INTELLIGENT COMPLEMENTARY RIDE-SHARING SYSTEM

Project ID: CDAP 19-055

Software Requirements Specification

B.Sc. Special (Honors) Degree in Information Technology
Specializing in Software Engineering

Submitted on: 29th of April 2019

Declaration

I hereby declare that this is my own work and this document does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Name	Registration Number	Signature
V. A. Wickramasinghe	IT16030190	
A. E. Edirisinghe	IT16025936	
G.L.S.R.Gunawardena	IT16011380	
R.M.A.N.Gunathilake	IT16033474	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Supervisor Date

…………………………… ……………………………

Dr. Janaka Wijekoon

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1. Introduction

1.1 Purpose

In this document is to provide a detailed description of the process and the requirements of the "+Go - Intelligent complementary ridesharing system". “+Go” Application is developing for the minimize traffic in an urban area in peak hours. In here mainly discussed the features, behaviors that includes what the functionality of the system and how achieving the goals and target audience. In addition to that, the document will discuss the workflow of the individual components and the system with required diagrams, functional and non-functional requirements, design constraints and User Interfaces. Moreover, this document illustrates user roles who interact with the final product and what the functionalities according to the user level. Therefore, stakeholders can get a proper idea about the software aspect of the component as well as it will be beneficial for the development team in future references.

1.2 Scope

This research project initiates to develop a beneficial solution to minimize the traffic congestion during office hours in urban areas. Therefore +Go is the ride sharing application that we proposed to minimize this problem. In here mainly +Go platform is targeting on office crowd in Sri Lanka .Because majority of the people are working in highly congested areas like Colombo area.

In the solution proposed, we came up with four research components to minimize the traffic congestion while providing travelling facilities to the users.

- User Profile Management
    - ❑ User Profiling
    - ❑ Document Validation and User rating
- Optimum Path Recognition
- Cost Calculation

When considering User Profile Management component, it will helps to identify the most suitable list of drivers for the passengers by matching the profiles of the passengers to the drivers and vice versa. Personal interests, trajectory details, driver ratings, gender preference, profession, vehicle type, time and date like specified factors are considered when filtering most related drivers from a driver pool to passengers.

Document validation is done by using computer vision methodologies to avoid fake registrations by identifying and checking the compatibility of driving license and National Identity Card (NIC).User rating will be helpful before a ride, since it allows the driver to rate the ride with the passenger and passenger to rate on driver, vehicle and other commuters.

In Optimum Path Recognition component, it will identify the closest path with least traffic which connect source and destination while minimizing the traffic congestion. In here, it uses crowdsourcing platforms to increase the accuracy of the suggested routes by reporting the details with images of the particular incident as there's any accidents, traffic jam or etc.

Fare calculation for the ride have to be shared among the passengers including the driver. Fuel consumption of the vehicle, distance, waiting time, No of current passengers and traffic condition are considered as the parameters for calculating the cost. Therefore this component is considered as the business logic of this proposed system.

According to the above mentioned research components, some significant advantages of our proposed solutions are mentioned as follows.

- Reduce the traffic congestion.
- Build the network among Professionals.
- Reduce the travelling cost.
- Ensure the security of the passengers.
- Minimize the environmental pollution.
- Helps to reduce the stress and improve productivity while travelling as a passenger.
- Compensation fee on both driver and passenger for any delay other than specified waiting time.

1.3 Definitions and Abbreviations

SRS	**Software Requirements Specification**
ICRSS	**Intelligent Complementary Ride Sharing System**
UPM	**User Profiling Management**
DVPRM	**Document Validation and Profile Rating Maintenance**
OPR	**Optimum Path Recognition**
FC	**Fare Calculation**
UI	**User Interface**
OBD	**On-board diagnostics**
OS	**Operating System**
API	**Application Programming Interface**
GPS	**Global Positioning System**
NIC	**National Identity Card**
RAM	**Random Access Memory**

1.4 Overview

Traffic congestion is a major concern which has drawn the attention of society. The major problem is the number of vehicles daily entering the urban areas a high [1]. Most of the time, many vehicles are coming from the same area to defined destination resulting the vehicles are underutilized. To this end, this study proposes a solution to minimize this problem by implementing a ride-sharing platform: **+Go**. An initial study of the **+Go** platform is basically targeting the office staff in Sri Lanka because the majority of the offices are operating in highly congested urban areas [3].

+Go ride-sharing platform matches the profiles of the passengers to the drivers, and vice versa, then suggests the rides using trajectory details, gender preference, personal interests, profession, rating, time and date as the parameters.

Furthermore, the **+Go** platform uses computer vision methodologies to avoid fake registrations, and nonetheless, it uses crowdsourcing platforms to increase the accuracy of the suggested routes. The passengers, including the driver, have to share the cost for the trip and that is intelligently calculated according to fuel consumption of the vehicle, distance, the time taken to arrive at the destination, are taken as the parameters for it. Further, to improve the experience of the user, the **+Go** system maintains a rating system which allows the driver to rate the ride with the passenger, and the passenger to rate on the driver, vehicle and the co-passengers.

The first part of the document brings you a comprehensive detail about all the component of the proposed application UPM, DVPRM, OPR, and FC. In the second section of the document, it describes each and every individual component used in the application and their relevant interfaces. Both the functional and non-functional perspectives are thoroughly covered in this document. At the latter half of the document, some non-functional attributes like reliability, maintainability, security, and availability of the software solution are further described.

2. Overall Descriptions

This whole document describes the main goals and how they are archived, the connectivity between other modules, all the interfaces required during the implementation etc. are comprehensively discussed in this document. Main goals, the way of achieving those and the effect of them to the stakeholders are described here.

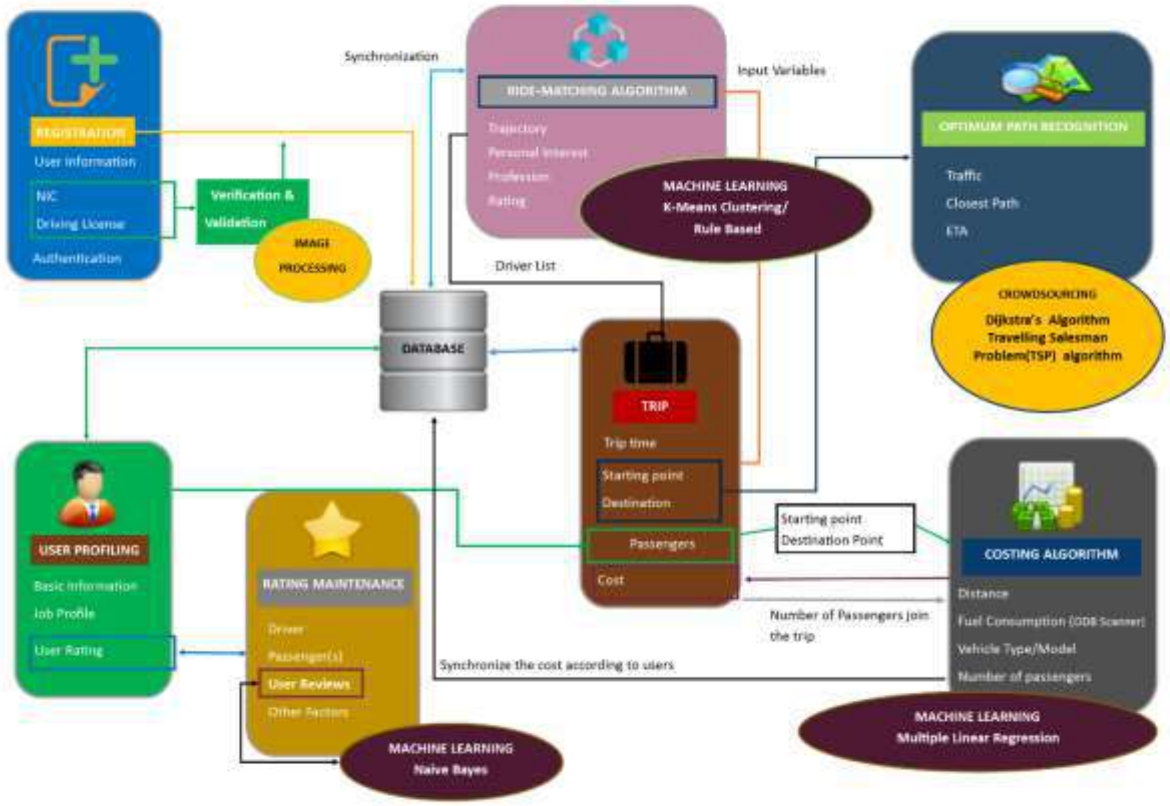


Image 2.0.1 – High Level System Diagram

In the registration phase, the user is verified by the image identification of the National Identity Card (NIC) and then with the License card.. Then, all the basic information including name, address etc. is collected and personal interests and information of spouse/guardian is taken. To finalize the registration, he/she needs to verify the phone number and add a payment method to create a successful account.

User Profiling is the mechanism used to divide each user of the system based on basic information, job profile, interests, and ratings. It always communicates with the rating class to keep a track on

the ratings and passenger details are taken from the trip class. All the results will be sent to the database and will be used to retrieve the output from an algorithm.
Ride Matching algorithm always synchronizes with the database to keep a track on the latest data to be processed. Basic inputs to the algorithm will be the personal details, trajectory, time, date, personal interests, ratings and more importantly, the profession.

At the completion of the trip, the passenger will be asked to rate his/her experience. Then the user can simply rate with 5 is everything is good; the Driver, Vehicle and the Co-Passengers will get the default rating 5. If the rating is below five, the passenger will be asked to specify which made them the journey uncomfortable. Further, they will be allowed to write their own review as well and the system will identify the user experience accordingly. Drivers too will get the chance to rate the passengers at the time they get down from the vehicle.

