule details	- Schedule details of the selected record will be shown after selecting View button. - Schedule information table will be updated. - The deleted record will not be shown in the table.	Same as expected result	Pass
7	View driver	1. Select Drivers from Home page or side menu.	-	- Drivers page will be shown.	Same as expected result	Pass
8	Add driver	1. Select Drivers from Home page or side menu. 2. Select Add Driver button. 3. Enter driver details. 4. Select Save button.	Driver details	- Add Driver page will be shown after selecting Add Driver button. - Drivers information table will be updated with new driver record.	Same as expected result	Pass

9	Update driver	1. Select Drivers from Home page or side menu. 2. Select Edit button. 3. Enter driver details. 4. Select Save button.	Driver details	- Modal form will be shown after selecting Edit button. - Driver information table will be updated with edited driver record.	Same as expected result	Pass
10	Delete driver	1. Select Drivers from Home page or side menu. 2. Select Delete button.	-	- Driver information table will be updated. - The deleted record will not be shown in the table.	Same as expected result	Pass

APPENDIX D: User Acceptance Testing Survey Form and Responses

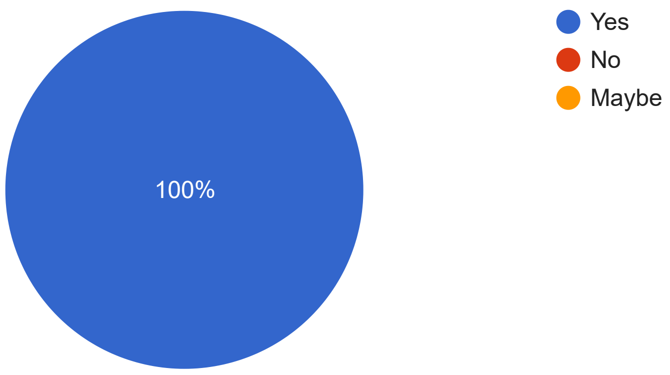
User Satisfaction Survey for UTAR Vehicles Tracking App (Student/Staff version)

5 responses

Publish analytics

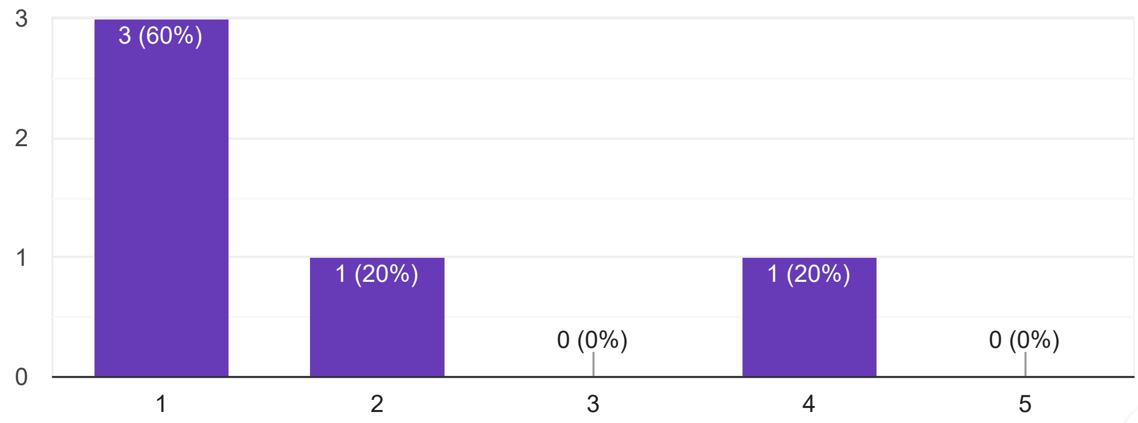
I think I would like to use this application in future while travelling with UTAR vehicles.

