Strathmore University

Faculty of Information Technology

UNIVERSITY CARPOOL SYSTEM

Mark Kamau

91372

Sichangi Mecolela

91389

An Information System final documentation submitted to the Faculty of Information Technology in partial fulfillment of the award of a Degree in Bachelor of Science in Informatics and Computer Science

Date of submission: 14th November 2017

Declaration

We declare that this project has not been submitted to any other university for the award of a Degree in Bachelor of Science in Informatics and Computer Science.

Students' signatures:

Mark Kamau

Student No: 91372

Sign: ____________________________ Date: ____________________

Sichangi Mecolela

Student No: 91389

Sign: ____________________________ Date: ____________________

Supervisor's signature:

Dr. Joseph Orero, PhD.

Sign: ____________________________ Date: ____________________

Abstract

Many people living in Nairobi often suffer travelling to work or school in the morning or leaving in the evening. The main means of reaching work or school is through the institution's buses, a personal vehicle, public transport and walking. Due to the wanting public transport, many people would rather drive to their destination.

It was noted that some of the members of Strathmore who are going in the same direction drive alone. Hence, we have a situation where hundreds of vehicles are on the road, leading to the severe traffic jams experienced daily. Moreover, it increases the amount of pollution which has a negative impact on the environment. Strathmore University is not immune to this problem. The number of the students and staff drive to institution is larger than the available parking spaces. This leads to the parking lot filling up early in the morning. Anyone who misses a spot will have to look for another space in the vicinity or return the vehicle home.

This problem can solved through the use of carpooling. Carpooling is the sharing of a car journeys so that more than one person uses the vehicle. By just a few people in one vehicle instead of in separate ones, traffic problem will immediately start to reduce.

In this project, an Android application has been developed that allows members of Strathmore to have an easier and stress free means of coming to the institution. Users are able to register as either a driver and provide transport or as a rider seek transport respectively. The system matches users coming from the same area and enable the sharing a ride. This will not only alleviate the congestion of vehicles within the organization but will also reduce the increasing congestion levels on the roads.

Table of Contents

List of figures

Chapter 1: Introduction

1.1 Background

Many people living in Nairobi often suffer travelling to work or school in the morning or leaving in the evening. The main means of reaching work or school is through the institution's buses, a personal vehicle, public transport and walking. Due to the wanting public transport, many people would rather drive to their destination.

It was noted that some of the members who are going in the same direction drive alone. Hence, we have a situation where hundreds of vehicles are on the road, leading to the sever traffic jams we experience daily. Moreover, it increases the amount of pollution which has a negative impact on the environment. This problem has been attempted to be fixed through the expansion of roads. This is insufficient because the roads which face the largest congestion, such as Mombasa Road, do not have any room for expansion. Another attempt was the building of the bypass. While this has had an impact, it is still not sufficient. It is designed for people on one side of the city to easily get to another side. As such those within the city are still affected.

The university offers transportation for its members to and from the CBD at specific times of the day; early in the morning to the university and twice in the evening while taking them back to the CBD. For one to use the bus, they only require there Strathmore identification card. There are also numerous hostels in its surroundings which makes it possible for a portion of its members to reach the institution through walking. For the rest of the population who do not use either of these means of transport, they either have personal vehicles, or use public means. For those with private vehicles, there is a parking area within the school grounds. Some staff members, especially those in administrative positions, have a designated parking area within the first phase of the school whereas the students and other members have a larger parking area within the second phase of the school.

Strathmore is continuously growing as an institution with more staff and students joining the institution. This has both good and bad implications to the institution. For the latter, the facilities are quite affected, especially the parking space. It was also noted that some of the members of the institution singly arrive in a vehicle which leaves numerous other seats that could be used to carry others. This would not only be advantageous to the institution but also to the environment as we try to fight global warming.

1.2 Problem statement

The issue with the current system is that people living within a certain area often use different means of transport towards the same place. This has contributed to road congestion as well as facility strain in the institution; there are only a limited number of parking spaces compared to the number of members with personal vehicles who would like to use them. During the semester, the parking is often filled up by 9 am.

1.3 Aim

To develop an Android application that will allow Strathmore members to easily find nearby members and use a shared means of transport to the institution.

1.4 Specific objectives

i. To analyze existing carpool systems.
ii. To investigate challenges faced by existing carpool systems
iii. To design, develop and test an Android based carpool system.

1.5 Research questions

i. What are the existing carpool systems?
ii. What are the challenges faced by the users?
iii. What are the available carpool systems?
iv. Does the developed system function?

1.6 Justification

The proposed system is a worthwhile undertaking because it will reduce congestion and pollution on Kenyan roads and in the institution as well as give members of Strathmore an efficient and headache free means of transport towards and from the institution.

1.7 Scope/limitation

The system will only be available for members of Strathmore. This is to provide accountability in case of any security issues.

The application will only be available on Android. An IOS version will not be available.

The application will require an active internet connection.

The application will require location services.

The application will not facilitate payment between the driver and the riders.

Chapter 2: Literature Review

2.1 Introduction

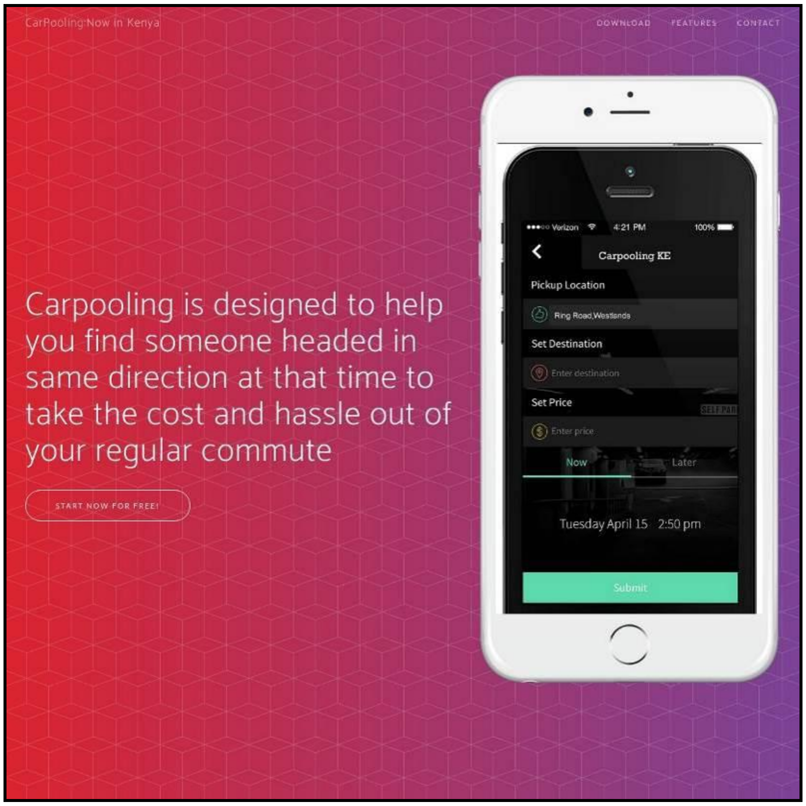
The transport systems available in Kenya are walking, public transport, private vehicles, an institution owned bus or a car for hire with a driver. Institution owned buses are restricted to members of the institution, private vehicles to their owners and the prices of car for hire are often high so the only options left are public transport and walking, the latter of which is not practical in some situations.

This chapter reviews the existing carpool systems, identifying their challenges and their gaps.

2.2 Existing carpool systems

2.2.1 Carrambee

According to an article written by The Standard in 2014 Carrambee is a web based carpooling facilitator developed by Edwin Ongola. Its website is supposed to be carrabmee.com but the domain is available for purchase. There is a Facebook and a Twitter profile of the same name, however, the last activity on the pages were in 2014 and 2015 respectively and offer no insight into the organization. Another website in the article is carpooling.co.ke (© 2016 carPooling Kenya)



Carpooling is designed to help you find someone headed in same direction at that time to take the cost and hassle out of your regular commute

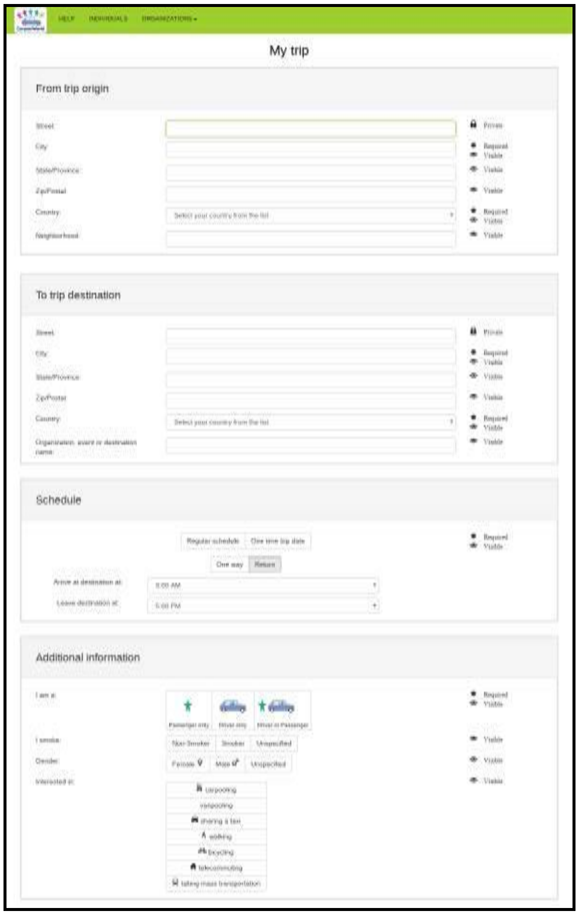
Figure 2.0.1: Carpooling.co.ke homepage. Source: carpooling.co.ke

This website is active, however, the links to download their apps or contact them lead nowhere. The only other mention of the Carrambee is from three nearly identical articles posted early this year that only list it as an available carpooling facilitator. It is therefore safe to assume that Carrambee is no longer active, if it ever was.