In optimum path recognition, we will be identifying the closest path with the least traffic. This helps to minimize traffic congestion in urban areas. When users set their locations, our supposed system will identify the order of the locations to visit. Then an algorithm will predict the optimal path, which travels, through each location (node) in the listed user-specified locations. In here, registered users will be able to enter the live updates regarding the relevant path within a specified time range by uploading the pictures of the particular incident.

Trip class mainly handles the data for each trip. A trip is subdivided into several segments depending on the number of passengers traveling at that time. If a new passenger joins the trip or a certain passenger finishes the trip, a new segment is created. Each Trip segment stores segment start time, start location and passengers details (Passenger's name, Current cost of the passenger). In addition to that, the trip class store overall trip started time (First segment start time), trip end time and details of the driver of the trip.

The estimated cost on the trips will be based upon fuel consumption of the vehicle, distance, waiting time, No of current passengers and traffic condition. We will be taking the readings from the odb2 scanner to take the actual fuel consumption in the scenario after the cost is calculated, all the data will be synchronized with the database. To formulate the final cost, we consider a

Passenger travelled distance, cost of fuel consumption and waiting costs to bring out a more dynamic cost formula for the users.

2.1 Product Perspective

As per our market research, we identified many existing solutions to the domain that we are addressing. Out of them, many solutions are a failure and does not address the need of people. Therefore, we did a deep study on each of the solutions to identify their features and loopholes, so that we can make our ICRSS a better one for the user. This ICRSS is a combination of features of the existing solutions as well as we have added many new features for the beneficiary of the user and to stand out from the existing solutions.

Features	UDIO	Carpooling.lk	RideShare.lk	Proposed Solution (ICRSS)
System mainly focused on office staff	✗	✗	✗	✓
Matching the passengers' profile with the suitable drivers	✗	✗	✗	✓
Allow the spouse/guardian to check the passenger's trip details	✗	✗	✗	✓
Suggestion of drivers per passenger interests and preferences	✗	✗	✗	✓
Consider gender preference when registering to provide high security	✓	✗	✗	✓
Validating the user by NIC and license by processing the images of them	✗	✗	✗	✓
Analyze the reviews given by users based on their severity and categorizing them	✗	✗	✗	✓
Allowing the passengers to rate the driver, vehicle and co-passengers separately at the end of trip.	✗	✗	✗	✓
The system will decide the estimated cost before joining the trip.	✓	✗	✗	✓

Vehicle fuel cost calculate according to the condition of the vehicle. (Engine Capacity, Manufacture year, Registered year, Mileage etc.)	✗	✗	✗	✓
Passengers can get off in any place where is the between source and destination because the price will calculate passenger travel distance	✗	✗	✗	✓
Crowdsourcing to improve the optimum path by analyzing more than one algorithm.	✗	✗	✗	✓
Allowing the registered users to enter the live updates by uploading images.	✗	✗	✗	✓

Table 2.1.1 - Feature Comparison Table

2.1.1 System Interfaces

As the initial step, ICRSS is run on the android platform and it is done using the Android Studio. For the backend of the application, node server is deployed on the cloud storage and firebase is used for some parts of the application to take the real time data. We have designed several system interfaces pertaining to all four components to have a synchronization between the back end data and front end data.

2.1.2 User Interfaces

Given below are UI used in the UPM.

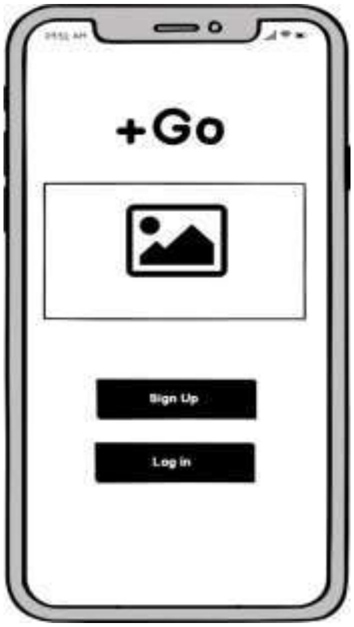


Image 2.1- Home Screen of ICRSS

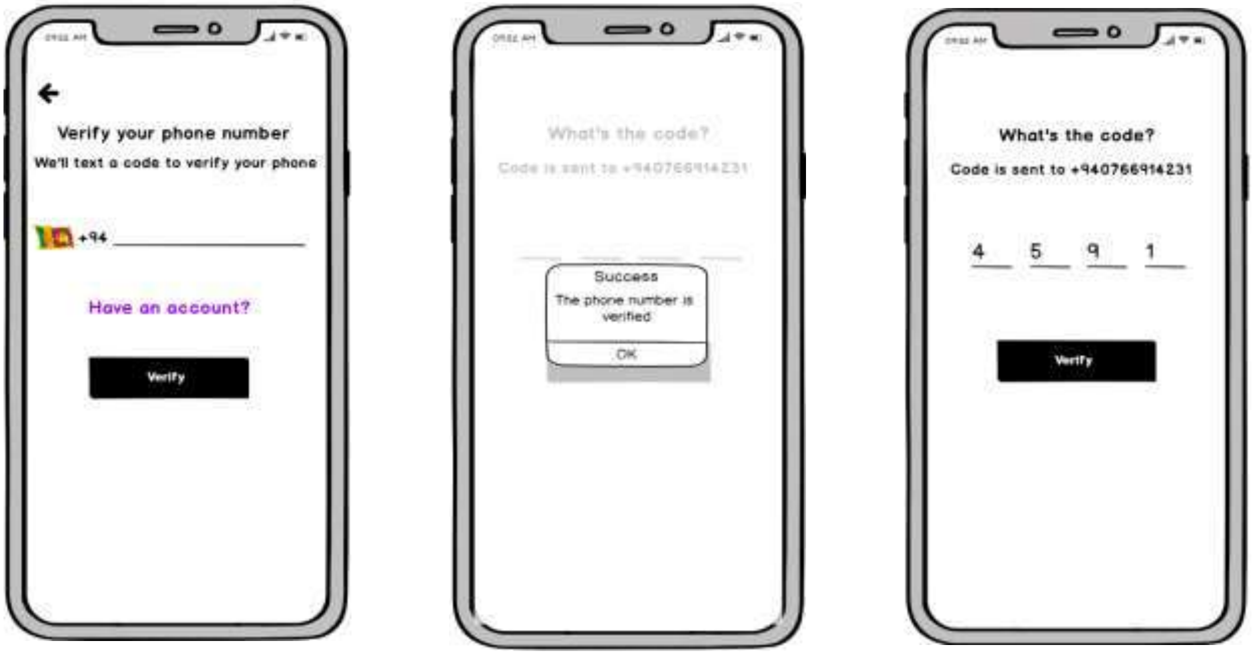

Verify your phone number
We'll text a code to verify your phone
+94
Have an account?
Verify
What's the code?
Code is sent to +940766914231
Success
The phone number is verified
OK
What's the code?
Code is sent to +940766914231
4
5
9
1
Verify

Image 2.2 - Phone Number Verification

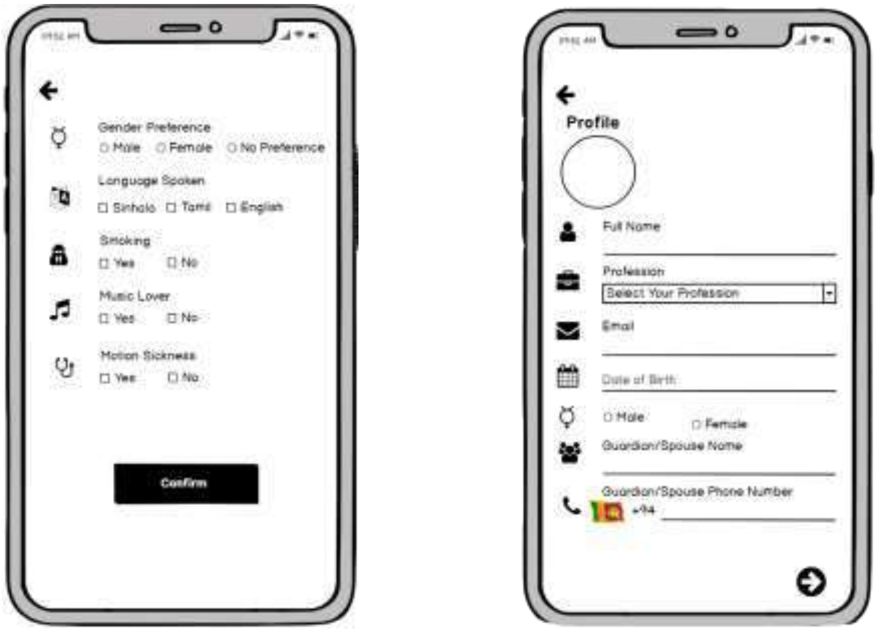


Image 2.3 - User Profile

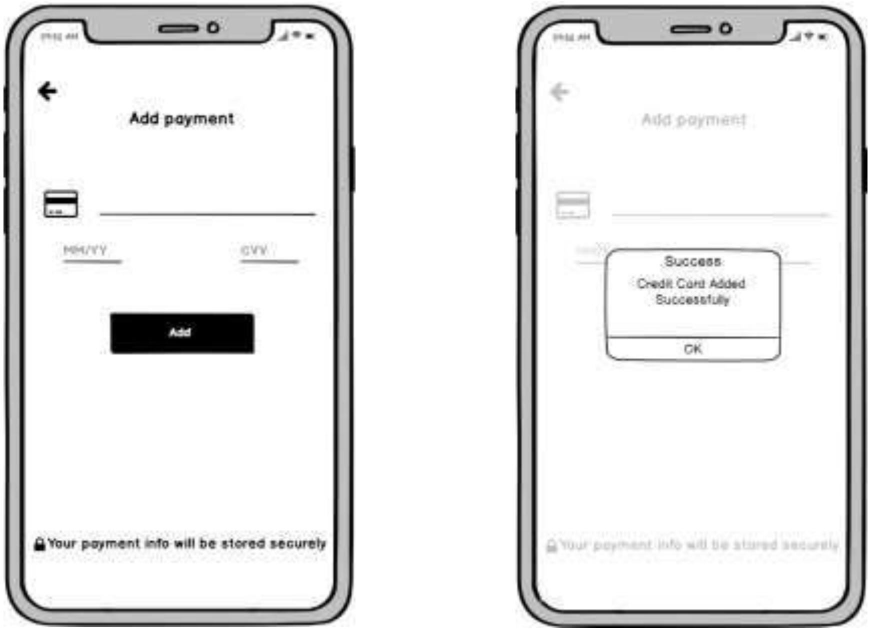

Add payment
MM/YY
CVV
Add
Your payment info will be stored securely
Add payment
Success
Credit Card Added Successfully
OK
Your payment info will be stored securely

