ONT4300 PROJECT IV JUNE DOCUMENTATION

Hikeshare

216443881

*Sisikelelwe Maqabangqa*

SUPERVISOR

*Dieter Steenberg*

Contents

[1.1 State of public transport in South Africa 1](#_Toc23748222)

[1.2 Project Objectives 2](#_Toc23748223)

[1.3 Problem Background 2](#_Toc23748224)

[1.4 Related Systems Analysis 3](#_Toc23748225)

[1.5 References 4](#_Toc23748226)

[2.1 Plan to address objectives 5](#_Toc23748227)

[2.2 Project Scope 6](#_Toc23748228)

[2.2.1 Information Scope 6](#_Toc23748229)

[2.2.2 Functional Scope 6](#_Toc23748230)

[2.2.3 Communication Scope 6](#_Toc23748231)

[2.3 Business Requirements 6](#_Toc23748232)

[2.4 Hardware and Software Requirements 7](#_Toc23748233)

[2.4.1 Software Requirements 7](#_Toc23748234)

[2.4.2 Hardware Requirements 7](#_Toc23748235)

[2.5 Design Constraints 7](#_Toc23748236)

[2.5.1 Security Constraints 7](#_Toc23748237)

[2.5.2 Interface Constraints 7](#_Toc23748238)

[2.5.3 Performance Constraints 7](#_Toc23748239)

[2.6 High-level use case diagram 8](#_Toc23748240)

[3.1 UML Class Model / Diagram 9](#_Toc23748241)

[3.2 Initial Relational Database Model Diagram 10](#_Toc23748242)

[3.3 User Interface Design 11](#_Toc23748243)

[4.1 Choice of Tools 14](#_Toc23748244)

[4.1.1 Visual Studio Code 14](#_Toc23748245)

[4.1.2 MySQL 14](#_Toc23748246)

[4.2 Relational Database Model Diagram 15](#_Toc23748247)

[4.3 Extracts of complex code 16](#_Toc23748248)

[4.4 Source code references (if applicable) 17](#_Toc23748249)

[4.4.1 Converting images to byte arrays 17](#_Toc23748250)

[4.5 Problems encountered 17](#_Toc23748251)

[4.5.1 Making use of the google maps API 17](#_Toc23748252)

[5.1 Test cases 18](#_Toc23748253)

[5.1.1 Register Livestock 18](#_Toc23748254)

[5.2 User interface evaluation 20](#_Toc23748255)

[5.3 Changes implemented 21](#_Toc23748256)

[6.1 Study conclusion 22](#_Toc23748257)

[Test Plan and Results Appendix 23](#_Toc23748258)

Business case and Related Systems Overview

The following document aims to describe and give clarity on the problem that I have identified in society. The document will expand on the problem by defining and explaining the problem and what my solution for the problem is. The document will define the design, implementation requirements and expectations for an application that will solve the problem in terms of the development process, and describe the basic problem areas that created the requirement for the application.

# 1.1 State of public transport in South Africa

Transport is essential in the development of any country. We live in a fast-moving society. To travel from one point to another, individuals need a reliable means of transportation. Unfortunately, not all households have a private means of transportation.

In South Africa, the public transport industry comprises of three main modes of transport:

* Gautrain and the Metrorail Train Service
* Bus Industry
* 16-Seater Minibus-Taxi Industry

In the 2013 National Household Travel Survey, findings obtained revealed that 68.8% of South African households use taxi services daily, followed by commuter bus (21.1%) and commuter rail operations (9.9%). Public transportation is a mode of force because many people who make use of it usually have no other option. With the rates of fuel being where they are and the costs that come with owning a car, using public transport as ones main mode of transport does at first seem like an affordable choice, but this is still not a cheap alternative, especially for people who are travelling to far destinations or making use of public transport on a daily basis.

Public transport forces its passengers to sacrifice comfortability and people find themselves spending most of their time commuting. A big disadvantage to the taxi industry is the amount of time people tend to spend on the road travelling to and from work, as well as the cost of travelling which drains at least 20-40% of the commuters’ disposal income. What is also common is the high rates of sexual harassment reported and road accidents caused because of drivers who fail to follow the rules of the road. Despite there being different modes of transport available, people would much rather travel comfortably and affordably. What has become apparent is that people are using social media platforms to request lifts from one city to another, lifts to festivals and to form lift clubs for their daily communes.

I’ve deduced that the reason being the drawbacks that come with public transport. Although social media is a great platform to reach people, the social groups and posts requesting and offering these lifts rarely get much of a response. Finding people to share a ride with is the challenge of carpooling as it is difficult to find a person going to the same place as you at a given time.

I have come across many posts online with no answer to them and groups created that have little activity. In cases where users do get responses, details and arrangements are made over the comments section which I find not to be ideal as things such as cell phone numbers and the users address, for example, is fairly private information seeing that the post is made public to people from all over the social media platform which is being used.

# 1.2 Project Objectives

“Carpooling is a collective transport system based on shared use of private cars, whose objective is to reduce the number of cars in use by grouping people.”

Carpooling has minimal incremental costs because it makes use of seats that would otherwise be empty, and the people can split the cost amongst themselves.

Carpooling is efficient and successful generally for journeys that have predictable or set schedules such as commutes or attending events and it is a mode of transport which can be casual or can be organized as a service.

The objective of the project is to present a web-based application which provides a communication platform between car owners and passengers, allowing them to reduce their trip expenses by sharing rides.

Users will be able to:

* Allow users to request and offer rides
* Allow users to receive requests and offers
* Allow users to share trips
* Allow users to exchange contact details and to plan travel arrangements
* Allow users to rate users they have shared trips with

# 1.3 Problem Background

Travelling when it comes to going to work can be very expensive when people live far from the workplace, which most commonly the case when people work in big cities. “People are spending an increasing amount of time commuting between their residence and workplace. An important factor behind this trend is related to residential affordability as housing located further away from central areas (where most of the employment remains) is more affordable. Therefore, commuters are trading time for housing affordability” - (Rodrigue, 2016).