2.2.2 CarpoolWorld

CarpoolWorld is an international ride sharing app that regards itself as Kenya’s most popular rideshare. It has 481881 registered users across the world.

Their website is https://www.carpoolworld.com (© 2000-2017 Datasphere Corporation). It offers a page to add details about your ride after which it will send a confirmation code to your email.



HELP INDIVIDUALS ORGANIZATIONS ▾

My trip

From trip origin

Street: Private

City: Required Visible

State/Province: Visible

Zip/Postal: Visible

Country: Select your country from the list Required Visible

Neighborhood: Visible

To trip destination

Street: Private

City: Required Visible

State/Province: Visible

Zip/Postal: Visible

Country: Select your country from the list Required Visible

Organization, event or destination name: Visible

Schedule

Regular schedule | One time trip date Required Visible

One way | Return

Arrive at destination at: 8:00 AM

Leave destination at: 5:00 PM

Additional information

I am a: Passenger only | Driver only | Driver or Passenger Required Visible

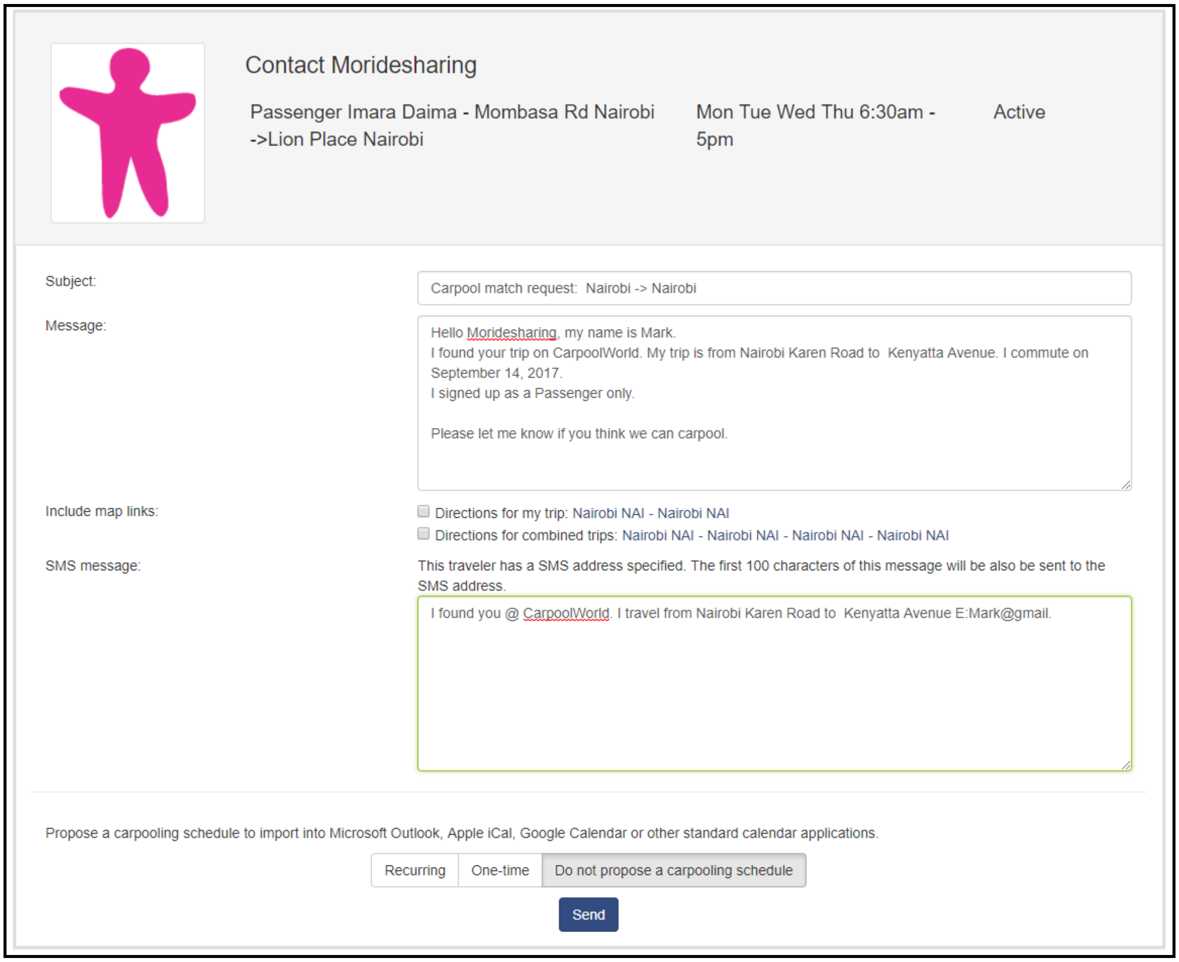
I smoke: Non Smoker | Smoker | Unspecified Visible

Gender: Female ♀ | Male ♂ | Unspecified Visible

Interested in: Visible
carpooling
vanpooling
sharing a taxi
walking
bicycling
telecommuting
taking mass transportation

Figure 2.0.2: CarpoolWorld signup form. Source: https://www.carpoolworld.com/trips_form.html

After logging in, it allows you to edit your trip, delete it or add a new trip. The system attempts to match your trips with other similar ones based on filters such as passengers or drivers, smokers or non-smokers, gender and regular schedule or one time trip. Once it matches you or you find a trip that you want, it offers you a contact form communicate with the person.



Contact Moridesharing

Passenger Imara Daima - Mombasa Rd Nairobi ->Lion Place Nairobi

Mon Tue Wed Thu 6:30am - 5pm

Active

Subject: Carpool match request: Nairobi -> Nairobi

Message:

Hello Moridesharing, my name is Mark.
I found your trip on CarpoolWorld. My trip is from Nairobi Karen Road to Kenyatta Avenue. I commute on September 14, 2017.
I signed up as a Passenger only.

Please let me know if you think we can carpool.

Include map links:

□ Directions for my trip: Nairobi NAI - Nairobi NAI
□ Directions for combined trips: Nairobi NAI - Nairobi NAI - Nairobi NAI - Nairobi NAI

SMS message: This traveler has a SMS address specified. The first 100 characters of this message will be also be sent to the SMS address.

I found you @ CarpoolWorld. I travel from Nairobi Karen Road to Kenyatta Avenue E:Mark@gmail.

Propose a carpooling schedule to import into Microsoft Outlook, Apple iCal, Google Calendar or other standard calendar applications.

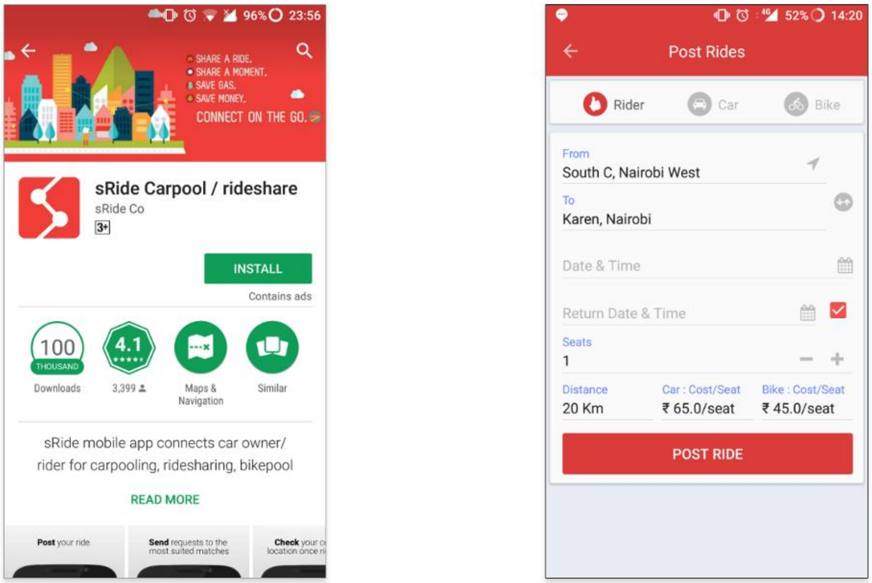
Figure 2.0.3: CarpoolWorld contact form. Source: https://www.carpoolworld.com/matches_email.html?ttin=604606&to_trip=561723

They also have an Android application but it only opens a browser window in the app.

2.2.3 sRide

sRide is an India based carpool app but is able to function in any country. You register in the app using a phone number and after verification, you are asked to input more details about yourself (first name, last name and email). After registration, you can post for a new ride as rider or a driver. As a driver you can either use a car or a bicycle.

The app also allows you to view rides posted by your friends, rides that you've been matched to, recommendations based on past rides, rides that are along your way and rides that are nearby. Due to its lack of usage in Kenya, its features could not be tested. Their website is http://sride.co



sRide Carpool / rideshare

sRide Co

sRide mobile app connects car owner/ rider for carpooling, ridesharing, bikepool

From: South C, Nairobi West

To: Karen, Nairobi

Seats: 1

Distance: 20 Km

Car : Cost/Seat: ₹ 65.0/seat

Bike : Cost/Seat: ₹ 45.0/seat

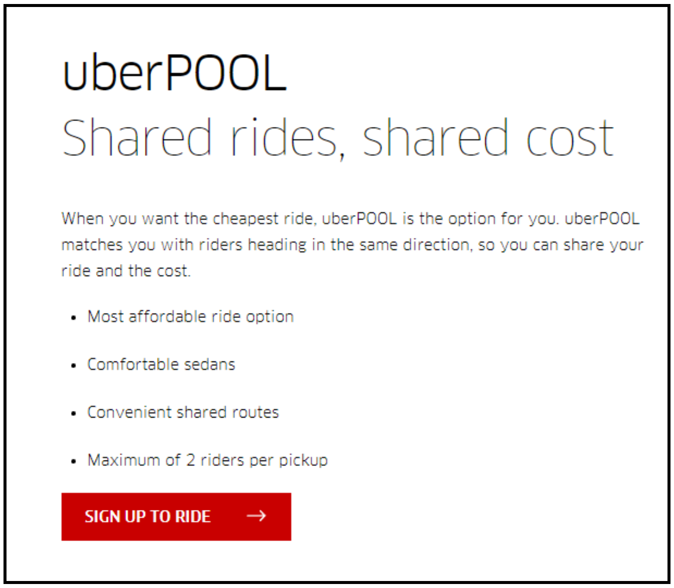
Play Store listing

Post new ride

Figure 2.0.4: sRide Android application

2.2.4 uberPOOL

Uber has a carpooling functionality that enables users to be matched with other riders along their route. It is however not yet available in Kenya. The webpage for the feature is https://www.uber.com/en-KE/ride/uberpool/ (© 2017 Uber Technologies Inc.)



uberPOOL

Shared rides, shared cost

When you want the cheapest ride, uberPOOL is the option for you. uberPOOL matches you with riders heading in the same direction, so you can share your ride and the cost.