Image 2.4 - Adding Bank Details

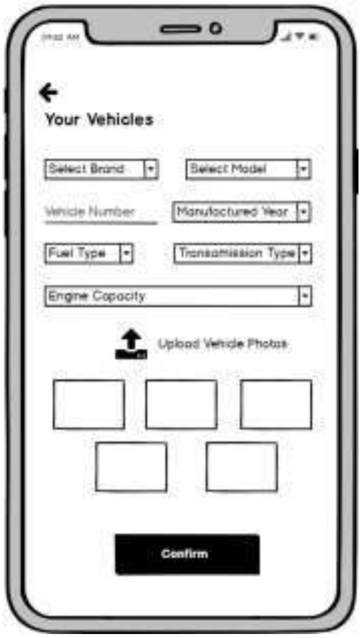
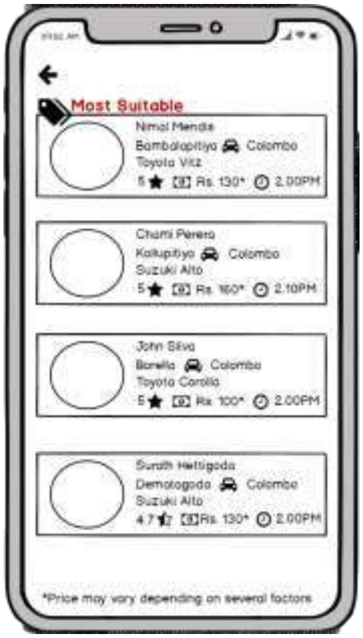


Image 2.5 - Adding Vehicle Details



Nimal Mendis
Bambalapitiya Colombo
Toyota Vitz
5★ Rs. 130* 2.00PM

Chami Perera
Kollupitiya Colombo
Suzuki Alto
5★ Rs. 160* 2.10PM

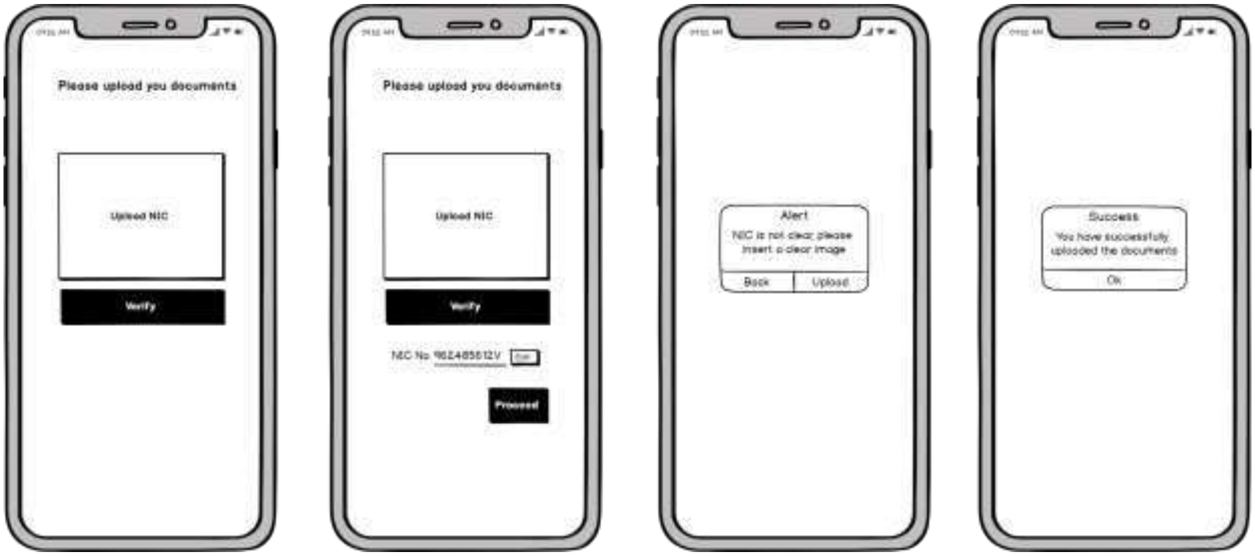
John Silva
Borella Colombo
Toyota Corolla
5★ Rs. 100* 2.00PM

Surath Hettigoda
Dematagoda Colombo
Suzuki Alto
4.7★ Rs. 130* 2.00PM

*Price may vary depending on several factors

Image 2.6 - Suggested Driver List

The main user interfaces for document validation are shown below.



Please upload you documents

Please upload you documents

NIC No 962485612V

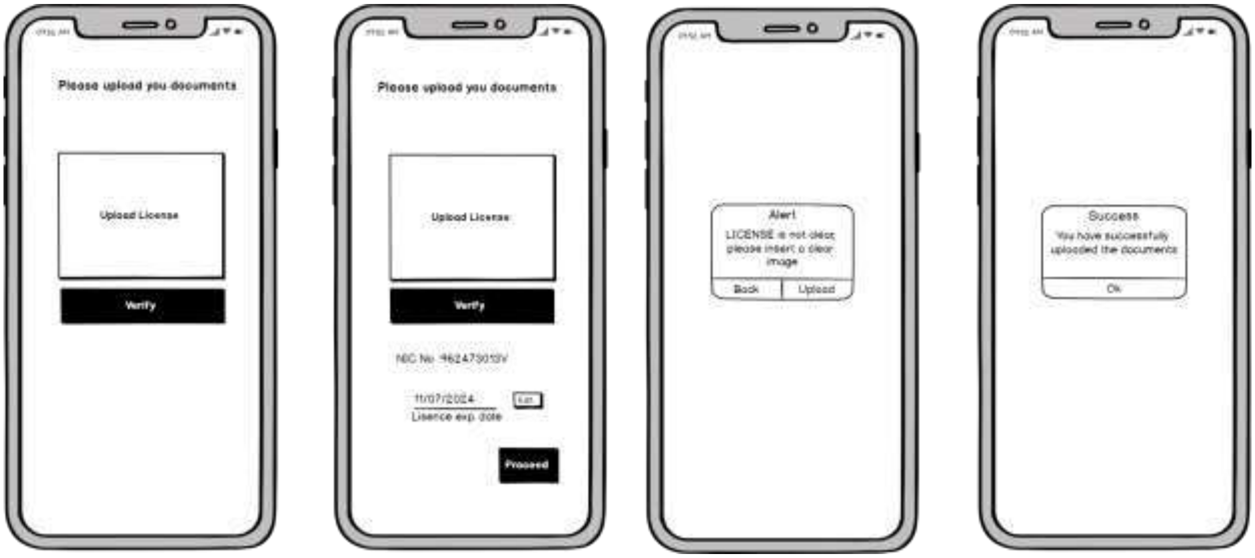
Alert

NIC is not clear, please insert a clear image

Success

You have successfully uploaded the documents

Image 2.7 - NIC upload



Please upload you documents

Please upload you documents

NIC No 962473013V

11/07/2024

Lisence exp date

Alert

LICENSE is not clear, please insert a clear image

Success

You have successfully uploaded the documents

Image 2.8 - License upload

Following are the mobile interfaces designed for rating maintenance

a. Driver Interfaces

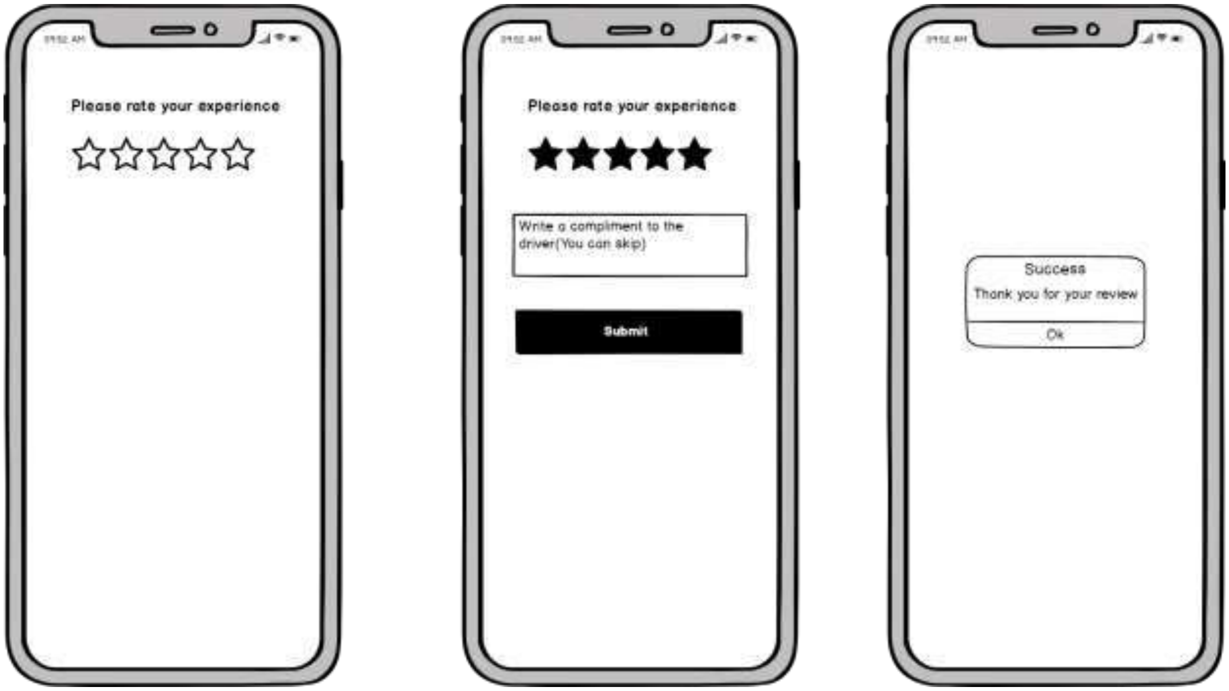

Please rate your experience
Please rate your experience
Write a compliment to the driver(You can skip)
Submit
Success
Thank you for your review
Ok