5 responses



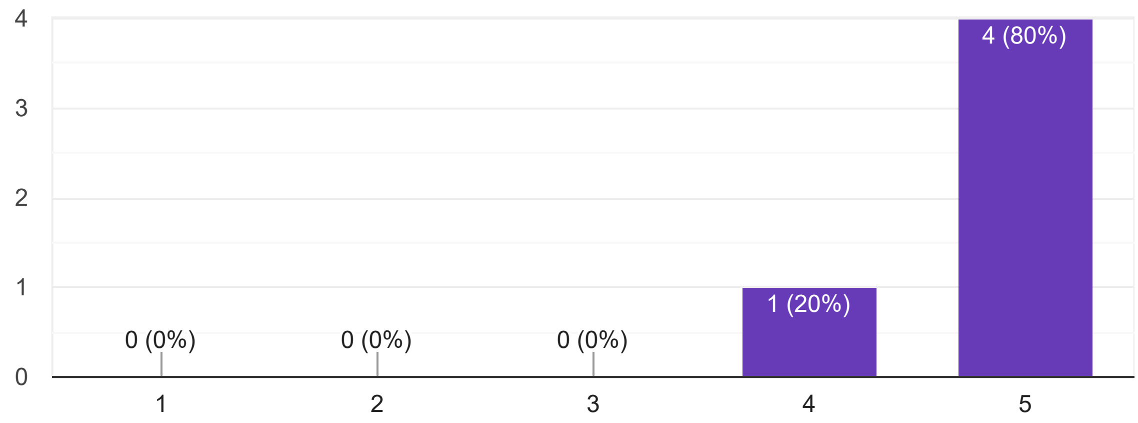
I found the application is too complex.

5 responses



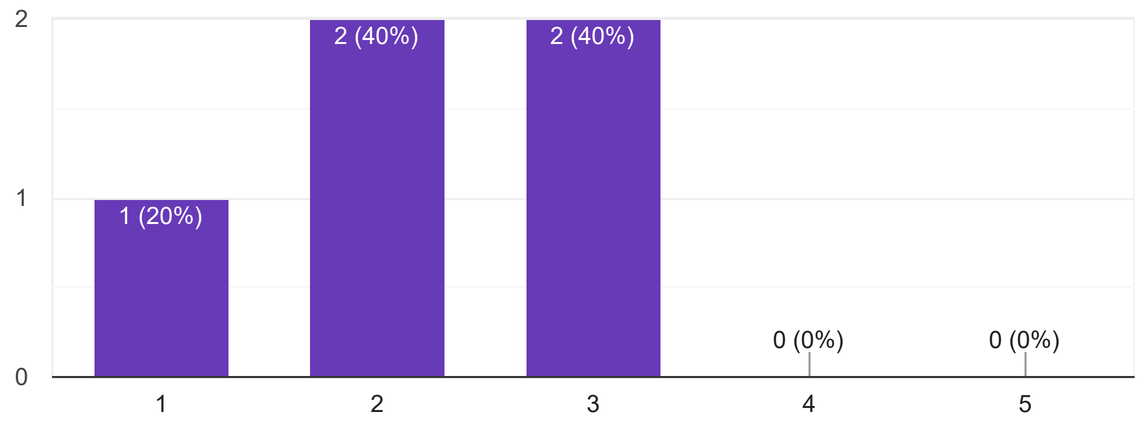
The application was easy to use.