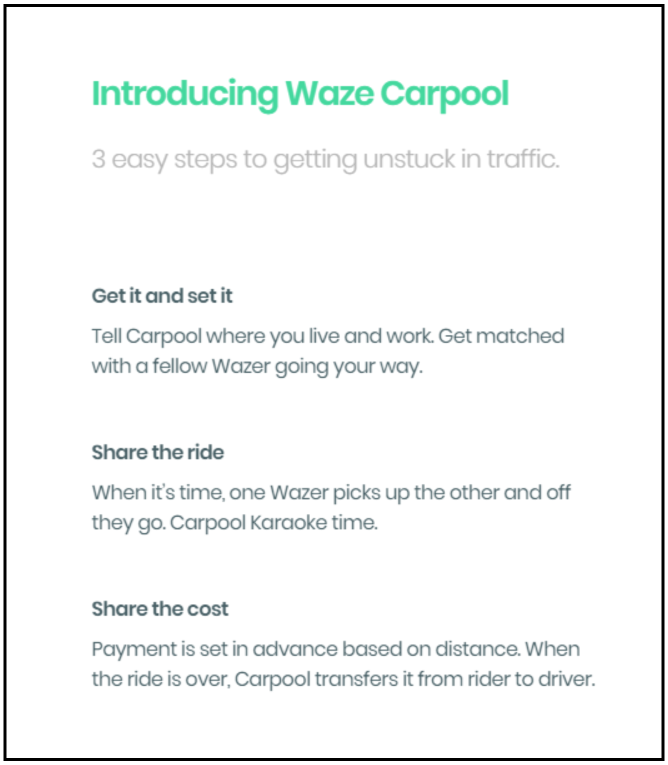
- Most affordable ride option
- Comfortable sedans
- Convenient shared routes
- Maximum of 2 riders per pickup

Figure 2.0.5: About uberPOOL. Source: https://www.uber.com/en-KE/ride/uberpool/

2.2.5 Waze Carpool

Another popular existing system is Waze Carpool. It is however not available in Kenya.

Their website is https://www.waze.com/carpool/ (© 2006-2017 Waze Mobile)



Introducing Waze Carpool

3 easy steps to getting unstuck in traffic.

Get it and set it

Tell Carpool where you live and work. Get matched with a fellow Wazer going your way.

Share the ride

When it's time, one Wazer picks up the other and off they go. Carpool Karaoke time.

Share the cost

Payment is set in advance based on distance. When the ride is over, Carpool transfers it from rider to driver.

Figure 2.0.6: About Waze Carpool. Source: https://www.waze.com/carpool/

2.3 Challenges faced by users

By far the biggest issue with the existing system is their lack of popularity in Kenya. Their functionality works in theory however there are not enough people in Kenya that are on their platforms. This is made worse by the fact that their users don’t have a common purpose for them to use the system, for example, a common destination. This causes their small user base to be spread thin.

Another challenge is security. Due to their hands-off approach they do not have any form of verification of the riders or more importantly, the driver. In the case of a crime, the data they have on their users cannot be trusted as legitimate. Our system will not face this challenge this due to it being limited to only members of Strathmore, the data on the users can be trusted.

2.4 Gaps in existing systems

Due to their inadequate popularity in Kenya, the features couldn’t be tested enough for any gaps to be identified.

Chapter 3: Development Methodology

3.1 Introduction

A system development methodology refers to the framework that is used to structure, plan, and control the process of developing an information system.

3.2 Development Model

We will be using an object-oriented analysis design (OOAD) as the system development methodology. It can be defined as a structured method for analyzing, designing a system by applying the object-orientated concepts. Under this methodology we will use an iterative model.

“Iterative and incremental software development (IID) offers a reliable and verified method for software development in the situation of insufficiently precise or evolving user requirements” (Ljubović, 2009). The iterative model starts with implementation of a small set of the requirements and iteratively enhancing it by implementing more requirements of the system over the development period.

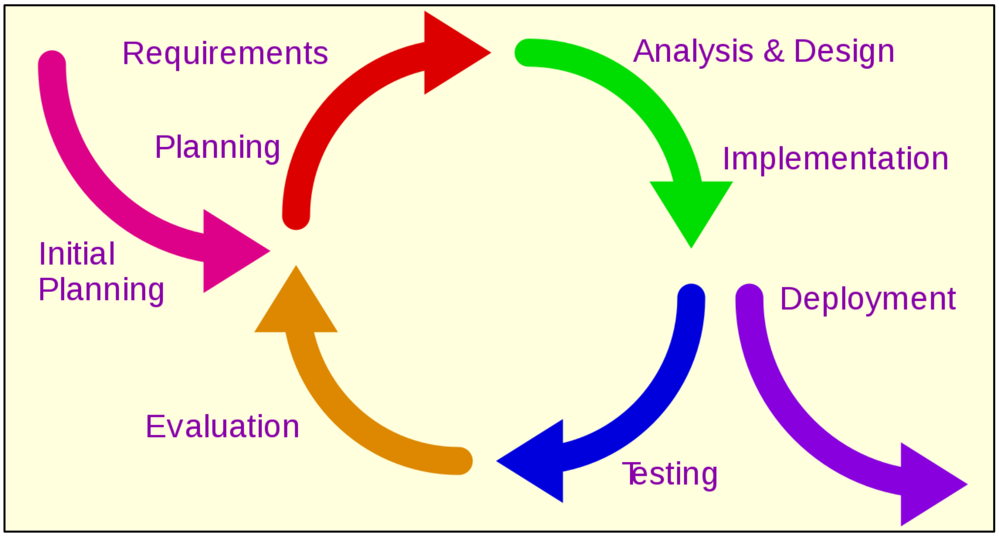


Figure 3.0.1: Iterative system development model. Source: https://commons.wikimedia.org/wiki/File:Iterative_development_model.svg

The model has five distinct stages:

3.2.1 Planning

This stage identifies the problem(s) that the system is intended to fix. The requirements and the scope of the system are specified at a high level. This stage also involves carrying you a feasibility study to determine whether or not the system is viable. Feasibility study looks at:

Economic feasibility - whether the system can be afforded and whether it is worth the money.
Operational feasibility - Whether the system will solve the problem and is of benefit.
Technical feasibility - Whether the developers are capable of developing the system and whether the end users are capable of using it once it is completed.

An advantage of this model is that since the system goes through the stages multiple times before its final release, defects are quickly found and rectified before they accumulate. Moreover, it allows for higher user interaction since they are progressively given the system as it is developed.

3.2.2 Analysis and design

In this stage, the system is broken down into its constituent modules and their requirements are given in more detail. Screen layouts and progress diagrams are made at this stage.

3.2.3 Testing

The system is tested at the various levels of software testing such as:

Unit testing - Individual modules are tested separately.
Integration testing - Modules are combined and tested as a group.
System testing - The whole system is tested as a whole.
User acceptance testing - It is verified that the system provides a solution to the end user.
This stage is crucial because it prevents a faulty system being deployed to the user.

3.2.4 Implementation

In this stage, the system is deployed to the user. Any training or transition from previous systems is performed at this stage.

3.2.5 Evaluation

Finally, the effectiveness of the system is measured and potential enhancements are specified.

3.3 Deliverables

After the completion of these project, the following will be available:

3.3.1 Android Application

An android application that enables easy sharing of rides. It will be the users means of interacting with the system.

3.3.2 Backend

A backend system that will handle storage user and ride details. Moreover, it will facilitate ride sharing.

3.3.4 System Documentation

A system documentation describing the relationship between different components of the application.

3.4 System Tools and Techniques

3.4.1 Firebase

A backend as a service platform by Google that provides services such as authentication, real time database and messaging.

Authentication - It has been used because it provides a simple and secure way to authenticate users.

3.4.2 Crashlytics

Crashlytics is a crash reporting solution by Fabric. It provides immediate reporting and in-depth information on the cause and source of crashes in an application.

3.4.3 Android Studio

The official IDE for Android development. It is a fork of IntelliJ IDEA managed by Google. It provides various advantages for Android development:

Advanced code completion and intellisense.

Android specific code refactoring and lint checks.

A rich layout editor that supports drag and drop.

Virtual device manager.

ProGuard integration and app-signing capabilities.

3.4.4 Kotlin

It is a “statically typed programming language for modern multi-platform applications” (kotlinlang.org) that is 100% interoperable with Java. It was first introduced in 2011 by JetBrains, the developers of IntelliJ IDEA and other IDEs. During Google I/O 2017, the Android team announced first-class support for Kotlin. It provides various advantages over Java, most notably, the reduction or complete removal of null pointer exceptions, “The Billion Dollar Mistake” (Hoare, 2009)

3.4.5 GraphQL

“GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data. GraphQL provides a complete and understandable description of the data in your API, gives clients the power to ask for exactly what they need and nothing more, makes it easier to evolve APIs over time, and enables powerful developer tools.” (http://graphql.org/). It has been used in the backend to provide data access.