Image 2.9 - Five-star rating interface by driver

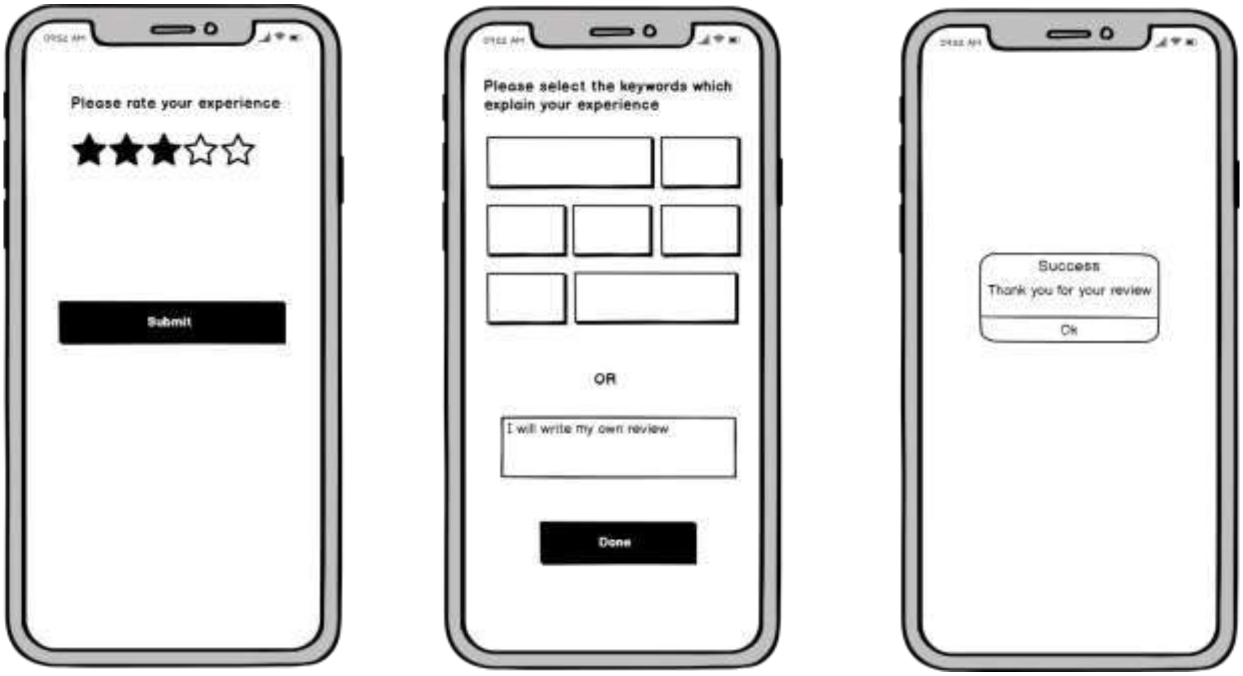

Please rate your experience
Submit
Please select the keywords which explain your experience
OR
I will write my own review
Done
Success
Thank you for your review
Ok

Image 2.10 - Low rating interface by driver

b. Passenger Interfaces

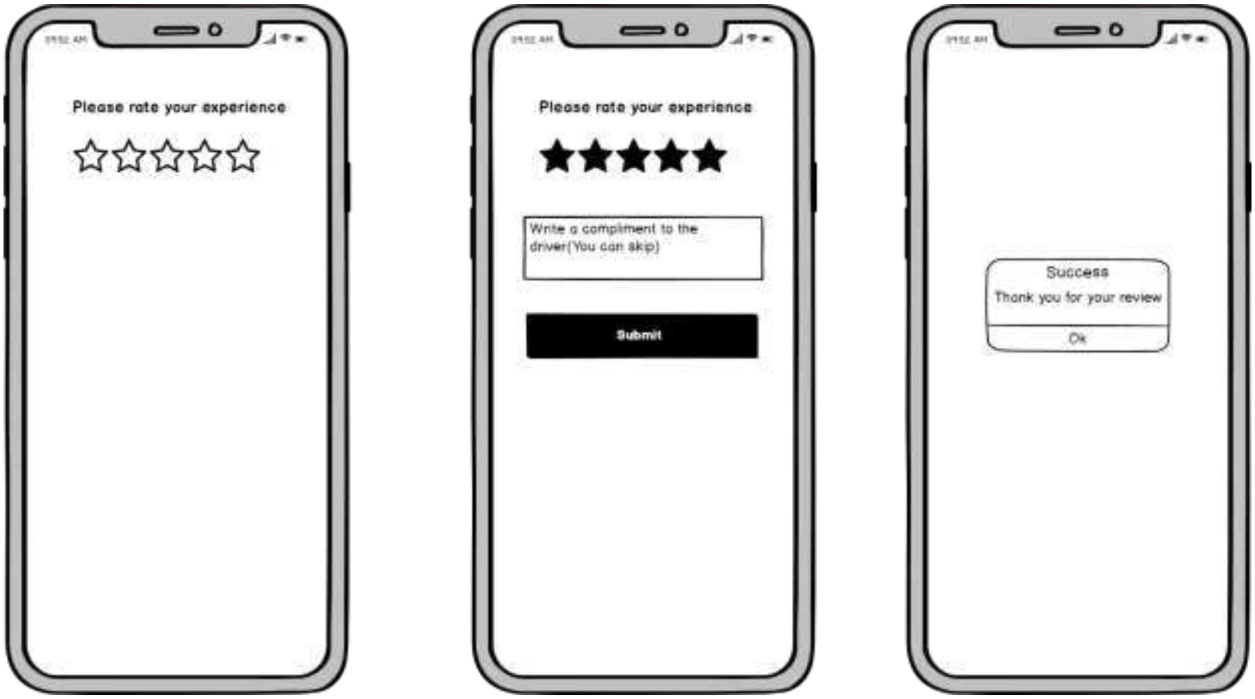

09:52 AM
Please rate your experience
09:52 AM
Please rate your experience
Write a compliment to the driver(You can skip)
Submit
09:52 AM
Success
Thank you for your review
Ok

Image 2.11 - Five-star rating interface by passenger

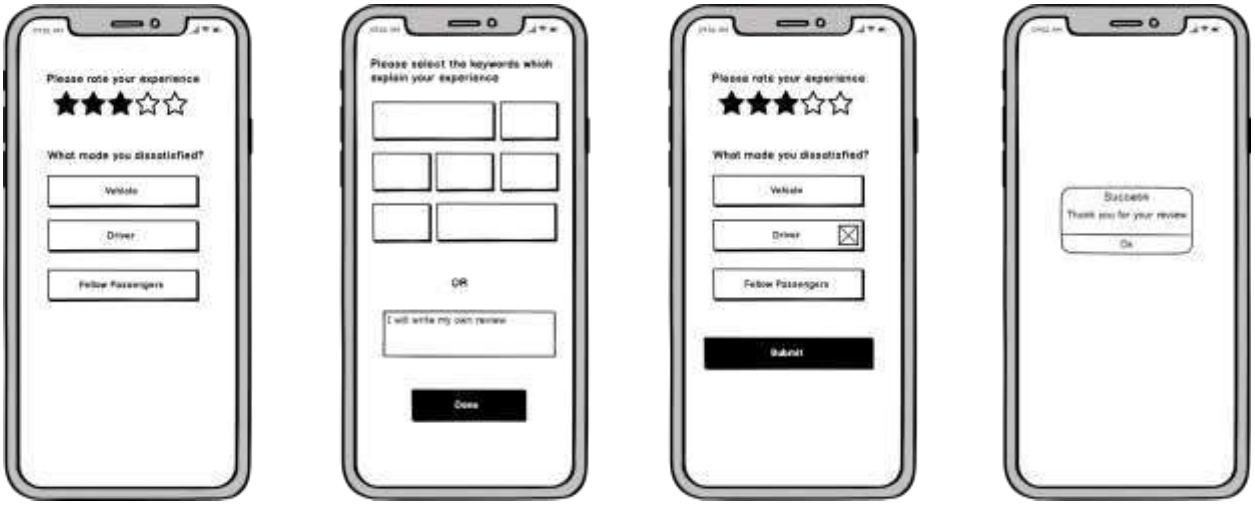
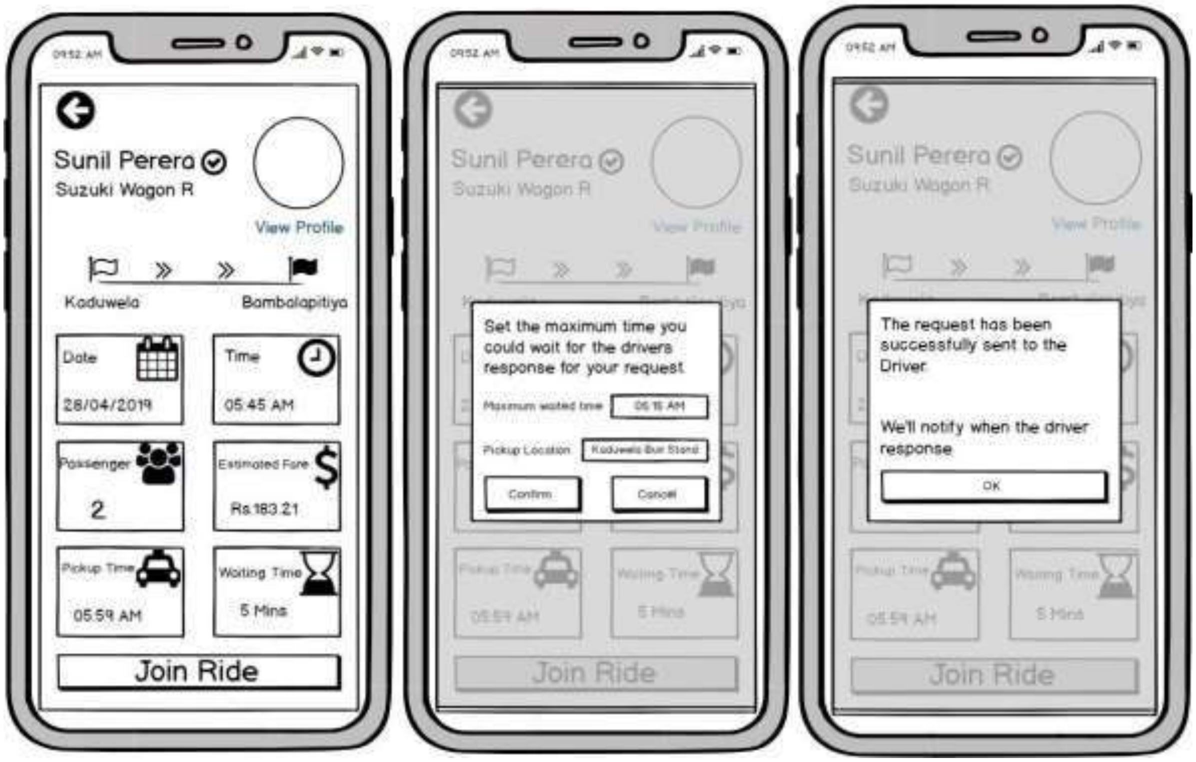