The convenience of staying further away from the workplace sees people spending more money on gas as they must travel long distances to work and as gas prices are continually increasing in our country, people’s expenses are increasing as well. “Motorists will pay more for a litre of petrol at the pumps from Wednesday (6 March), thanks to an increase in the international petroleum price, and a weaker local currency. By the end of the hikes, SA motorists will be paying R5.63 and R5.49 in taxes per litre for petrol and diesel, respectively.” - (Writer, 2019).

Another problem with travelling that must be considered is public transport. Although public transport is seen as a cheap and affordable solution by some, many challenges come along with it.

“Despite the available modes of transportation, South African transport is still plagued with several challenges. These comprise of low ridership, lack of public transport accessibility in rural areas, equity imbalances and congestion. The South African public transport industry is currently under immense inquiry as captive users of these systems face unsafe, unreliable and costly systems.” - (Walters, 2008).

Sources of public people use to travel range from trains, mini-bus taxis, and buses.

People’s main complaints with these modes of transport are usually along the lines of the transport being very overcrowded, unsafe, lack of structure of scheduling (In the case of minibuses there is no schedule which results in people being late if they travel to work or school, and their schedule being derailed if they travel to other cities or provinces), high chance of mini-buses being operated by drivers who do not own a license, violence associated with the transport mode, sexual harassment, drivers breaking laws, and in the case of buses, an infrequent bus service during peak hours.

Findings obtained in the National Household Travel Survey (NHTS) study conducted in 2014, which assessed the degree to which public transport services are offered shows us that “only 30% of households in South Africa own a car with the other 70% depending on taxis, buses, trains, and other non-motorised transport modes.” (Mtizi, 2008)

# Related Systems Analysis

1.4.1 Waze Carpool

* Platform: Android, IOS
* Description:

Provides a platform for car-poolers choose riders & drivers based on profiles, star ratings, distance, and price. Filter by gender, co-workers, or custom groups to find their best match and schedule a trip

* Features I would like to incorporate:

Waze Carpool offers free tools to kick start a carpooling group for workers in their company.

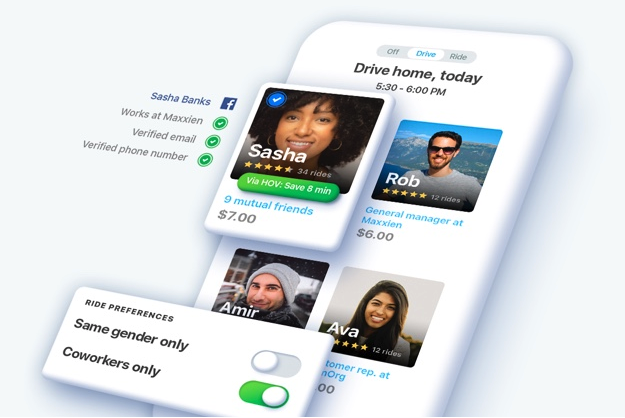


Figure 1.4.1 Waze Carpooling Ride Search PageCar Trip

1.4.2 Car Trip

* Platform: Web, Android, IOS
* Description:

Allows users to find a rider or ride offer, message the driver to make the arrangement, pay the driver on the Car Trip platform, and review the driver

* Features I would like to incorporate:

Car Trip allows users to not only search for riders traveling but also riders going to events. It provides an event search function for trips that are going to these events.

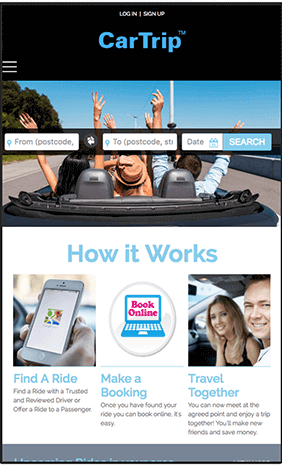


Figure 1.4.2 CarTrip website page explaining how the system works

1.4.3 Jrney

* Platform: Android, IOS
* Description:

Allows users to find a rider or ride offer, message the driver to make the arrangement, pay the driver on the CarTrip platform, and review the driver

* Features I would like to incorporate:

Alerts and notification making it easier for passengers who are using the service to stay updated about the arrangements of their trip



Figure 1.4.3 "Ride Now" service of the Jrney App

1.4.4 LiftShare

* Platform: Web
* Description:

Allows users to create user profiles and arrange lifts as well as create liftshare teams for their trips

* Features I would like to incorporate:  
  Corporate Car Sharing feature



Figure 1.4.4 Home page of the liftshare web application

# 1.5 References

Jacobs, D. (2016, July 3). Urban vs Suburban living costs in SA. (S. Writer, Interviewer)

Mayo, E., & Maringolo, D. D. (1994). *First Ladies political role and image.* Washington, DC: Smithsonian Union.

Oxford, T. (2013, October 04). *Mail & Guardian: Africa's Best Read.* Retrieved from https://mg.co.za/: https://mg.co.za/article/2013-10-04-00-the-state-of-sas-public-transport

Writer, S. (2019, March 4). *Here is the official petrol price for March 2019*. Retrieved from Business Tech : https://businesstech.co.za/news/energy/302978/here-is-the-official-petrol-price-for-march-2019/

System Requirements Specifications

The following section contains the user and system requirements for the carpooling application.

# 2.1 Plan to address objectives

The following objectives were stated in the project business case:

* + 1. Allow users to post trips
    2. Allow users to offer lifts
    3. Allow users to request lifts
    4. Allow users to message each other to exchange contact details and discuss travel arrangements
    5. Allow users to rate users that they have shared a trip with

In most cases when an individual needs to take a trip to a different city or find someone for a lift to work, they need to make a post on social media asking if there is anyone in their followers/friends list that is travelling to their destination (E.g. “Anyone travelling from Bloem to Jeffreys Bay area either on Thursday 18 April or Friday 19 April?” – Facebook User) or make an ad on Gumtree. A web-based application will allow users to give their destination details and find available lifts travelling to their destination. This will allow users to easily search for trips and if at the current time there are no trips, the system will allow the users to be notified once a matching trip has been posted.