Chapter 4: System Analysis and Design

Analysis is defined as “the procedure by which we break down an intellectual or substantial whole into parts or components.” (Ritchey, 2009)

4.1 System Requirements Analysis

4.1.1 Functional Requirements

These are what the system should accomplish in order for it to accomplish its intended purpose.

4.1.1.1 Authenticate users

Users should be able to securely create an account and sign in using their Strathmore email address.

4.1.1.2 Request for a ride

Riders will be able to request for a pickup on demand.

4.1.1.3 Schedule a ride

The rider will be able to schedule rides beforehand.

4.1.1.4 Offer to fulfil requests

Drivers will be able to offer to fulfil requests made by riders.

4.1.1.5 View ride history

Both riders and drivers will be able to view a history log of their past rides.

4.1.2 Non-Functional Requirements

These are extra qualities of the system that improve the user experience.

4.1.2.1 Complete Information

The application should give its users clear and adequate information in regards to those they will be riding with.

4.1.2.2 Intuitive design

The application should offer the users a simple and clear means to achieve their objective on the application, from setup to ride request to confirmation of payment and rating after the ride.

4.1.2.3 Privacy

The application should only request for their location when they intend to take a ride and not any other time.

4.2 System Narrative

Users are first required to sign up using a valid Strathmore email address. Once they have been verified, they will be able to choose whether they will function as a driver or as a rider.

If they choose to be a rider, they can enter their destination address (their current location will be automatically detected) and request a ride. From there, the drivers will be able to see their request and offer to fulfil it. Alternatively, the rider can schedule a ride and drivers can offer to fulfil the request. If they choose to function as a driver, they will be able to view current requests and scheduled requests and offer to fulfil them.

Once the driver picks the rider, they are to both check in. Once the ride if finished, they are to both check out. Both the rider and the driver will be able to view their past rides.

Users will also be able to personalize their profile.

4.3 Design Diagrams

The following designs show how the various components of the system interact with each other.

4.3.1 Use case Diagram

A use case diagram shows the key actors in a system as well as their behavior. Our class diagram show the rider and the driver and their operations such as signing up, logging and requesting for a ride.

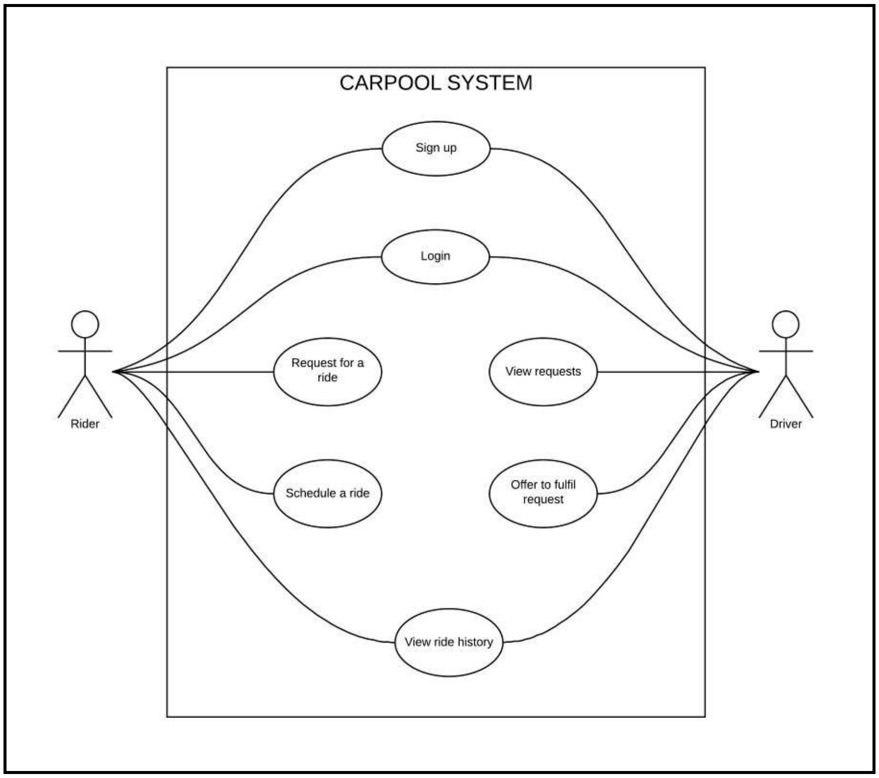


Figure 4.1: Use case diagram

4.3.2 Data Flow Diagram (DFD)

A data flow shows business processes and the data that flows between them. An example of such is when the ridder attempts to log in, their details are sent to the login process after then to the users' data store. After this a login response is sent back to the rider via the login process.

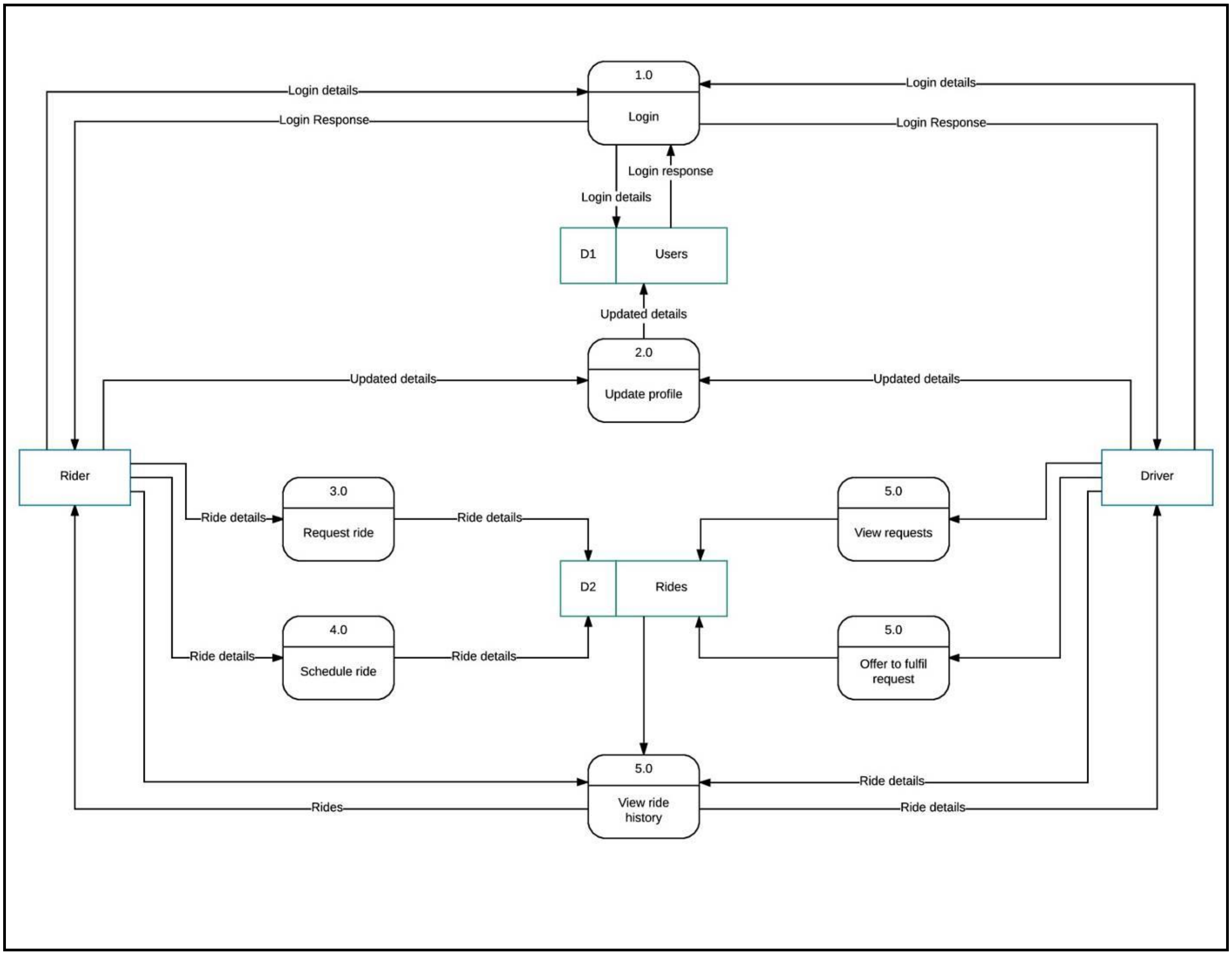


Figure 4.2 Data flow diagram

4.3.3 Entity Relationship Diagram (ERD)

Entity Relationship Diagrams depict only structural features and provide a static view of the system. In our case, a rider has an id, full name, email and phone number. They request for a ride while a driver fulfils a ride.