Please rate your experience
What made you dissatisfied?
Vehicle
Driver
Fellow Passengers
Please select the keywords which explain your experience
OR
I will write my own review
Done
Please rate your experience
What made you dissatisfied?
Vehicle
Driver
Fellow Passengers
Submit
Success
Thank you for your review
Ok

Image 2.12 - Low rating interface by passenger



Sunil Perera

Suzuki Wagon R

Kaduwela Bambalapitiya

Date 28/04/2019

Time 05.45 AM

Passenger 2

Estimated Fare Rs.183.21

Pickup Time 05.59 AM

Waiting Time 5 Mins

Set the maximum time you could wait for the drivers response for your request

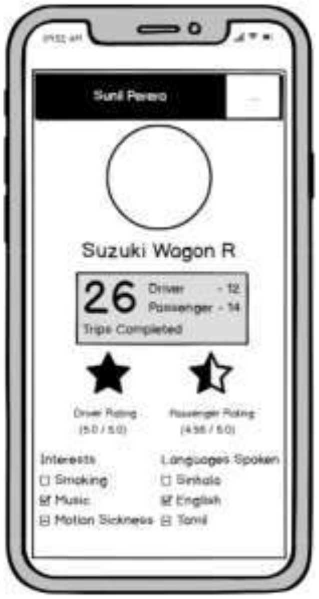
Maximum waited time 06.15 AM

Pickup Location Kaduwela Bus Stand

The request has been successfully sent to the Driver

We'll notify when the driver response

Image 2.13 - Passenger Request to the ride - Passenger



Sunil Perera

Suzuki Wagon R

26 Driver - 12
Passenger - 14
Trips Completed

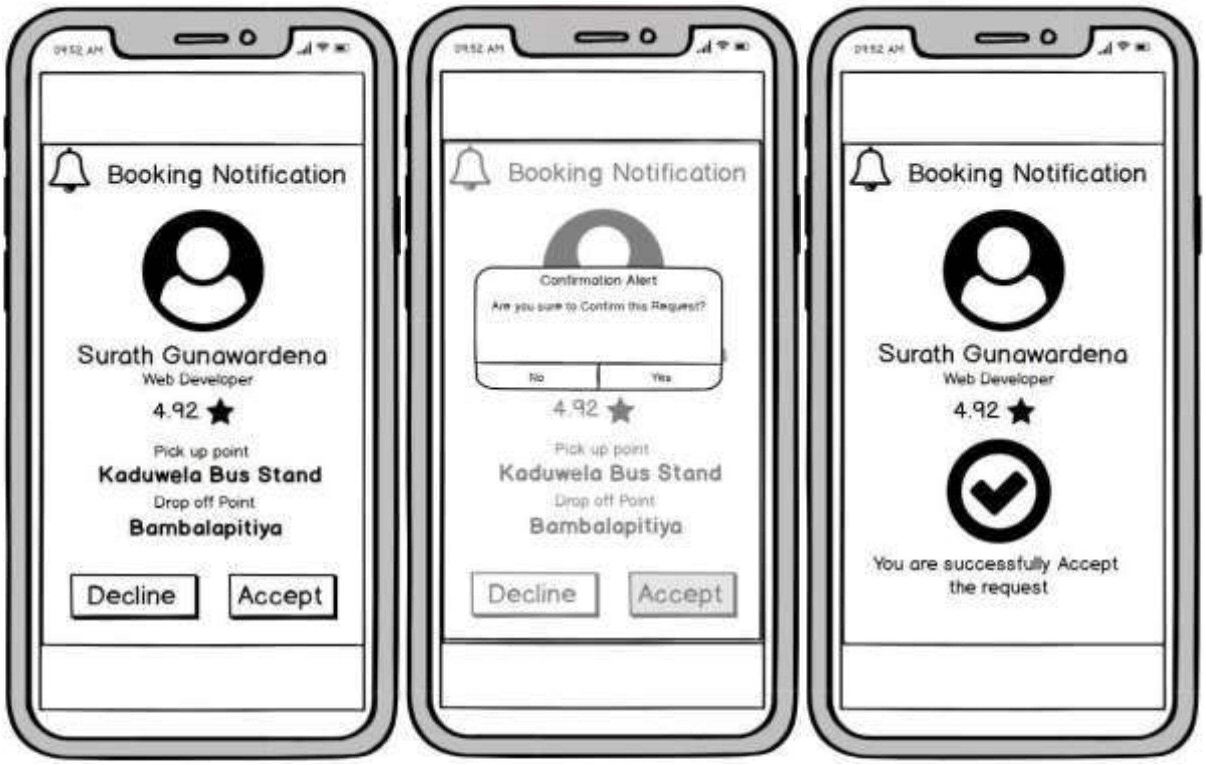
Driver Rating (5.0 / 5.0)

Passenger Rating (4.56 / 5.0)

Interests
□ Smoking
☑ Music
⊟ Motion Sickness

Languages Spoken
□ Sinhala
☑ English
⊟ Tamil

Image 2.14 - View user details – Driver and Passenger



Booking Notification

Surath Gunawardena
Web Developer
4.92 ★

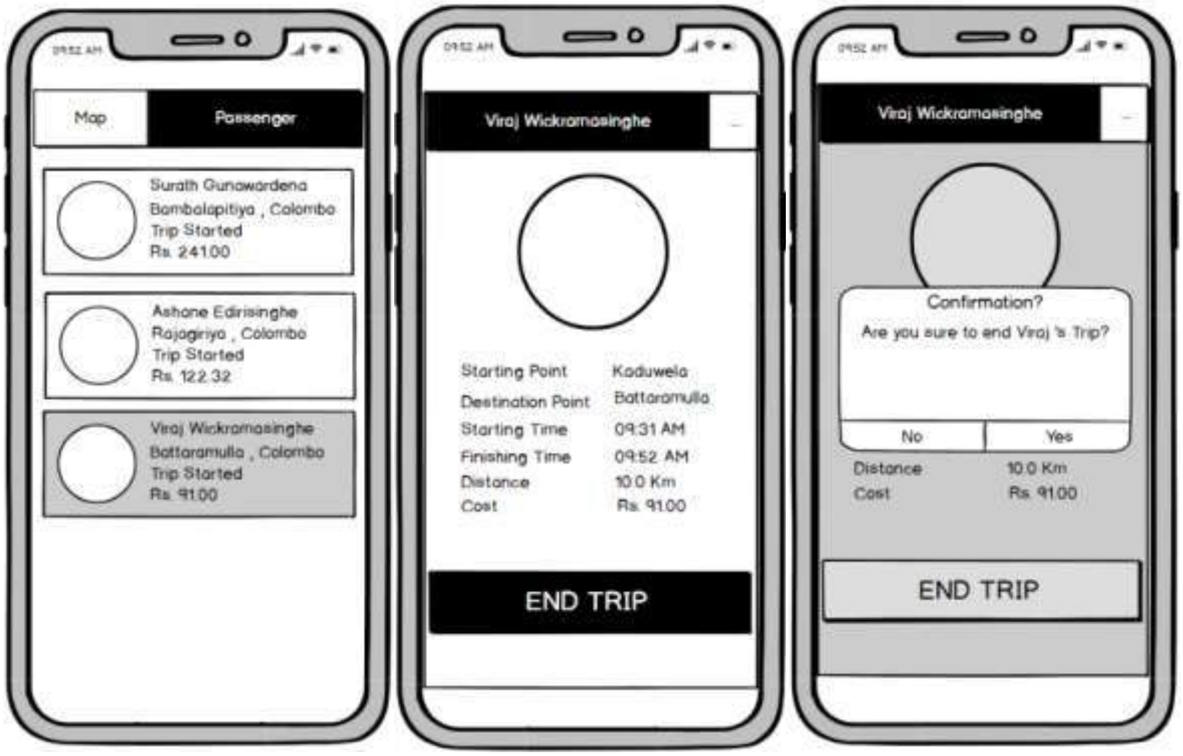
Pick up point
Kaduwela Bus Stand

Drop off Point
Bambalapitiya

Are you sure to Confirm this Request?