In a similar fashion to how people enquire for lifts on social media or Gumtree, the application will provide a means for car owners to offer lifts to passengers, and passengers to request lifts from car owners.

The application can be divided into two parts. The first being trips where car owners can post their trips, receive requests and/or make offers to passengers going the same way, as well as passengers posting lift requests and receiving offers and/or making requests to car owners travelling the same way. The second part being frequent trips that occur on a weekly basis (a user who commutes to work for example).

Once a user receives a request or offer for a trip the user may choose to accept or decline. If the user choses to accept the application should cater for users to message one another to exchange their contact details and details regarding the trip.

# Project Scope

## 2.2.1 Information Scope

* Private information will be hidden from unauthorized users. Unauthorized users will not be allowed to access personal information and passwords. User passwords will be encrypted on the database for extra protection.
* Contact information will be made available only with user’s permission, users will have full control of what contact details to show other user’s and what information they provide to users they are communicating with.
* All user information, as well as communication between users, will be stored on the system database. The applications administrator will have access to all information stored in the database as well as the history of communication made between users.

## 2.2.2 Functional Scope

* Search for and arrange trips:
  + The application will allow users looking for lifts to search if any drivers are going travelling to their destination. The user will be required to enter details such as the date of the trip, their current city and the destination. Their details will be matched to a driver’s trip. The user will be able to see the price of the trip by the driver, the driver’s ratings and reviews, as well as the drivers' contact details. If the user would like to request the trip, they will be able to communicate with the driver and discuss the trip details such as meeting place, cost etc.
  + A driver is any user that owns a car.
  + A passenger is any user that would like to join a driver in a trip he/she posted and agrees to conditions specified by the driver.
* Requesting and offering matching trips:
  + When a user finds a matching trip, the user should be able to request to share (as a passenger) or make an offer to share (as a driver) the trip.
  + As a passenger: The user will need to specify how many seats they are requesting, and the cost should be incremented if the user requests more than one seat to make up for the costs that would have been contributed from the other users.
  + As a driver: The user will make an offer to the passenger (1 seat)
  + Users will be notified of requests or offers to their posts and have the option to accept or decline.
* Communication between users:
  + Users will be able to message one another on the application and discuss details of their trip
  + Users will be able to send messages prior to trips being accepted in cases where users have enquiries about the trip details
* Rating users:
  + Users will be able to rate other users once the trip is past.
  + This will allow fellow users on the application to get a better idea of what type of passenger or driver the user is.

## 2.2.3 Communication Scope

The application will communicate with users via online messaging provided. All users will be provided with notification section in which they can view messages sent to them by users and communicate with them through the online messaging platform provided.

# 2.3 Business Requirements

Passengers:

* Passengers require an application that will allow them to create a profile and edit their profile details. They should be able to set and update their preferences (smoking/no smoking, pets/no pets).
* The system should have a platform in which they can search for drivers driving to their destination. The passengers should be able to enter their trip details which will be matched to the details of drivers offering lifts.
* The application should allow passengers to choose a reliable driver and allow them to judge their reliability based on the ratings and the reviews that drivers have received from users.

Drivers:

* Drivers require an application that allows them to create a profile and advertise lifts.
* The application will allow drivers to create a profile and edit their profile details. They should be able to set and update their preferences (smoking/no smoking, pets/no pets).
* The application should have a platform allowing them to make offers to users needing lifts and allow them to offer lifts and communicate with users once a user requests the lift from the drivers’ lift offer.
* The drivers should be able to see reviews of passengers that have been made by drivers which will give them insight into what kind of conduct and etiquette the rider has.

# 2.4 Hardware and Software Requirements

## 2.4.1 Software Requirements

* The application will be web-based
* The application will be developed using Visual Studio Code
* The application will have a Model-View-Controller architecture and use PHP, Javascript, MySQL technologies and AJAX language for software development
* phpMyAdmin will be used to handle the administration of MySQL over the web
* The application will require any phone that has access and able to connect to the internet
* The application will run in any browser
* The application should be lightweight so that it can work on almost any phone even with slow internet connections

## 2.4.2 Hardware Requirements

* To use the web application on mobile devices users will need to have a device that has access to the internet and a web browser
* The web application will be accessible on any device that has a browser

# 2.5 Design Constraints

## 2.5.1 Security Constraints

* Users will need to create profiles and provide the necessary details and credentials about themselves. They will be logging into their accounts using email addresses and passwords therefore care must be taken to encrypt their passwords on both the database and the system. Unauthorized users should not have access to their profiles.
* Users will be communicating with other users through the systems services. These will need to be kept private and the chats with the users will need to be only accessible by the users interacting within the chat.
* Access to personal user information such as user ride history (both passengers and drivers) should be kept private and not accessible by all users

## 2.5.2 Interface Constraints

* The user interface will need to be of responsive design so that users can use the system on both mobile devices and desktop computers.
* The system should be user-friendly and friendly to use so that even the most novice users do not have trouble interacting with the system.

## 2.5.3 Performance Constraints

* The system should respond fast and efficiently when users search for rides.
* The system should allow users to make searches categorically for the type of carpooling rides the user is looking for such as rides for festivals, university, work lift clubs, sports events and basic city to city, etc.
* System status notifications along every process should be displayed to users as well as validation when they enter details into the system.
* User reviews and ratings should be accurately displayed to users to show the reliability of the users they are interacting with and to add an element of safety.