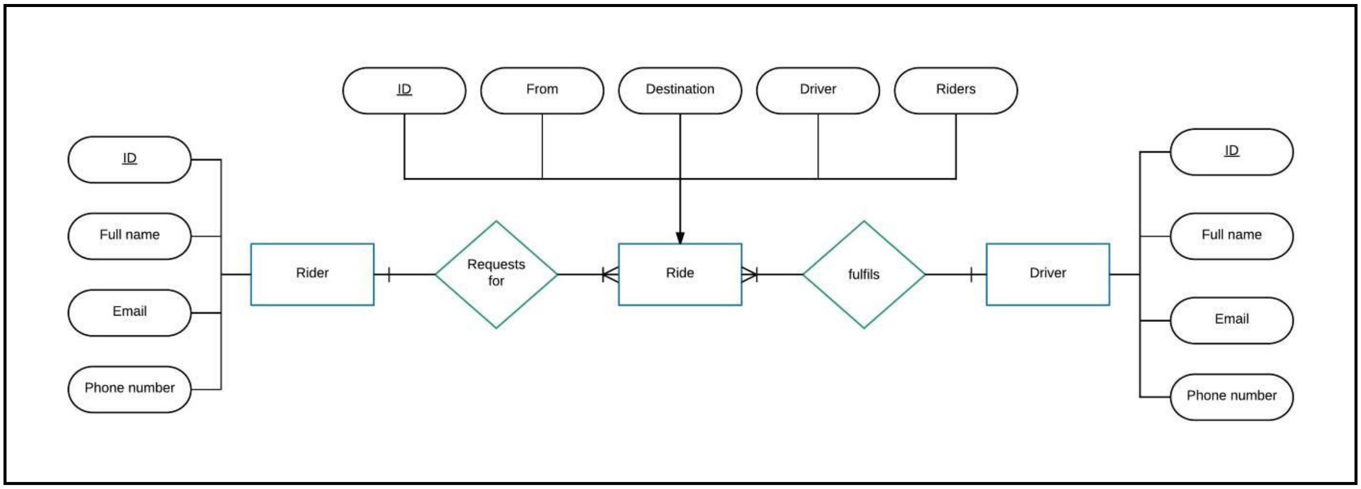
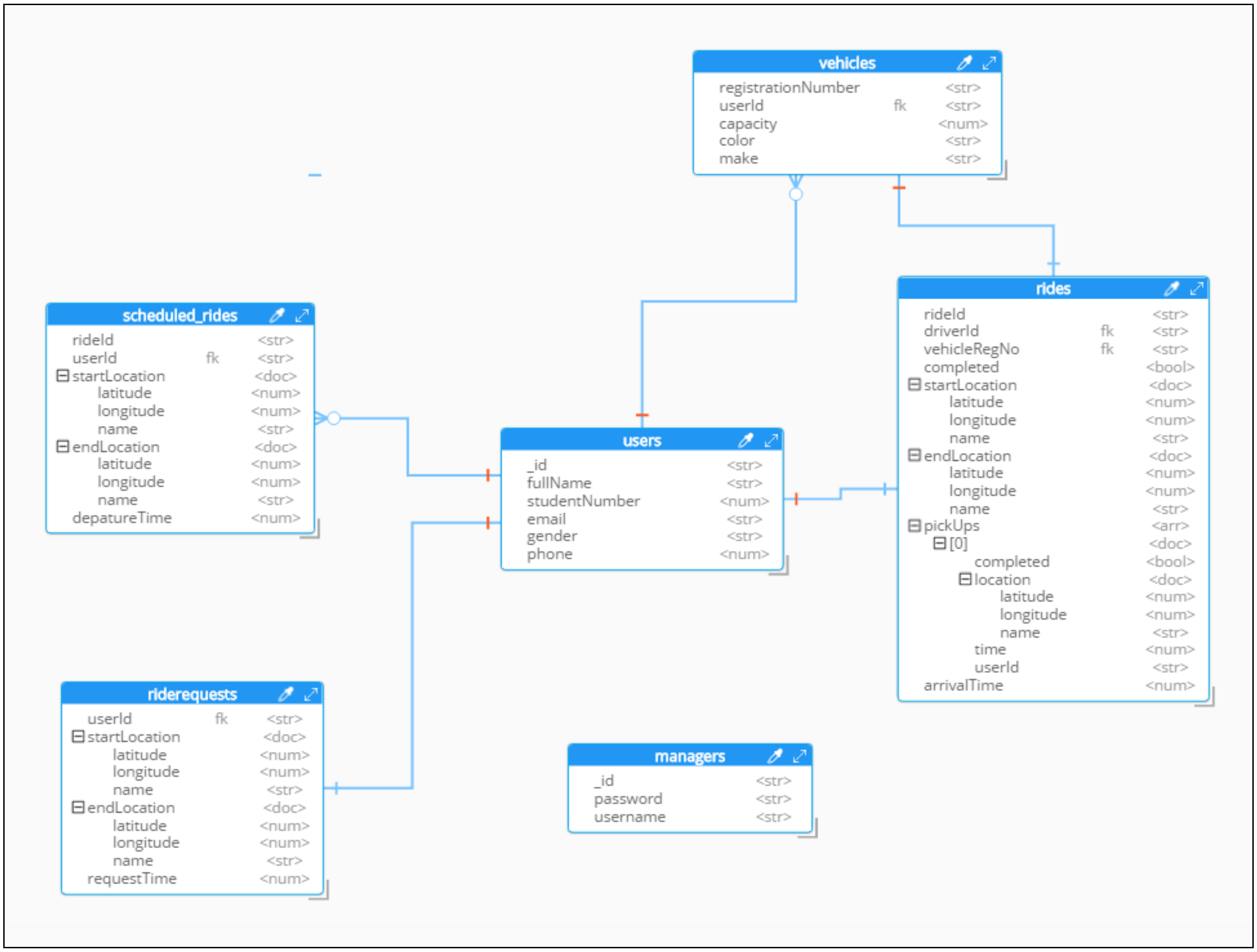


Figure 4.3 Entity relationship diagram

4.3.4 Database Schema



vehicles		
registrationNumber		<str>
userId	fk	<str>
capacity		<num>
color		<str>
make		<str>

scheduled_rides		
rideId		<str>
userId	fk	<str>
⊟ startLocation		<doc>
latitude		<num>
longitude		<num>
name		<str>
⊟ endLocation		<doc>
latitude		<num>
longitude		<num>
name		<str>
depatureTime		<num>

users	
_id	<str>
fullName	<str>
studentNumber	<num>
email	<str>
gender	<str>
phone	<num>

rides		
rideId		<str>
driverId	fk	<str>
vehicleRegNo	fk	<str>
completed		<bool>
⊟ startLocation		<doc>
latitude		<num>
longitude		<num>
name		<str>
⊟ endLocation		<doc>
latitude		<num>
longitude		<num>
name		<str>
⊟ pickUps		<arr>
⊟ [0]		<doc>
completed		<bool>
⊟ location		<doc>
latitude		<num>
longitude		<num>
name		<str>
time		<num>
userId		<str>
arrivalTime		<num>

riderequests		
userId	fk	<str>
⊟ startLocation		<doc>
latitude		<num>
longitude		<num>
name		<str>
⊟ endLocation		<doc>
latitude		<num>
longitude		<num>
name		<str>
requestTime		<num>

managers	
_id	<str>
password	<str>
username	<str>

Figure 4.5: Database Schema generated using Hackolade

Chapter 5: Implementation and Testing

5.1 Introduction

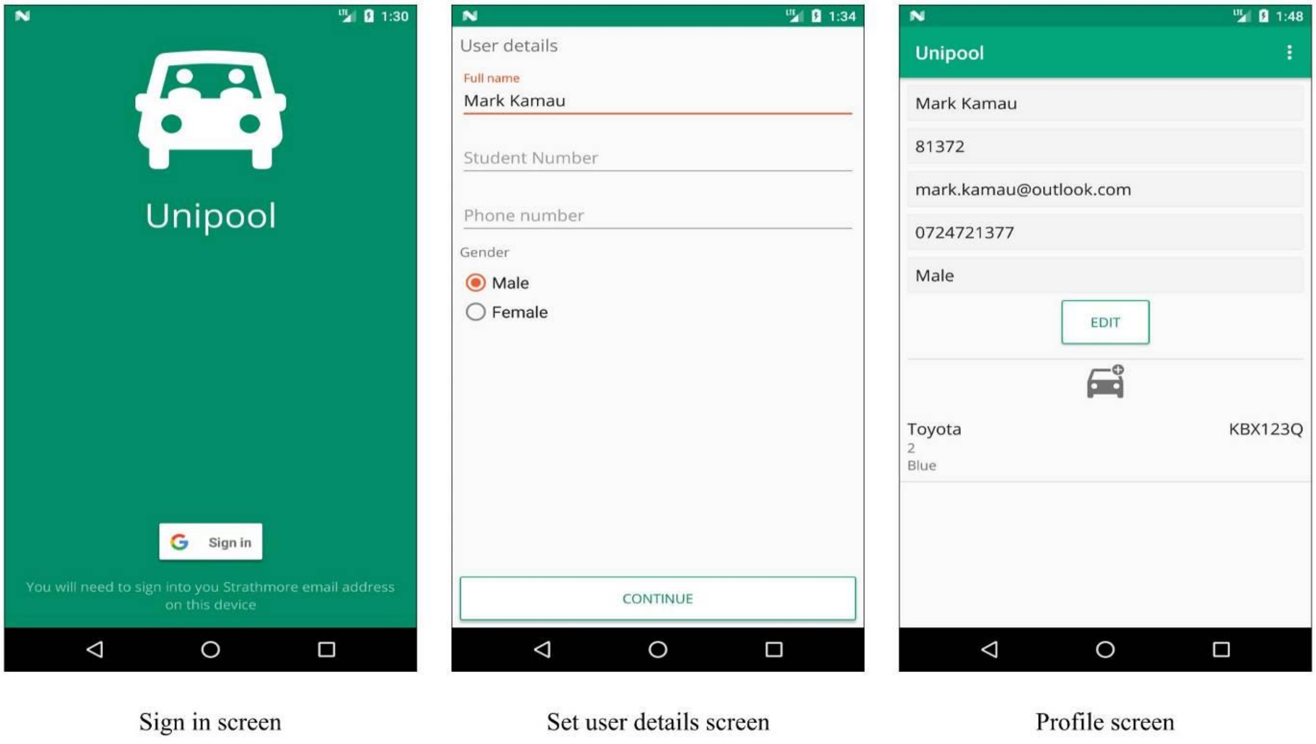
This chapter discusses the different interfaces and modules that have been developed in this project. The project contains 4 major modules.

5.2 User Module

The user module is used to handle user authentication and user details. Authentication uses Google Sign-In and requires use of a Strathmore google account (@strathmore.edu). Firebase is used to store the list of users that have made an account.

When the user first opens the app, they are presented with a screen that asks then to sign in. The sign in button brings up a dialog box that asks the user to pick an account that they have already signed into or add another account. Once that is completed, they are taken to a screen that requires them to input more user details if they have not previously made an account. Alternatively, they are taken to the main screen.