You are successfully Accept the request

Image 2.15 - Driver accepts the request which passenger sent to the ride - Driver



Surath Gunawardena
Bambalapitiya , Colombo
Trip Started
Rs. 241.00

Ashane Edirisinghe
Rajagiriya , Colombo
Trip Started
Rs. 122.32

Viraj Wickramasinghe
Battaramulla , Colombo
Trip Started
Rs. 91.00

Viraj Wickramasinghe

Starting Point	Kaduwela
Destination Point	Battaramulla
Starting Time	09:31 AM
Finishing Time	09:52 AM
Distance	10.0 Km
Cost	Rs. 91.00

Are you sure to end Viraj 's Trip?

Image 2.16 - End Trip- Driver

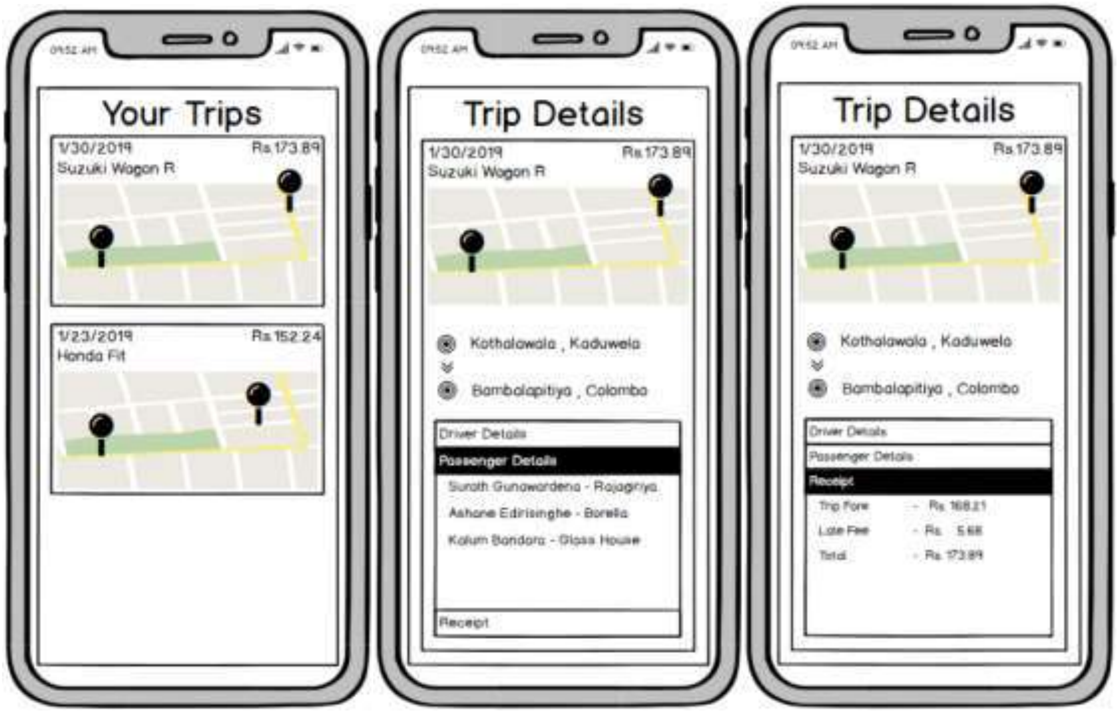


Image 2.17 - Trip History – Driver/Passenger

User Interfaces proposed for the Optimum Path Recognition are shown below,

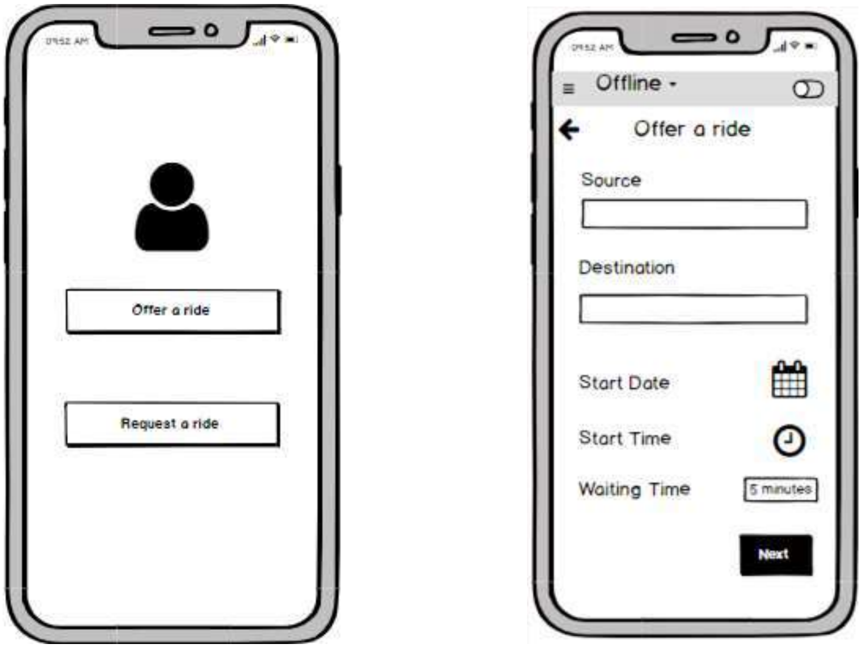
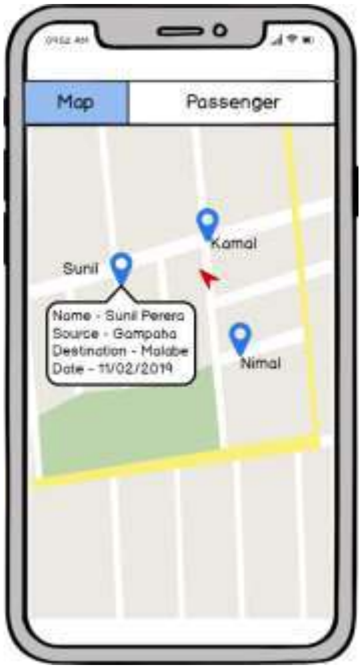


Image 2.18 - Driver offer a ride



Name - Sunil Perera
Source - Gampaha
Destination - Malabe
Date - 11/02/2019

Image 2.19 - View Booking Summary

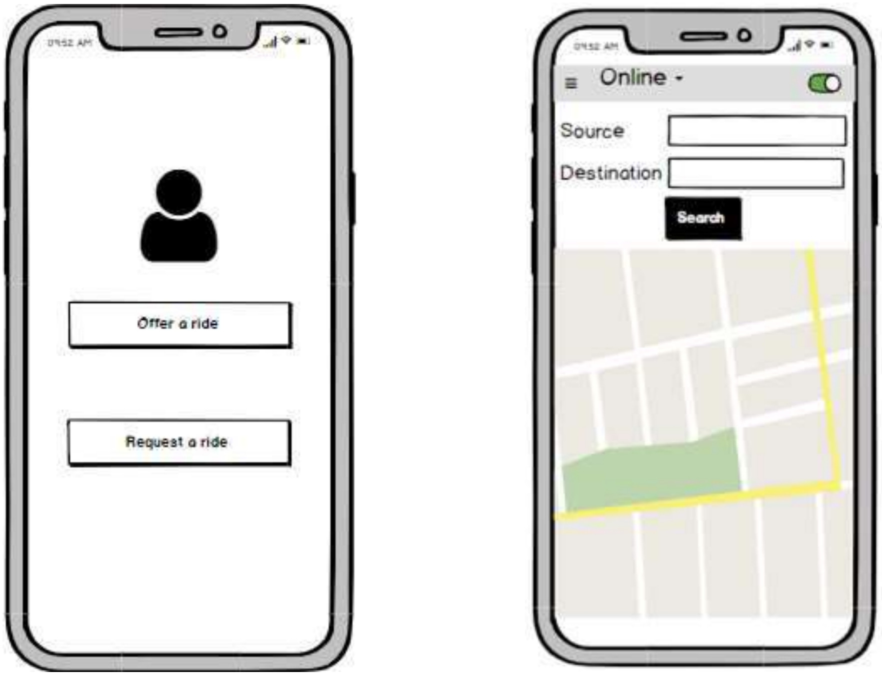


Image 2.20 - Passenger finds a ride

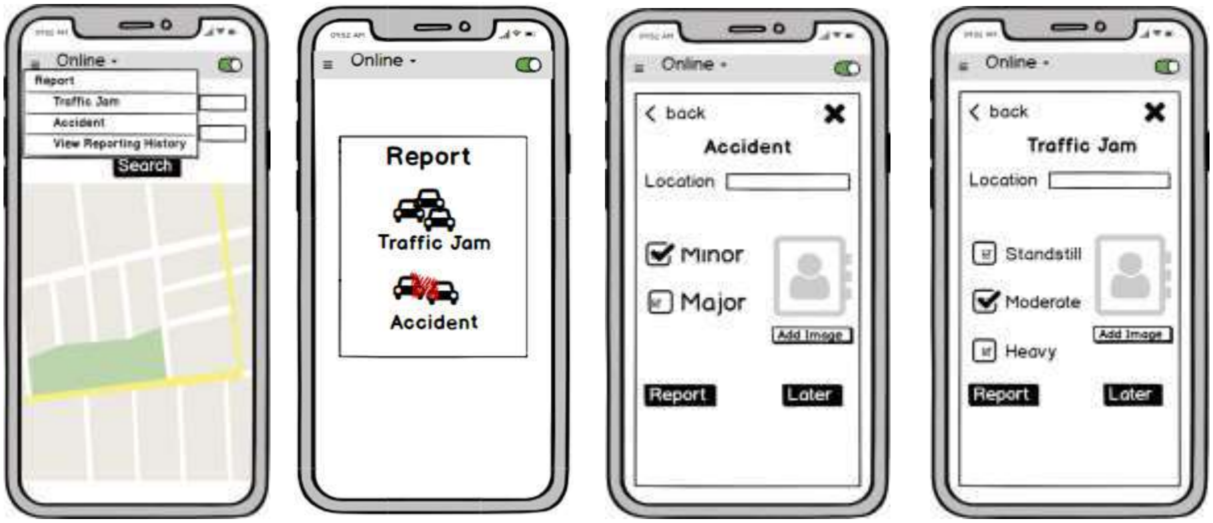


Image 2.21 - Report traffic jam and accidents

2.1.3 Hardware Interfaces

As the solution is a mobile application, this will be using a little amount of hardware. The user will be needing a smartphone with internet connectivity. User will need to allow the camera to capture images and GPS module should be accessed by the mobile phone. Apart from that ODB scanner has to be used with Bluetooth connectivity to retrieve the vehicle details to the app.