# 

# 2.6 High-level use case diagram

**Figure 2.6.1** presents a high-level use case diagram of the proposed system and its users.

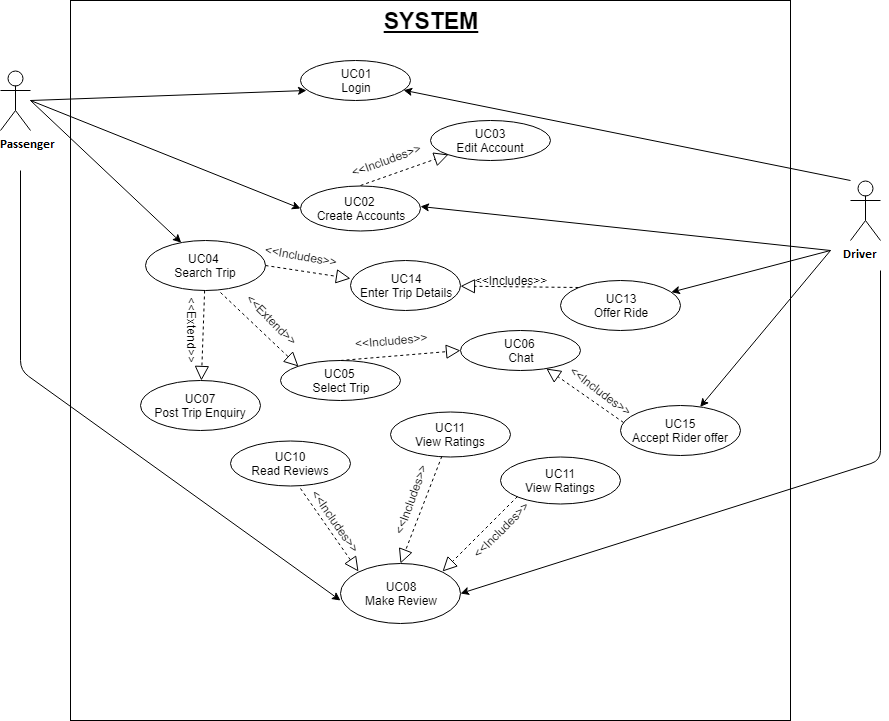


Figure 2.6.1 High Level Use Case

Technical Design

The following section includes the technical design of the application. Although it is not yet the final version of the technical design, it provides the initial class diagram, entity-relationship diagram, as well as a low fidelity prototype of the application.