From the app bar, the user can access their profile using the person icon. The profile screen allows them to edit some of their details (name, phone number and gender). They can also add a vehicle from this screen. The details collected for the ride include: the registration number, the make of the car, the passenger capacity and the color.



Unipool

G Sign in

You will need to sign into you Strathmore email address on this device

User details

Full name
Mark Kamau

Student Number

Phone number

Gender
● Male
○ Female

CONTINUE

Unipool

Mark Kamau

81372

mark.kamau@outlook.com

0724721377

Male

EDIT

Toyota KBX123Q
2
Blue

Sign in screen

Set user details screen

Profile screen

Figure 5.1: User module screens

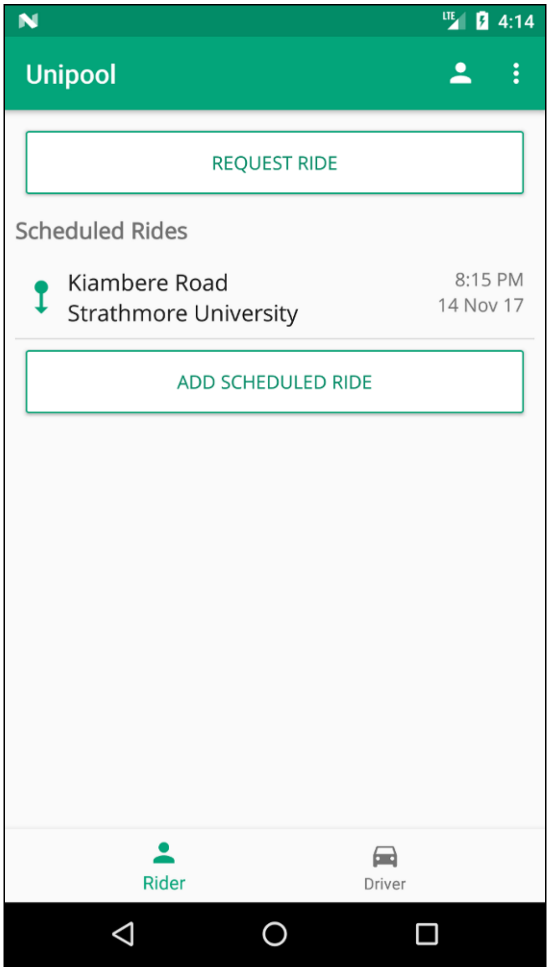
5.3 Ride Module

The ride module is used to facilitate ride sharing. It can be broken down into two sub modules:

5.3.1 Rider Ride Module

This module is accessible to all users once they have completed setting up their account. The user can select their start location and confirm they want a ride to the university's location. Once they have make the confirmation, drivers can see their location on the map. If the user exits the screen, the drivers will not be able to see them on the map.

Once a driver gives them an offer, then see the driver's name and how much they will be charged. They can either choose to accept the offer or reject it.



Kiambere Road
Strathmore University
8:15 PM
14 Nov 17

Figure 5.2: Rider home screen

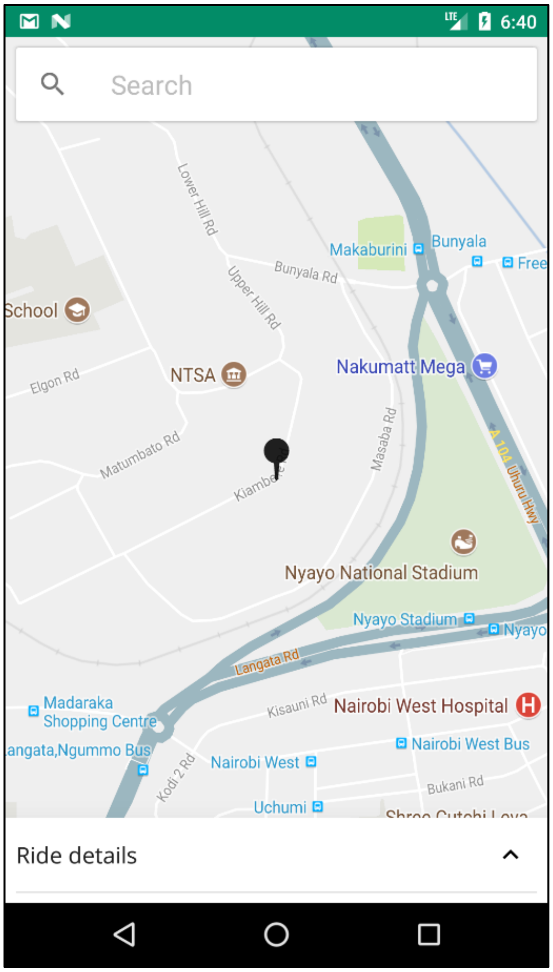
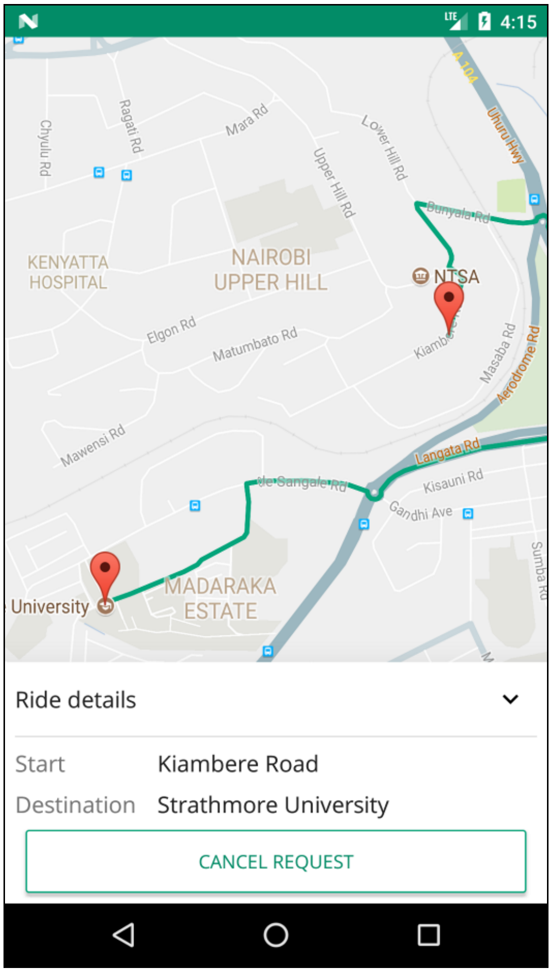


Figure 5.3: Rider map screen

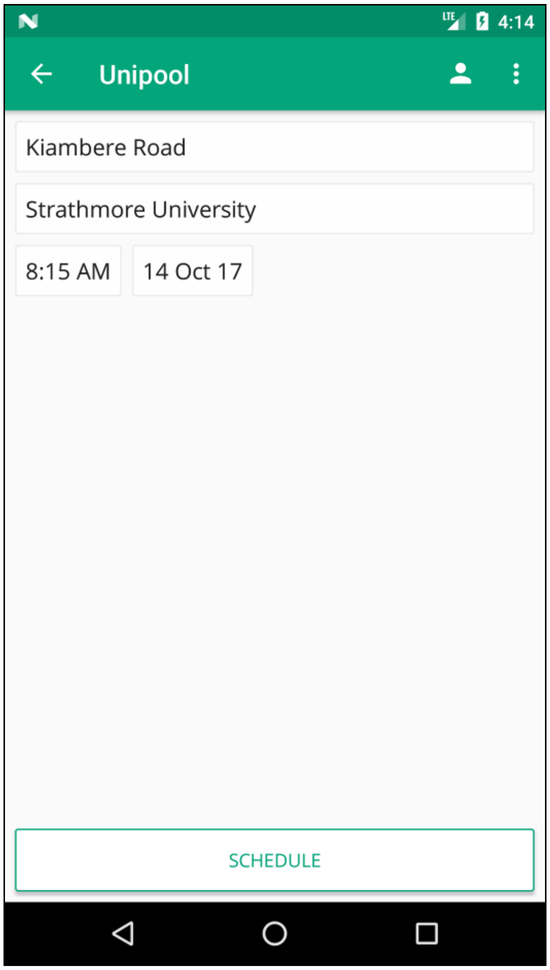


Ride details

Start Kiambere Road

Destination Strathmore University

Figure 5.4: Rider map screen - When a ride is being requested



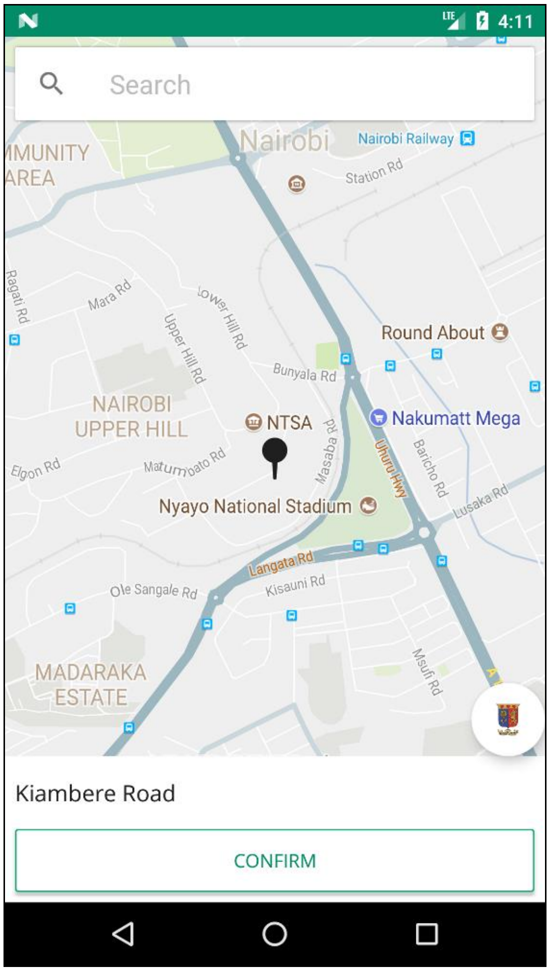
Kiambere Road

Strathmore University

8:15 AM

14 Oct 17

Figure 5.5: Schedule ride screen



Kiambere Road

Figure 5.6: Select location screen

5.3.2 Driver Ride Module

This module is only accessible to users if they have added at least one vehicle to their account. To use the module, they are first required to select which vehicle they are currently using, the amount they want to charge and their end location. They can then view the users who are currently requesting a ride and can choose offer them a ride. Once they have at least one rider, they can start the ride. On their screen, they will see a route that will pass by all the riders on their way to the institution. Once they pick up a rider, they set them as picked up in the application. They can also choose to remove a rider from the ride. When they arrive at the school, they mark the ride as completed.