2.1.4 Software Interfaces

The main software interfaces used in our application are,

- For the application development: Android Studio
- Emulating purposes: Genymotion Emulator and Android Emulator
- Backend Database Server: MySQL
- Real-time data synchronization: Firebase
- For the Inbuilt data storage: SQLite
- For the implementation of algorithms: Python
- For the implementation of web API: Express.js

2.1.5 Communication Interfaces

Stable internet connection is an essential thing for the communication between the mobile app and web server, because synchronization between the databases, smooth data transaction, continuous loading the maps. As well as, Bluetooth facility will use for the between mobile and OBD Scanner.

2.1.6 Memory Constraints

The android mobile application is required,

- Android version should be 6.0 or higher
- 2 GB RAM(Minimum) and 4GB RAM is Recommended
- 100 MB Memory space

2.1.7 Operations

Most of the operations are done by the Passenger in the UPM and registration process is shared between both the passenger and the driver. Below are user operations done by Passenger, Driver and additionally Spouse/Guardian.

- User should sign up to our application
- User can update/delete existing account
- User should verify the phone number
- User should provide personal details
- User should provide personal interests and preferences
- User should add bank details
- User should add vehicle details
- Passenger should search for a driver
- Passenger should select a suitable driver from the suggested list
- Spouse or Guardian can reported suspected drivers

All the operations that are specific to the “Document Validation and Profile Rating maintenance” are listed below.

- User should upload an image of valid NIC
- User should upload an image of valid driving license for the validation

- Driver should rate / review the passengers at the end of the journey
- Passengers should rate / review the Driver, Vehicle and Co-Passengers at the end of journey.
- If a user blocks another user, it will be stored in the database.

All the operations that are specific to the “Fare Calculation” are listed below.

- Passenger should choose a preferred driver from the suggested list
- Passenger should set up the maximum waiting time to the response from the driver
- Passenger should provide the pickup location
- The driver should response the passenger request
- Driver ends the trip
- Passenger can decide to get off before destination
- The system should calculate fare of the trip

All the operations belong to the “Optimum Path Recognition” are listed as follow.

- Once the user login to the system as the driver, then user can set the source, destination, start date, start time and waiting time to offer a ride.
- If the user login to the system as a passenger, user can request for a ride by providing the particular source and destination. When the passenger click the search button, it will fetch all driver details from the database and compare each driver profiles which was gained through profiling technique and later on identify the closet path to reach for the destination.
- Drivers can view all booked summary details of each passenger and display their drop offs in a map.
- Drivers can add/report the traffic jam and accidents by entering the corresponding information with an image of the incident.
- If the authenticated user is a driver, he/she would be able to view all the reported incidents.

2.1.8 Site Adaptation Requirements

English is used as the common medium to represent the application interfaces. For the application to be used, the mobile phone should have internet connectivity, access to camera and access to GPS and Bluetooth connection. As the application is for ride-sharing among professionals, all the users should have a valid NIC and driving license.

Unless the mobile phone number is verified, no user is allowed to create an account at the first instance. This application only works in the OS above API 23. Older OS will not be compatible with the advanced features of our application.

2.2 Product Functions

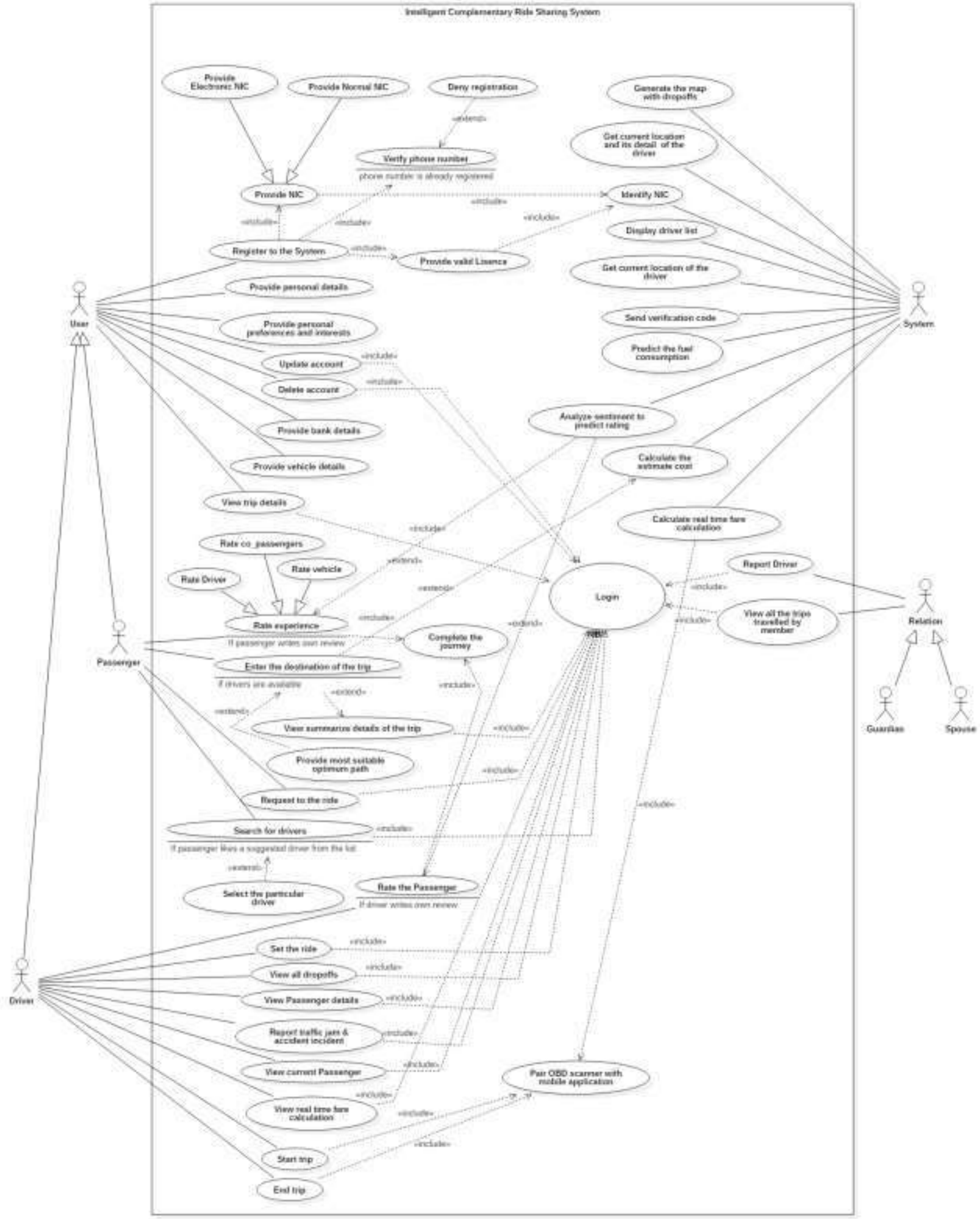


Image 2.2.1 – Use case diagram of +Go Application

Use Case ID	UC001
Use Case Name	Register to the system
Preconditions	User should have a valid mobile phone number
Actor	User
Main Success Scenario	1. Verify the phone number 2. Provide Personal Details 3. Provide Personal Interests/Preferences 4. Provide Bank details 5. Provide Vehicle Details
Extensions	1.a. If phone is already registered to the system, prompt an error message to the user. 4.a. If invalid bank details is provided, prompt an error message to the user. 5.a. If vehicle images are not uploaded, prompt an error message to the user.

Table 2.2.1 – Use Case 01

Use Case ID	UC002
Use Case Name	Login to the system
Preconditions	The user should be registered to the application.
Actor	User

Main Success Scenario	1. Provide the username 2. Provide the password 3. Click to login
Extensions	1.a. If the Username is invalid, system will prompt an error message. 2.a.1. If the Password is invalid, system will prompt an error message. 2.a.2. If the Password is forgotten, reset link is sent to the email.

Table 2.2.2 – Use Case 02

Use Case ID	UC003
Use Case Name	Search for drivers
Preconditions	The user should be registered to the application and have a valid account.
Actor	Passenger
Main Success Scenario	1. Include (Login to the system) 2. Search for a destination 3. Check the driver profile from the suggested list 4. Select the most suitable driver

Extensions	4. a. If the suggested driver not suits the passenger, passenger can select the driver that he/she wants.

Table 2.2.3 – Use Case 03

Use Case ID	UC004
Use Case Name	Report drivers
Preconditions	The spouse/guardian should have a valid account associated with the passenger.
Actor	Spouse/Guardian
Main Success Scenario	1. Include (Login to the system) 2. Check the trips travelled by associated passenger. 3. Report any drivers who travelled more frequently with associated passenger.

Table 2.2.4 – Use Case 04

Use Case ID	UC005
Use Case Name	Remove reported drivers from the search list
Actor	System

Main Success Scenario	1. Include(Report drivers) 2. Checks the database with the passenger's unique ID 3. Retrieve any reported driver IDs' 4. Remove those IDs' from the suggested list
Extensions	1.a. If database connectivity is not established, reconnect with the server.