# 3.1 UML Class Model / Diagram

**Figure 3.1.1** represents the UML class diagram for the Hikeshare system.

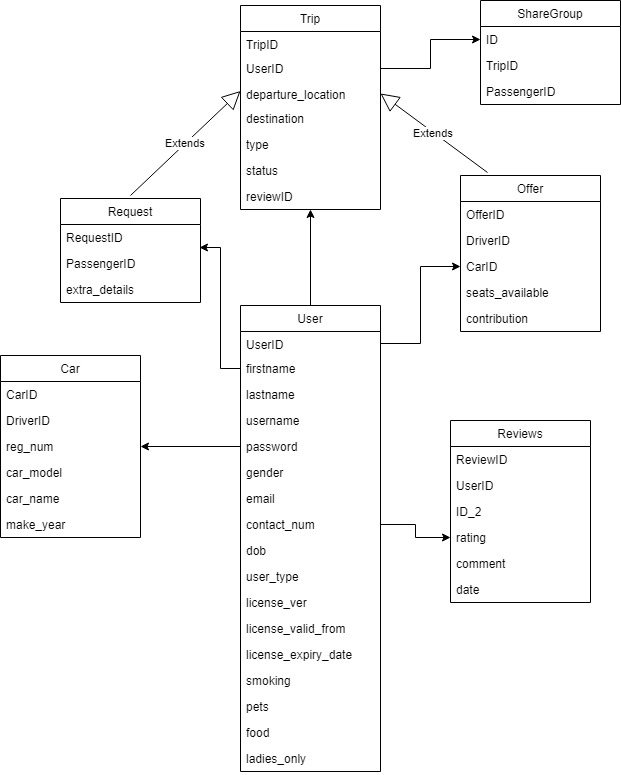


Figure 3.1.1 UML Class Model/Diagram

# 3.2 Initial Relational Database Model Diagram

**Figure 3.2.1** represents the entity relationship diagram for the Hikeshare system.

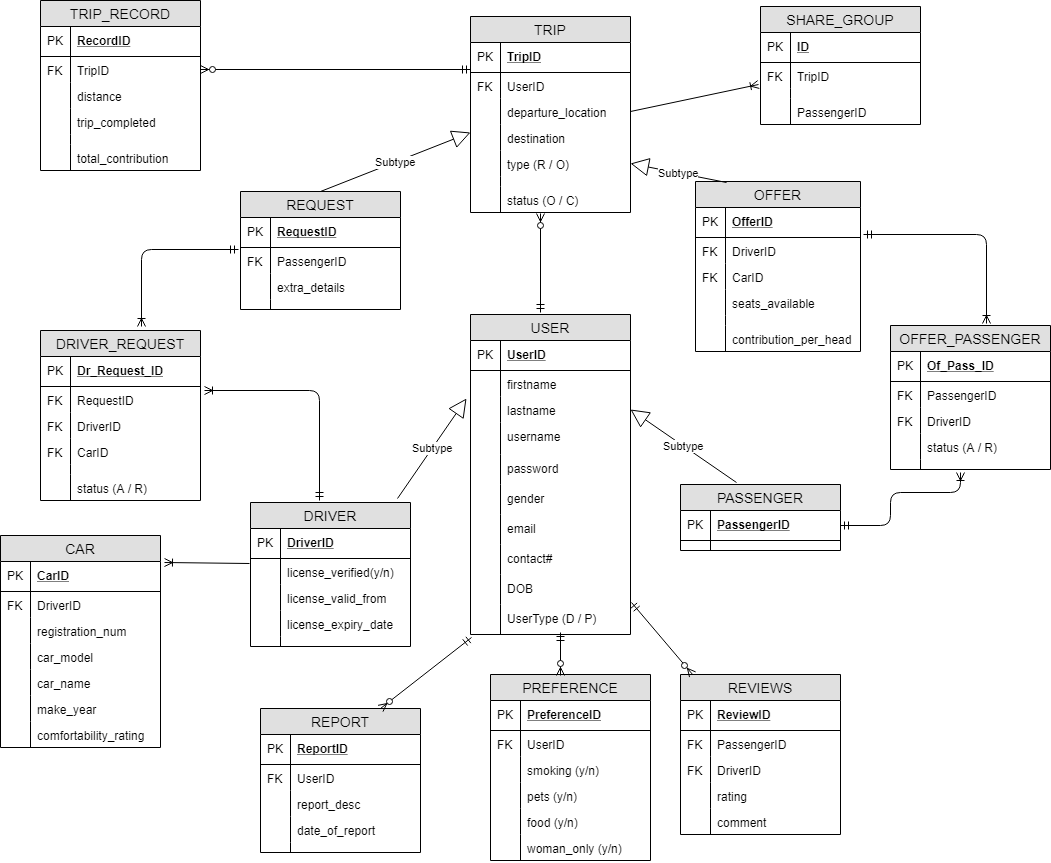
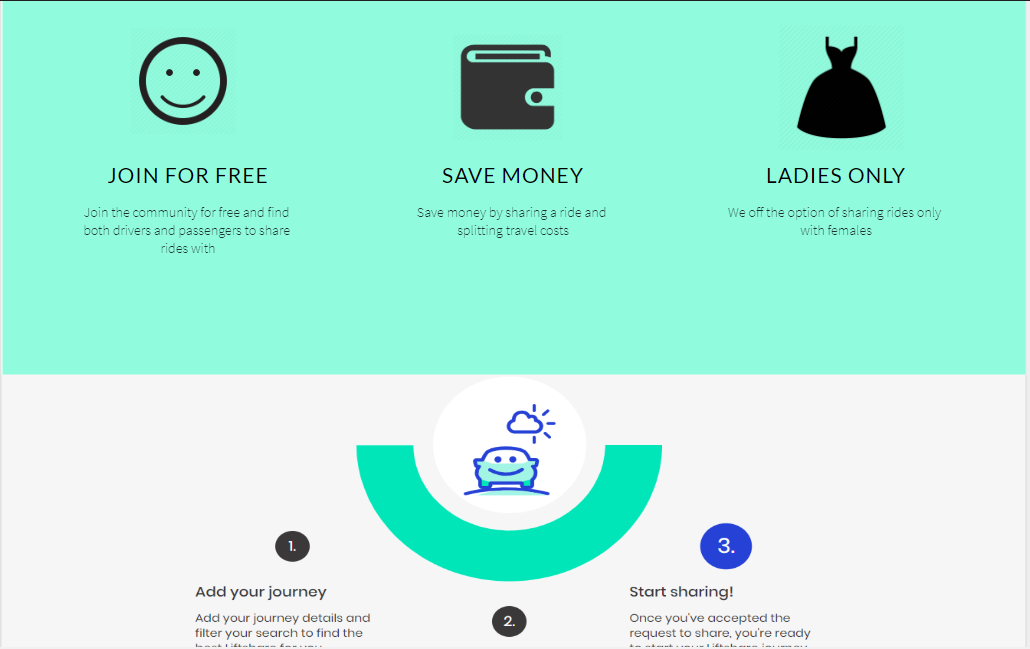
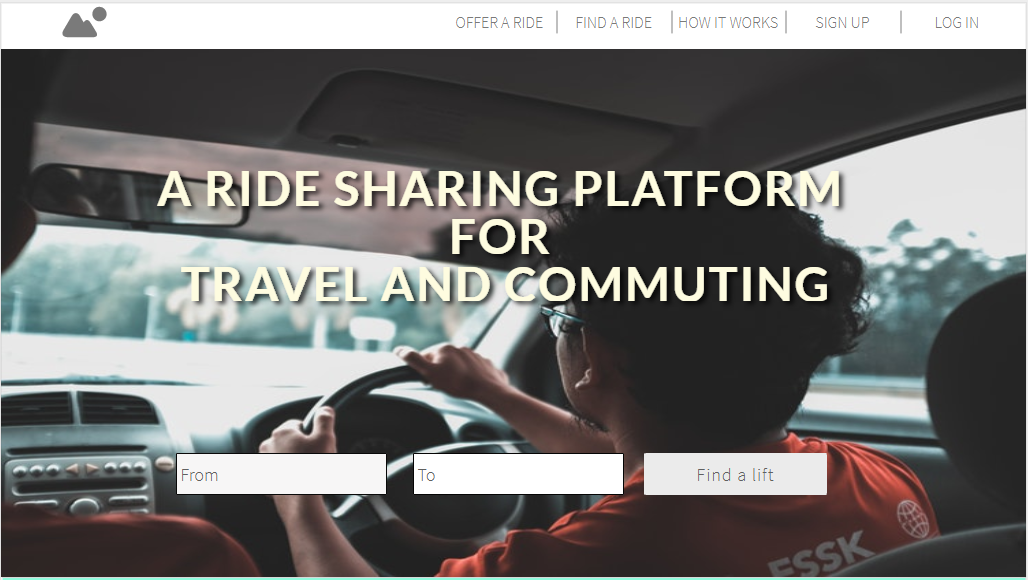
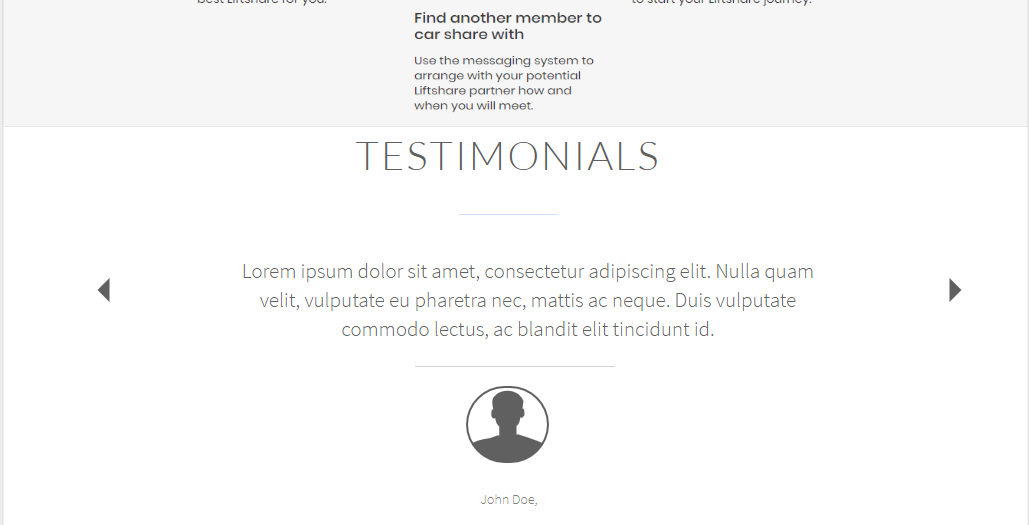


Figure 3.2.1 Entity Relationship Diagram

# 3.3 User Interface Design

**Figure 3.3.1** represents the home page of the application. The screen gives a brief introduction and describes how the application works, some of the features that come with the application and how using the application can benefit the user.