5 responses



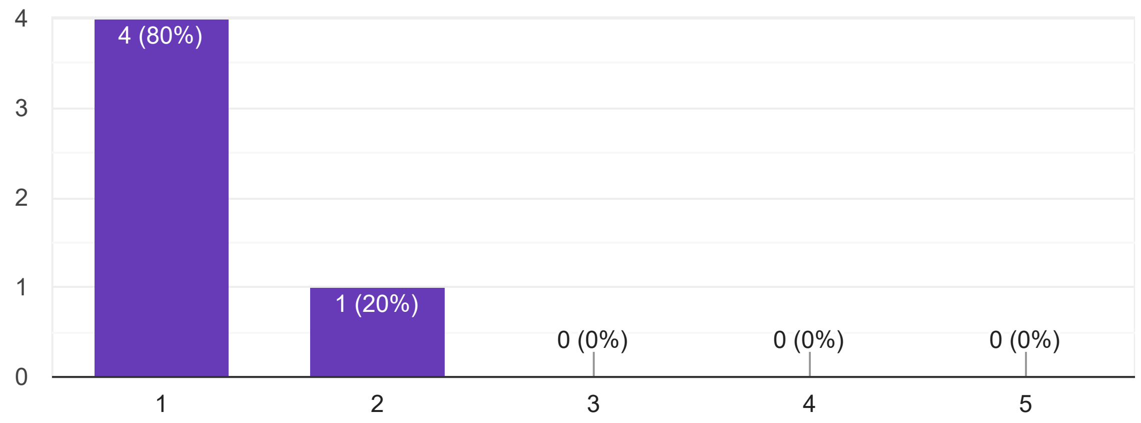
I would need technical support in using this application.

5 responses



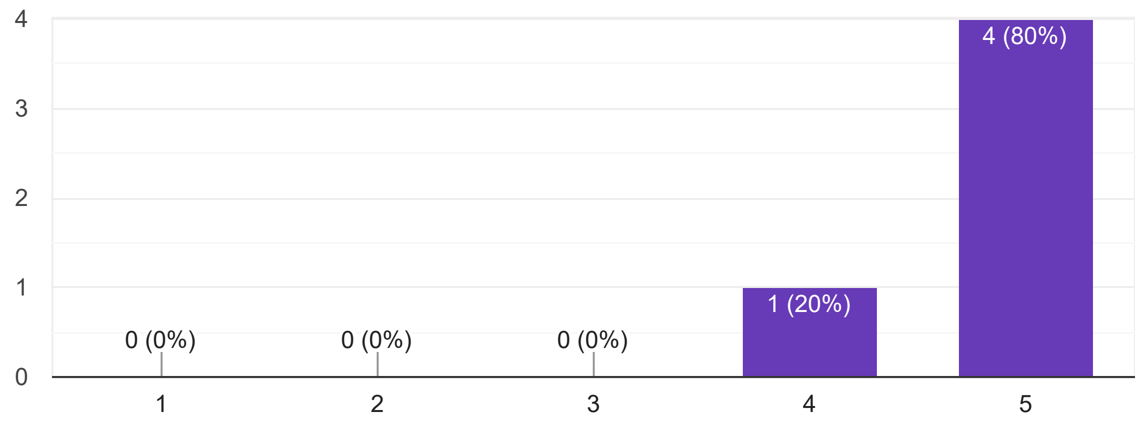
There was too much inconsistency in this application.

5 responses



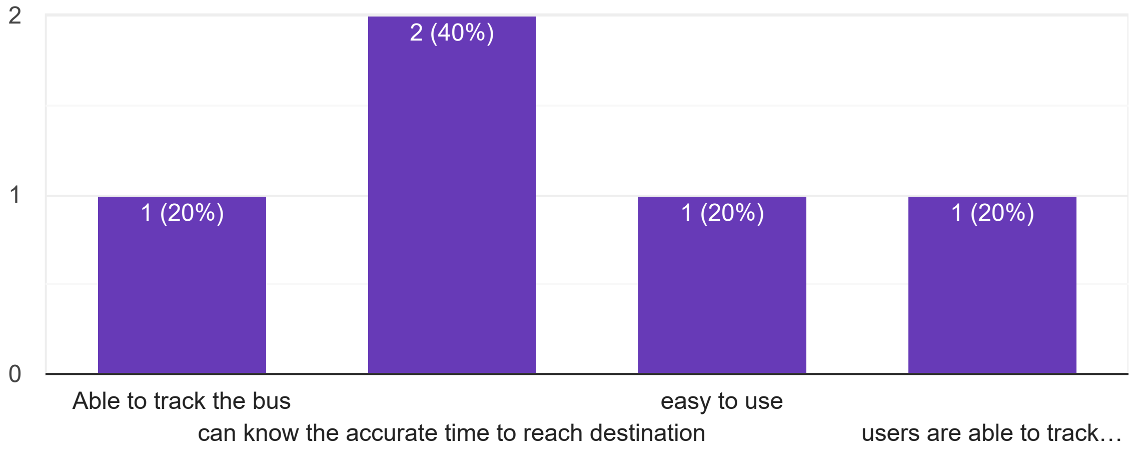
Most people would learn to use this application very quickly.