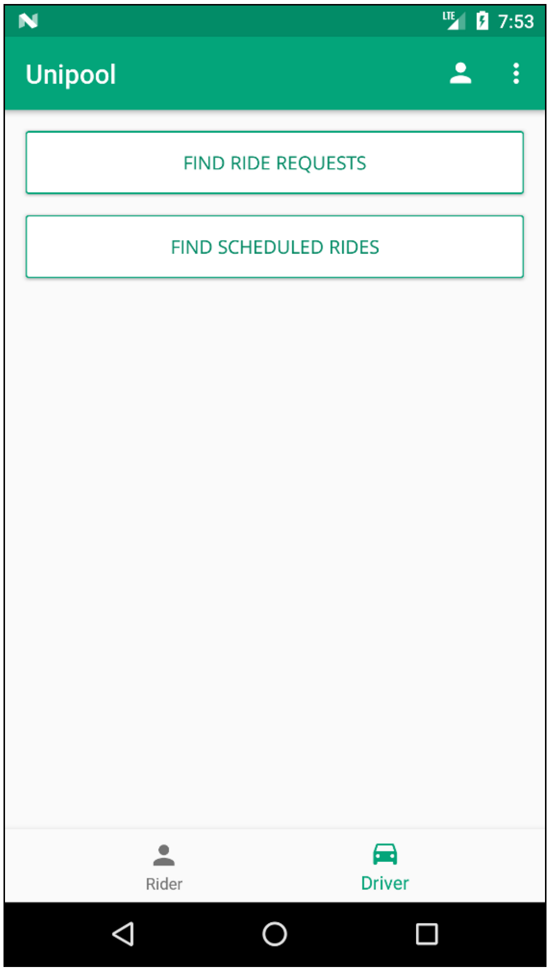
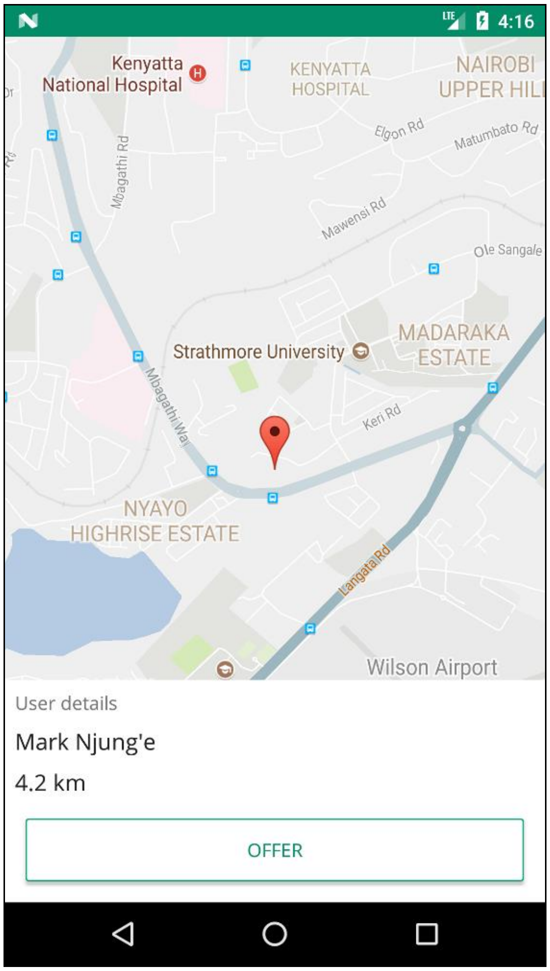


Figure 5.7: Driver home screen

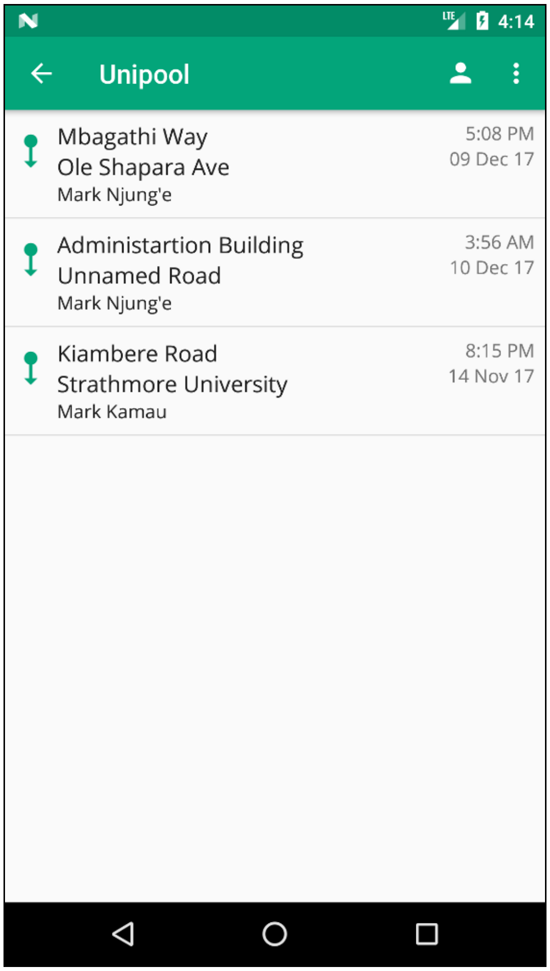


User details

Mark Njung'e

4.2 km

Figure 5.8: Driver map screen



Mbagathi Way
Ole Shapara Ave
Mark Njung'e
5:08 PM
09 Dec 17

Administartion Building
Unnamed Road
Mark Njung'e
3:56 AM
10 Dec 17

Kiambere Road
Strathmore University
Mark Kamau
8:15 PM
14 Nov 17

Figure 5.9: Driver scheduled rides screen

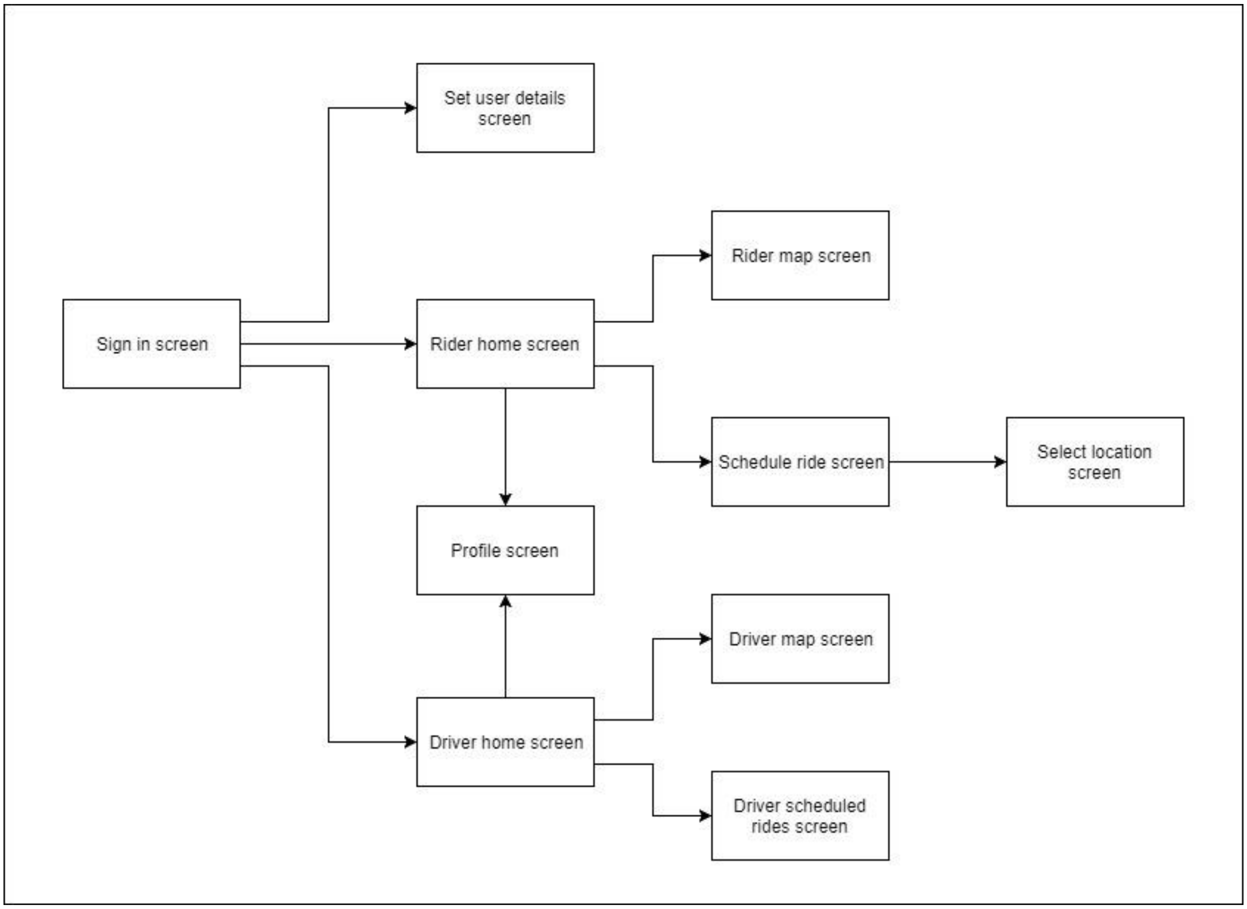


Figure 5.10: App screens flow

5.4 Backend Module

The backend module is used to facilitate data transfer between the mobile application and the database. It comprises of two parts: a GraphQL server and an MQTT broker. They are both built using NodeJS and hosted on Heroku (https://www.heroku.com/).