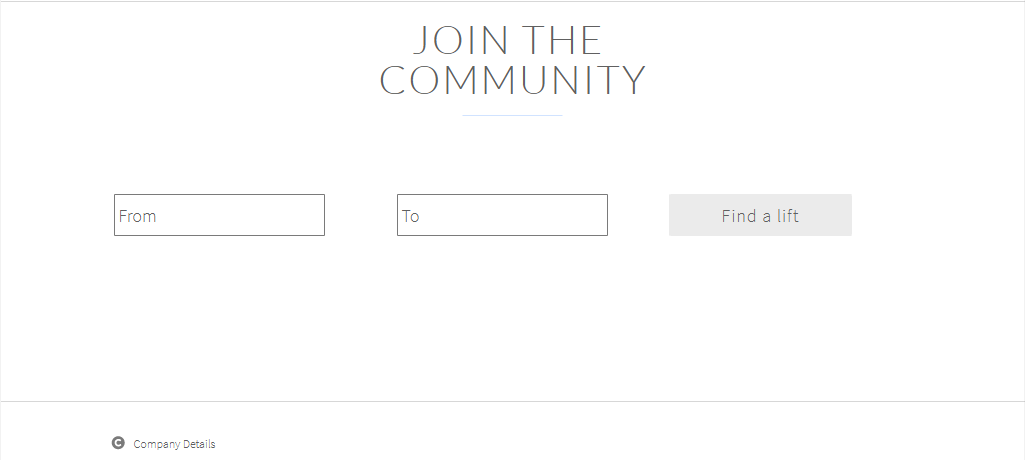


Figure 3.3.1 Landing page

**Figure 3.3.2** represents the login page of the application. It provides the option to login using the user’s social media accounts or their google account. It also gives the user the option to login using the user profile they created using the system. Once a user successfully logged in the user will be redirected to their profile.

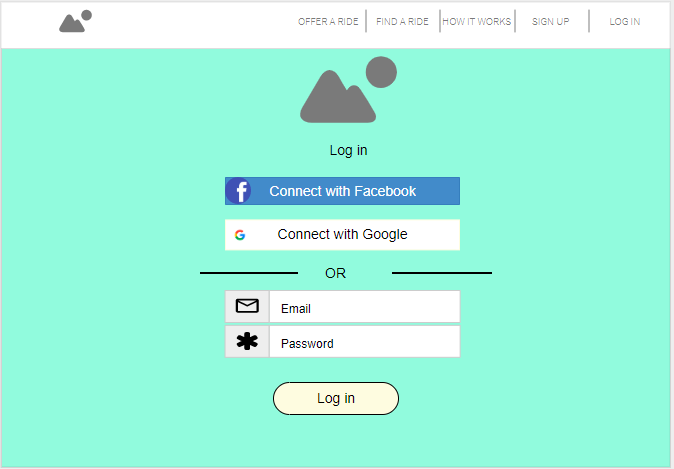


Figure 3.3.2 Login page

**Figure 3.3.3** represents the user profile dashboard.

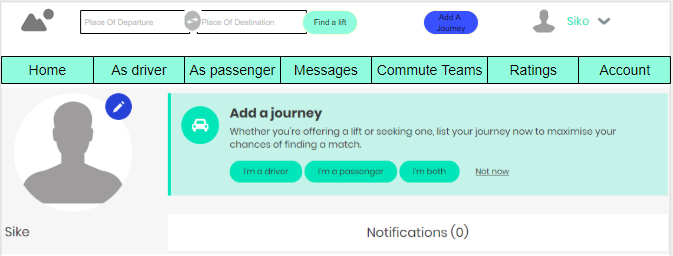


Figure 3.3.3 User Profile Dashboard

**Figure 3.3.4** represents the offer lift page as a driver. The user will be required to enter the location of departure and their destination, as well as the date they would like to leave. They can then search for rides that match the information entered and make requests for rides they feel the most comfortable with.

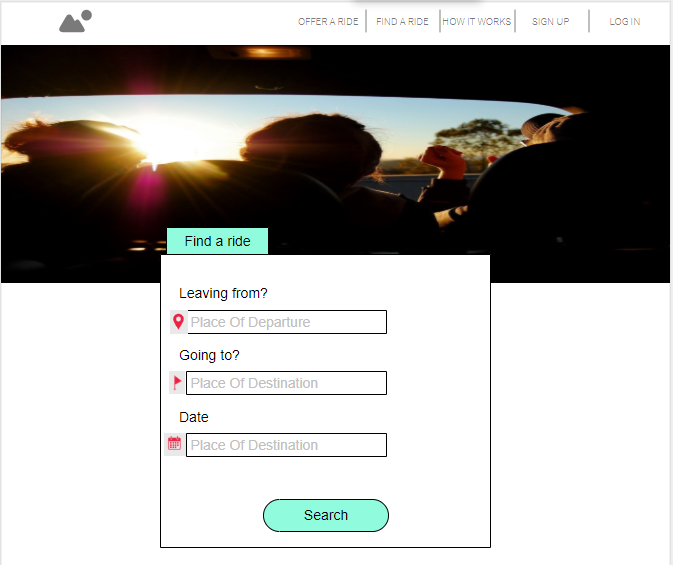


Figure 3.3.4 Find a ride page

# Implementation

The following document will outline the implementation aspects of the system which is being developed. It will take a look at the choice of tools used to build the system, an updated version of the relational database model diagram, an extract of complex code, code referencing as well as a short summary of a problem I faced while developing the system, what was done to overcome it and what was learned in the process.