5 responses



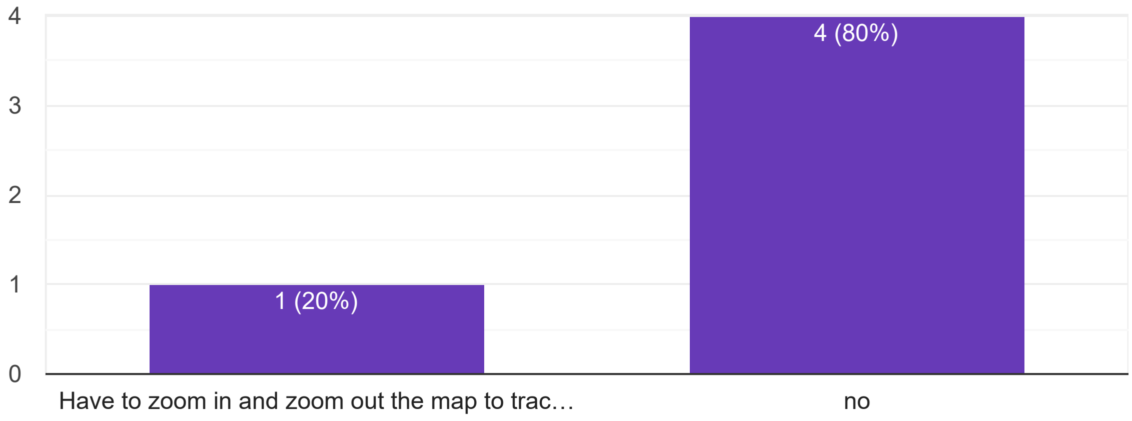
What do you like best about this application?

5 responses



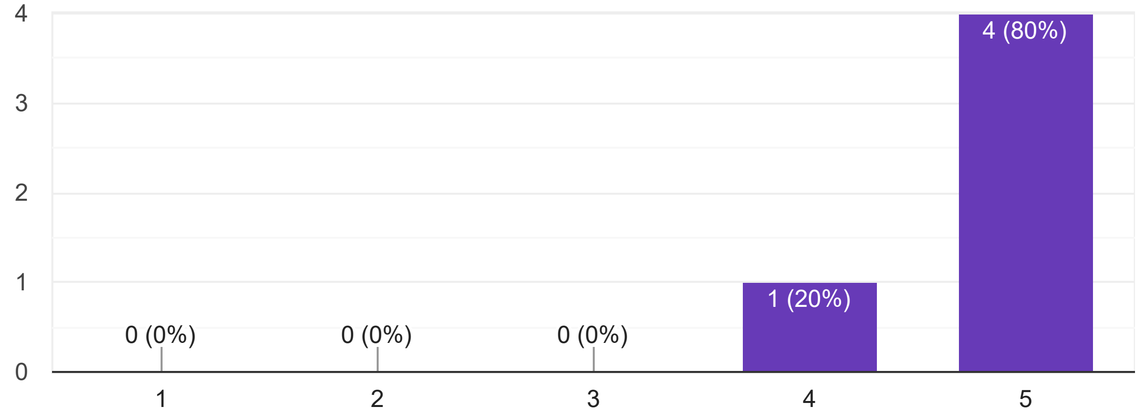
What do you like least about this application?