The GraphQL server is used to query the database and return data to the mobile app. The GraphQL schema is as shown below.

The MQTT broker is used to facilitate real time communication between users, specifically during a ride. When a driver makes an offer, the details are sent to the user. Their response is added to the data and sent back to the driver.

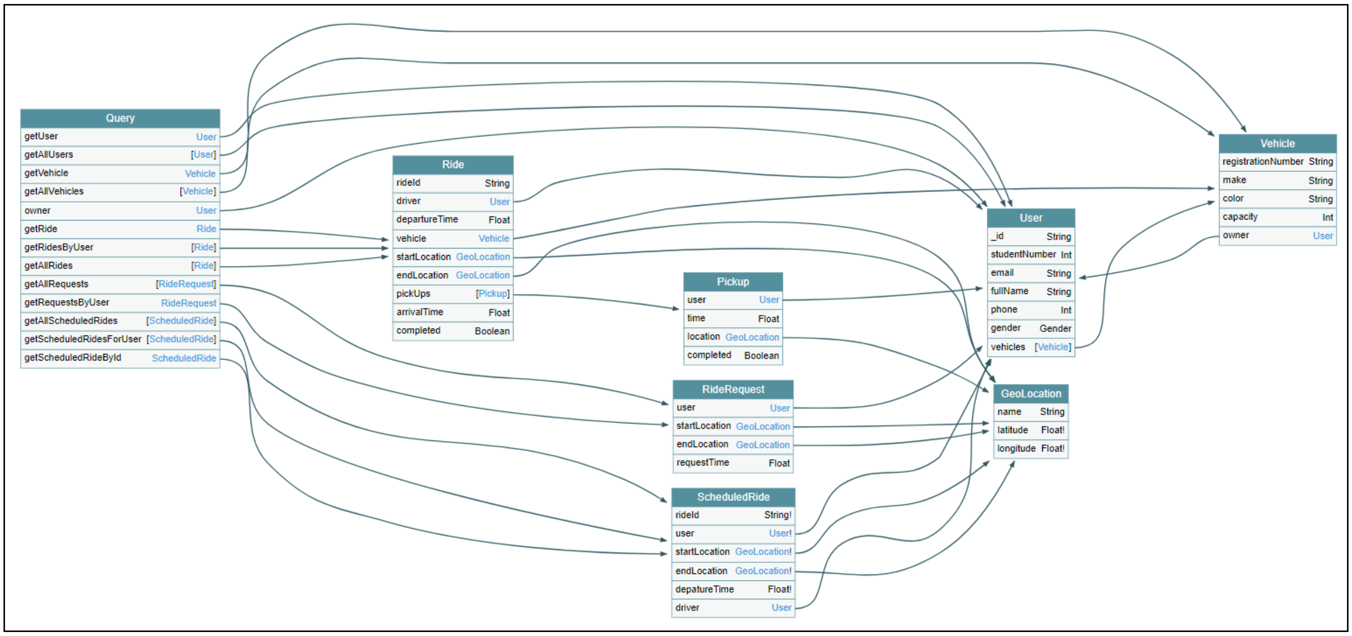


Figure 5.11: GraphQL schema generated using GraphQL Voyager

5.5 System Testing

“System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.” (IEEE, 1991). Testing is an important phase because it confirms that the system satisfies the requirements.

The different stages of testing include:

5.5.1 Development Testing

In this stage, the system is tested as it is developed. It is done by developers and designers

5.5.2 Release Testing

In this stage, the completed system is tested before it is released to users.

5.5.3 User Testing

In this stage, the users test the system in their own environment.

Chapter 6: Conclusions, Recommendations and Future Works

6.1 Conclusions

In conclusion, the system allows users to request to share a ride at the current time as well as schedule one for later.

6.2 Recommendations

Currently, the sign up is limited to members of Strathmore University. This regulation can be removed to allow the system to function for public user base.

A form of payment is also missing in the system. Inclusion of such a feature will give an incentive for more people to sign up as drivers.

6.3 Future Works

A future implementation of this system should include some form of payment. This will give an incentive for more people to join as drivers. In addition to that, the app is only available for Android. An IOS app should therefore be developed.

References

IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. (1991). IEEE. doi:10.1109/IEEESTD.1991.106963

Liquid Telecom cuts diesel use by one third on carpooling roll out, staff home broadband links. (2017, January 24). Retrieved September 13, 2017, from http://smedigest.co.ke/liquid-telecom-cuts-diesel-use-one-third-carpooling-roll-staff-home-broadband-lin/

Selecting a development approach. (2008, March 27). Retrieved November 2, 2017, from https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf

Today, B. (2017, September 05). Car pooling cuts Liquid Telecom transport cost. Retrieved September 13, 2017, from https://businesstoday.co.ke/car-pooling-cuts-liquid-telecoms-diesel-cost/

Sunday, F. (2014, August 11). Car-pooling website saves travellers costs. Retrieved September 13, 2017, from https://www.standardmedia.co.ke/business/article/2000131215/car-pooling-website-saves-travellers-costs

Hoare, T. (2009, August 25). Null References: The Billion Dollar Mistake. Retrieved September 10, 2017, from https://www.infoq.com/presentations/Null-References-The-Billion-Dollar-Mistake-Tony-Hoare

Ritchey, T. (1991). Analysis and synthesis: On scientific method – based on a study by Bernhard Riemann. doi:10.1002/sres.3850080402

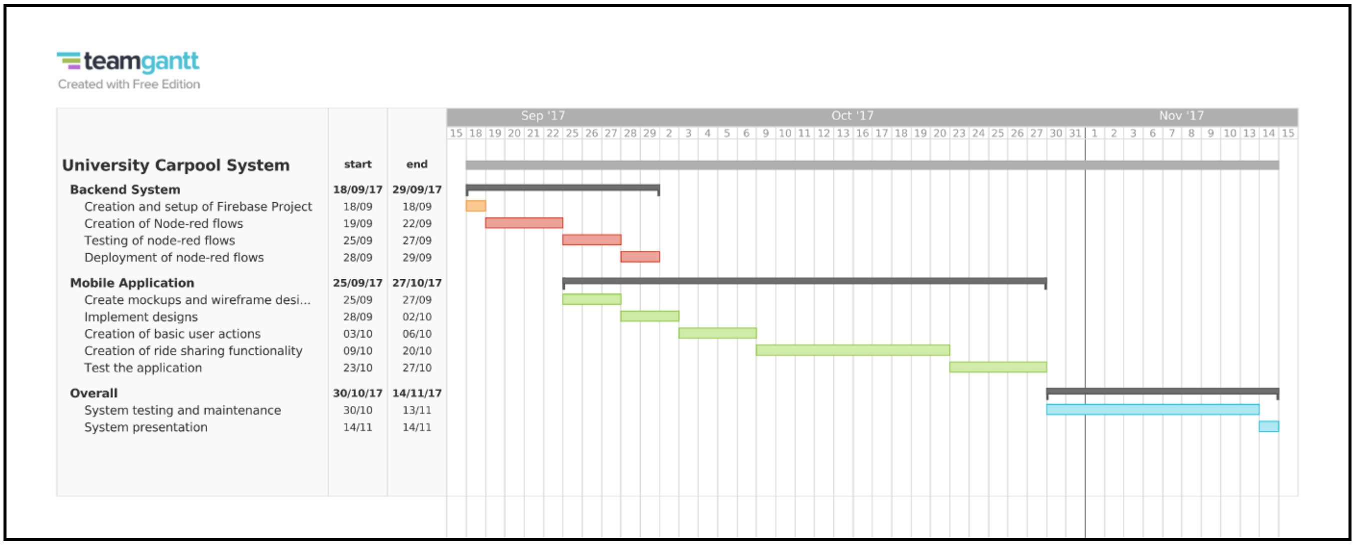
Amadala, V. (2017, January 24). Liquid Telecom cuts its diesel use by one third through carpooling and staff home broadband links. Retrieved September 13, 2017, from http://www.dhahabu.co.ke/2017/01/24/liquid-telecom-cuts-diesel-use-one-third-carpooling-staff-home-broadband-links/

Ljubović, V., & Šupić, H. (2009). An Iterative Approach in Development of the Student Information System: Lessons Learned. IEEE. doi:10.1109/ICAT.2009.5348414

Westlake, Z. (2017, August 18). Anything Java can do Kotlin can do better. Retrieved September 10, 2017, from https://medium.com/@Pinterest_Engineering/anything-java-can-do-kotlin-can-do-better-a1c1ddae8ffd

Appendix

Appendix A1: Time Schedule



University Carpool System	start	end
Backend System	**18/09/17**	**29/09/17**
Creation and setup of Firebase Project	18/09	18/09
Creation of Node-red flows	19/09	22/09
Testing of node-red flows	25/09	27/09
Deployment of node-red flows	28/09	29/09
Mobile Application	**25/09/17**	**27/10/17**
Create mockups and wireframe desi...	25/09	27/09
Implement designs	28/09	02/10
Creation of basic user actions	03/10	06/10
Creation of ride sharing functionality	09/10	20/10
Test the application	23/10	27/10
Overall	**30/10/17**	**14/11/17**
System testing and maintenance	30/10	13/11
System presentation	14/11	14/11

Appendix A: Time Schedule

Appendix A2: Supervision sheet

Appendix A3: Final system documentation marking sheet