* 1. **Choice of Tools**

In this section I aim to describe the tools used to develop the application and why the specific choices were made. The programming language I have chosen to use is PHP as the system will be a web-based system. The programming environment chosen is Visual Studio and for the database I have used MySQL.

* + 1. **Visual Studio Code**

Visual studio is an editor that combines simplicity of a source code editor with powerful developer tooling such as IntelliSense code completion and debugging. The editor is lightweight, cross-platform, free and open-source. The well thought GUI makes it easy to navigate and make use of the developer tooling that it provides through extension installations. A recurring theme with Visual Studio Code is that features that you want to use seem to always be visible when you need them. This means that you do not have to know the keyboard shortcuts to be an efficient user of the editor. With its powerful IntelliSense feature it informs you which parts of your code are invalid according the set of extensions you have installed. When a user realizes the editor doesn’t support a certain aspect of a language it suggests of what extension to install depending on your current workspace. It takes away the time the user might have spent searching the internet of what they need to install. The editor also supports GIT for version control. This makes saving and keeping your code secure on the platform easier as all the commands are made available to the user and it allows the user to see the full code difference for each file instead of running a git diff and having to remember what was in the diff when using git commit.

Visual Studio Code comes with great features and packages that has made my development more efficient as it supports all the needs, I need in order to build the Hikeshare system.

* + 1. **MySQL**

MySQL is an effective database. It served to be a great database to use along with PHP programming. It is one of the world’s most famous open source databases, has high quality and convenience. In PHP, you can “interface” to the MySQL benefit utilizing PDO to communicate with the database.

Some added benefits of using MySQL include:

1. Data Security
2. On-Demand Scalability
3. High Performance
4. Round-the-clock Uptime
5. Comprehensive Transactional Support
6. Complete Workflow Control
7. Reduced Total Cost of Ownership
   1. **Relational Database Model Diagram**

This section presents the final version of the Hikeshare relational database model.

Figure 4.2.1 represents the relational database model diagram of the Hikeshare system.

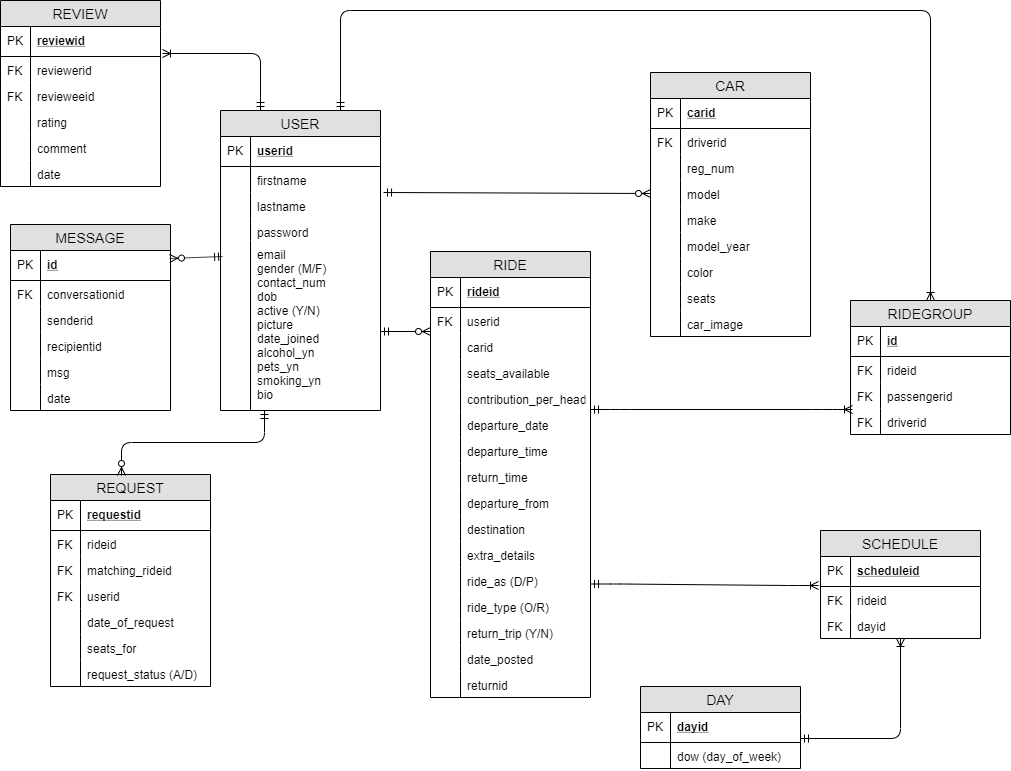


Figure 4.1 Relational Database Model Diagram

* 1. **Extracts of complex code**

This section represents an extract of complex code with the system. The abstract chosen to display was taken from the Bootstrap functionality of the system which is utilized within the systems MVC structure.

Figure 4.3.1 represents an extract of complex of the Hikeshare system.

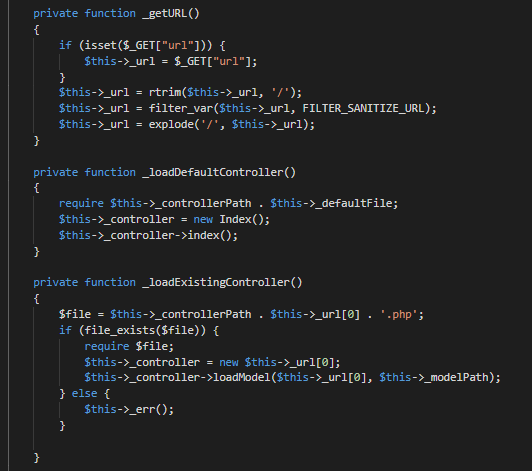


Figure 4.3.1 Methods for getting the url; loading default controller; loading existing controller

The system makes use of a .htaccess file to retrieve the URL and load the corresponding controller based on what is stored in the URL variable. The .htaccess file redirects the URL to the index page and we use a query string to store the page name in the URL variable. The \_getURL method serves the purpose of retrieving the value of the query string, which is then used to load the corresponding controller in the \_loadExistingController method. Within the \_loadExisitingController method we first check if the file exists, if the file exists, we instantiate the controller, or else we throw an error.