5 responses



How likely would you recommend this application to others?

5 responses



Do you have any other comments for this application?

2 responses

It is an user-friendly application

APPENDIX E: Lighthouse Application Audit Report

http://127.0.0.1:8000/drivers

74	81	100	80
Performance	Accessibility	Best Practices	SEO

▲ 0–49 50–89 90–100

There were issues affecting this run of Lighthouse:

- **There may be stored data affecting loading performance in this location: IndexedDB. Audit this page in an incognito window to prevent those resources from affecting your scores.**

74

Performance

Metrics

	First Contentful Paint	0.7 s	Time to Interactive	3.5 s
	Speed Index	0.7 s	Total Blocking Time	150 ms
▲	Largest Contentful Paint	3.4 s	Cumulative Layout Shift	0

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

View Original Trace

Opportunities — These suggestions can help your page load faster. They don't directly affect the Performance score.

Opportunity	Estimated Savings
Eliminate render-blocking resources	0.42 s
Remove unused CSS	0.4 s

Diagnostics — More information about the performance of your application. These numbers don't directly affect the Performance score.

- ▲ Serve static assets with an efficient cache policy — 3 resources found
- ▲ Ensure text remains visible during webfont load
- Avoid enormous network payloads — Total size was 3,480 KiB

Avoid chaining critical requests — 6 chains found

Keep request counts low and transfer sizes small — 12 requests • 3,480 KiB

Largest Contentful Paint element — 1 element found

Avoid large layout shifts — 1 element found

Avoid long main-thread tasks — 3 long tasks found

Passed audits (27)

81

Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

ARIA — These are opportunities to improve the usage of ARIA in your application which may enhance the experience for users of assistive technology, like a screen reader.

▲ Elements with an ARIA `[role]` that require children to contain a specific `[role]` are missing some or all of those required children.

Contrast — These are opportunities to improve the legibility of your content.

▲ Background and foreground colors do not have a sufficient contrast ratio.

Navigation — These are opportunities to improve keyboard navigation in your application.

▲ `[id]` attributes on active, focusable elements are not unique

Internationalization and localization — These are opportunities to improve the interpretation of your content by users in different locales.

▲ `<html>` element does not have a `[lang]` attribute

Names and labels — These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

▲ Links do not have a discernible name

Tables and lists — These are opportunities to improve the experience of reading tabular or list data using assistive technology, like a screen reader.

▲ Lists do not contain only `<li>` elements and script supporting elements (`<script>` and `<template>`).

Additional items to manually check (10) — These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review](#).

Passed audits (16)

Not applicable (22)

100

Best Practices

Passed audits (17)

Not applicable (1)

80

SEO

These checks ensure that your page is optimized for search engine results ranking. There are additional factors Lighthouse does not check that may affect your search ranking. [Learn more](#).

Content Best Practices — Format your HTML in a way that enables crawlers to better understand your app's content.

▲ Document does not have a meta description

Crawling and Indexing — To appear in search results, crawlers need access to your app.

▲ Links are not crawlable

Additional items to manually check (1) — Run these additional validators on your site to check additional SEO best practices.

Passed audits (8)

Not applicable (4)

Runtime Settings

URL	http://127.0.0.1:8000/drivers
Fetch Time	Mar 26, 2021, 8:45 PM GMT+8
Device	Emulated Desktop
Network throttling	40 ms TCP RTT, 10,240 Kbps throughput (Simulated)
CPU throttling	1x slowdown (Simulated)
Channel	devtools
User agent (host)	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36
User agent (network)	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4143.7 Safari/537.36 Chrome-Lighthouse
CPU/Memory Power	1231
Axe version	4.1.1

Generated by **Lighthouse** 7.0.0 | File an issue