* 1. **Source code references (if applicable)** 
     1. **Converting images to byte arrays**

***Description***

You can store an image in its required database table by converting it into the base64 format. In PHP the base64\_encode() method can be used for base64 conversion. Before storing it in the database I appended the data:image/’.$img\_file\_type.’;base64, text with base64 value.

To display the image all we need to do is fetch the value and use it as an image source.

***Usage***

User profile

***Reference***

Singh, Y. (2019, September 9). Upload and Store An Image In The Database With PHP. Retrieved September 16, 2019, from <https://makitweb.com/upload-and-store-an-image-in-the-database-with-php/>

* 1. **Problems encountered**

This section represents reflection on an area in the system I struggled with when it came to an aspect in the development of the project.

* + 1. **Making use of the google maps API**

I had no experience with using google API’s. My aim was to utilize a map and the Google Place Autocomplete API. The general process I was going for was, the user enters their departure address and destination address in the corresponding textboxes, and to have the google api autocomplete to get a accurate address, and have the route and markers displayed on the map once the user has entered their addresses as well as to change the route if the user decides to enter a different request. Implementing the autocomplete API was simple and straight forward but what I struggled on was using the map and displaying the routes between the two points.

Eventually I discovered that I had to use the Google Maps, Place and the DirectionService API together to do this. I also struggled in changing the code to fit exactly what I was going for but I figured it out and was successful in my objective for the use of these API’s.

Test plan and results

The introduction starts with describing the purpose of this document and what it will cover.

* 1. **Test cases**

This section is used to present the test cases used with the system. Provide test cases for at least 3 system functions. Provide each functional grouping with its own heading, e.g.

**5.1.1 Register Livestock**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PROJECT** | Farm Manager Application | | | | | |
| **DATE CREATED** | 25 / 08 / 2016 | | | | | |
| **FUNCTIONAL SPECIFICATION** | Register Livestock | | | | | |
| **TEST OBJECTIVE** | To determine that the details for the new Livestock or correct and valid | | | | | |
| **USE CASE ID** | UCD 0001 | | **TEST CASE ID** | | TC0001 | |
| **TEST DATA(INPUT)** | **Livestock Name** = “Grizilla” , **Weight** = 20 kg , **Date of Birth** = 20-05-2015, **Herd Name** = “North”, **Breed Name** = Jersey Cow,  **Livestock Type** = Cow | | | | | |
| **Test No.** | **Steps** | **Input Data** | | **Expected Results** | | **Actual Results** |
| 1 | User enters the Livestock Name, Weight, Date of Birth, Herd Name, Breed Name, Livestock Type and Press Submit Button | **Livestock Name** = “ ”,  **Weight** = “ ” , **Date of Birth**  = “ ”, **Herd Name** = “ ”, **Breed Name** = “ ”, **Livestock Type** = “ ” | | Should display Error Message that all fields are required. | | Fail / Pass |
| 2 | User enters the Livestock Name and Press the Submit Button | **Livestock Name =** “Grizilla” | | Display an Error Message that all fields are required. | | Fail / Pass |
| 3 | User enters the Livestock Name, Weight, Date of Birth, Herd Name, Breed Name, Livestock Type and Press Submit Button | **Livestock Name** = “Grizilla”  , **Weight** = “—Select—“ , **Date of Birth** = 20-05-2015, **Herd Name** = “North”, **Breed Name** = Jersey Cow, **Livestock Type** = Cow | | Display error Message that selected Livestock Weight is Invalid | | Fail / Pass |

* 1. **User interface evaluation**

This section presents the user interface evaluation survey. It needs to present the survey as well as the following details:

* Requirement for participant selection, e.g. for a system collecting healthcare data, the requirement for a participant would be someone who works daily on the real-life data collection process.
* Date of evaluation(s): The date and time on which the evaluation took place. At least one is required. If more are done, list all the dates and times.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Questions / Statements** | **Rating** | | | | |
| 1 | Use of the system allowed me to complete the given task/s in an acceptable time. | **1** | **2** | **3** | **4** | **5** |
| **Comments:** | | | | | | |
| 2 | It was easy to perform the given task(s). | **1** | **2** | **3** | **4** | **5** |
| **Comments:** | | | | | | |
| 3 | The system is easy to learn. | **1** | **2** | **3** | **4** | **5** |
| **Comments:** | | | | | | |
| 4 | The system is easy to navigate. | **1** | **2** | **3** | **4** | **5** |
| **Comments:** | | | | | | |
| 5 | The system is stable and reliable. | **1** | **2** | **3** | **4** | **5** |
| **Comments:** | | | | | | |
| 6 | The screen layout allows me to quickly accomplish my goals. | **1** | **2** | **3** | **4** | **5** |
| **Comments:** | | | | | | |
| 7 | The system allows me to gracefully recover from errors. | **1** | **2** | **3** | **4** | **5** |
| **Comments:** | | | | | | |
| 8 | Validation guides me towards entering the correct type of information. | **1** | **2** | **3** | **4** | **5** |
| **Comments:** | | | | | | |
| 9 | Clear feedback is given on system actions. | **1** | **2** | **3** | **4** | **5** |
| **Comments:** | | | | | | |
| 10 | Help is easy to find. | **1** | **2** | **3** | **4** | **5** |
| **Comments:** | | | | | | |

* 1. **Changes implemented**

**­**

This section is used to present any changes made to the system as a result of suggestions / comments or your own insights after working through the test cases and user interface evaluation, e.g.

The users found it difficult to find help on the Add Transaction screen. To address this, a Help icon was added next to the page title. It was also difficult for the user to find the correct entry in a grid. A general search was added to allow the user to search across Name, Surname and ID number with a single query.

Conclusion

**6 Study conclusions**

Use this section to conclude your study / project. It should be at least half a page in length, but no longer than two. Touch on what the project set out to do, what the study accomplished and any insights you might have gained or ways you want to improve / expand upon the project in future.

# Test Plan and Results Appendix

Attach scans (or decent smartphone photographs if you don’t have access to a scanner) of the completed user interface evaluation form(s).