Table 2.2.5 – Use Case 05

Use Case ID	UC006
Use Case Name	Get current location of the drivers
Actor	System
Main Success Scenario	1. Send request to database in every 5 seconds 2. Get the current location of the drivers 3. Update the current location of the MySQL database
Extensions	1.a. If polling doesn't work, prompt a error message to user to reconnect internet in the mobile phone.

Table 2.2.6 – Use Case 06

Use Case ID	UC007
Use Case Name	Update account
Actor	User
Main Success Scenario	1. Include (Login to the system) 2. Go to profile 3. Update the details 4. Click update button
Extensions	1.a. If login fails, re prompt to enter the credentials 3.a. If all the details are not entered, prompt an error message to the user.

Table 2.2.7 – Use Case 07

Use Case ID	UC008
Use Case Name	Delete account
Actor	User
Main Success Scenario	Include (Login to the system) Go to profile Click delete button

Table 2.2.8 – Use Case 08

Use Case ID	UC009
Use Case Name	Register to the app
Actors	User
Preconditions	There is internet connection to the mobile
Post conditions	User should be directed to a page to insert another user information
Main Success Scenario	1. The use case starts when the user selects sign up in the app 2. Direct an activity to insert image of NIC card 3. Show the extracted NIC number from NIC card in a text box 4. User clicks “next” 5. Direct an activity to insert image of License card 6. Use case ends when the license gets validated and new activity starts
Extensions	2a. If the image does not recognize a human face in it, will ask the user to insert another image 2b. If NIC number is not recognized, will ask user to add clear image 3a. If the user edits the identified NIC number in textbox, user input will be saved in temporary database. 5a. If the image does not contain human face, will ask user to upload another image.

	5b. If NIC number is not identified by the image of License, will ask the user to upload new image 5c. If NIC number extracted does not match more than 75% to the NIC number in temporary storage, will ask the user to upload new image

Table 2.2.9 – Use Case 09

Use Case ID	UC010
Use Case Name	Rate the passenger
Actors	Driver
Preconditions	1. Driver has already completed trip 2. There is internet connection
Main Success Scenario	1. Use case starts when a passenger gets down after completing his /her trip 2. Direct an activity for driver to rate the experience 3. Driver adds the ratings 4. The use case is closed when the activity is over
Extensions	3a. If driver skips rating, activity is closed 3b. If driver gives rating below 5, he will be given a set of keywords to clarify the dissatisfaction, and if keywords are not sufficient, will be asked to write their own review on experience.

Table 2.2.10 – Use Case 10

Use Case ID	UC011
Use Case Name	Rate overall experience
Actors	Passenger
Preconditions	Passenger has completed the trip and got down There is internet connection
Main Success Scenario	1. Use case starts when passenger finish the trip 2. Passenger rates the overall experience 3. Passenger rates experience less than 5 4. System asks which made the rating low as “Driver”,” Vehicle”,” Co-passengers” 5. System provides set of keywords to be selected regarding experience based on above selection. 6. Use case ends when the rating activity finishes
Extensions	2a. If passenger skips rating experience, all “driver”, “vehicle” and ”co-passengers” get default rating. 3a. If passenger rates experience as 5, he can write a complement to the driver 4a. If the selection is “Co-passengers”, list of passengers of the session will be displayed to be selected 5a. If passenger wants to write a review, a text box will be appeared instead of the keywords

Table 2.2.11 – Use Case 11

Use Case ID	UC012
Use Case Name	Calculate the estimated cost
Actors	System
Preconditions	The Passenger should log in to the system Suggested users (drivers) should add to the vehicle in the system Passenger should enter the destination of the trip
Main Success Scenario	1. Retrieve the suggested driver list from the user profile management component 2. Get the vehicle information 3. Calculate the distance between source and destination 4. Get traffic condition
Extension	1.a. If drivers are available

Table 2.2.12 – Use Case 12

Use Case ID	UC013
Use Case Name	Predict the fuel consumption
Actors	System
Preconditions	The driver should log in to the system

Main Success Scenario	1. Vehicle details get from the driver 2. Vehicle details pass to the algorithm and get the output (Predicted fuel consumption) 3. Resulted output send to the database according to the vehicle id

Table 2.2.13 – Use Case 13

Use Case ID	UC014
Use Case Name	Request for the trip
Actors	Passenger
Preconditions	The passenger should log in to the system
Main Success Scenario	1. Choose the driver from the suggested list 2. Redirect to the trip details page 3. Open a popup window when clicking the button named as “Join a Ride.” 4. Set up the maximum waited time to the response from the driver and pickup point 5. Click the confirm button to verify the details 6. Receive the Notification as “Wait for the driver response.”

Extension	2.a. If the passenger not satisfied the trip details, then back and choose another driver 6.a. If driver not response or reject the passenger request send a notification to the passenger

Table 2.2.14 – Use Case 14

Use Case ID	UC015
Use Case Name	Start Trip
Actors	Driver
Preconditions	The driver should log in to the system The passenger has already sent the request for the ride The Bluetooth connection should work with a mobile application and OBD Scanner
Main Success Scenario	1. Accept the request which is passenger sent to the driver and that passenger add to the current passenger list with "Not Started Yet" Status 2. Reaching pick up point of the passenger using provided navigation from the system 3. Click on the correct passenger from current passenger list and press the "Start Trip" button to start the trip 4. Passenger's status will change as "Trip Started."

	5. The passenger receives the notification about the trip started.
Extension	1.a. If the driver does not accept the passenger request send a notification to the passenger and passenger have to choose another driver

Table 2.2.15 – Use Case 15

Use Case ID	UC016
Use Case Name	Calculate the real-time fare calculation
Actors	System
Preconditions	The driver should log in to the system The passenger has already sent the request for the ride The Bluetooth connection should work with a mobile application and OBD Scanner
Main Success Scenario	1. Get current travel distance get from every 30 from OBD Scanner 2. Get estimate fuel consumption of the vehicle 3. The current market price of the liter of fuel (Petrol / Diesel) 4. Number of passengers currently joined the trip 5. Calculate the waiting time

Table 2.2.16 – Use Case 16

Use Case ID	UC017
Use Case Name	Offer a ride
Actors	Driver
Preconditions	Driver should be a registered user
Post conditions	The user will be able to set his or her trip with required details
Main Success Scenario	1. Login to the system as the driver by entering user credential. 2.Click offer a ride button 3. Set the trip by entering source, destination, start date, start time and waiting time. 4. Finally click next button and it will show the confirmation message of the trip.

Table 2.2.17 – Use Case 17

Use Case ID	UC018
Use Case Name	View booking summary details
Actors	Driver
Preconditions	User(driver) should be an registered user

Post conditions	Showing all the destinations(drop offs) arranged according to the passengers preferences with their details
Main Success Scenario	1.Login to the system as a driver 2. Select the interface where map and passenger details are displayed. 3. Click the map tab. 4. Then the system will load the map with all drop off points.

Table 2.2.18 – Use Case 18

Use Case ID	UC019
Use Case Name	Set the destination point
Actors	Passenger
Preconditions	User(passenger) should be an registered user
Post conditions	User will be able to request the ride while matching the passenger’s profile with suitable drivers and identify the best optimum path to reach the destination

Main Success Scenario	1. Login to the system as the passenger by entering user credential. 2. Click request a ride button 3. Request the ride by entering source and destination. 4. Click search button

Table 2.2.19 – Use Case 19

Use Case ID	UC020
Use Case Name	Add or report traffic jam and accidents by uploading the images of the relevant incident
Actors	Driver
Preconditions	Driver should be a registered user
Post conditions	Entered data will be saved into the database and user will be able to view those reports
Main Success Scenario	1. Login to the system as a driver 2. Click the report tab 3. Then select either traffic jam option or accident reporting option 4. Next fill the relevant details 5. Then click add image button and add and image relevant to the incident 6. Finally click the report button

Table 2.2.20 – Use Case 20

Use Case ID	UC021
Use Case Name	View traffic jam and accidents
Actors	Driver
Preconditions	The user should be already login to the system
Post conditions	User will be able to see all the reports of the incidents
Main Success Scenario	1. Login to the system by entering user credentials 2. Then select report tab 3. Next select view reporting history option 4. Finally system will show all the reports in detail related to particular incidents.

Table 2.2.21 – Use Case 21

2.3 User Characteristics

"+Go" application is developing for the traffic minimization in urban areas. This product is mainly focused on office staff. They can either drive the vehicle or go on the vehicle as a passenger. User should have understandability of how to use smartphone/application and knowledge of the English language. User age limit should be higher than 18 years as well as have both NIC and driving license where those information will be used for the user registration.

2.4 Constraints

"+Go" ride sharing mobile application targeted to the Android mobile phone users as this is the preliminary phase of the project. Following requirements are considered to run +Go mobile application.

Mobile Application

- User’s smartphone must be Android OS of API level 23 with Lollipop version.
- Android version should be 6.0 or higher
- Smartphone with stable internet connectivity is required and GPS location should be enabled.
- Mobile phone CPU should be 1.6GHz or higher for a best performance.
- Mobile phone RAM should be 2GB or higher for an optimum performance.
- Mobile phone should have a camera with higher resolution of 8MP or above.
- Smartphone should have a proper Bluetooth accessibility without any issues.

Implementation Constraints

- Android is used for the development of the mobile application
- Android Studio IDE is used for the application development as the development environment.
- Firebase would be used for real time data synchronization as a real-time database.
- SQLite would be used for data storing purposes as an in built data storage.

2.5 Assumptions and Dependencies

- Smartphone with stable internet connectivity
- Registering user should be having a vehicle to register to the system
- GPS module is enabled in the mobile phone
- Obtaining the current location of the drivers is heavily depend on the firebase synchronization
- Frequently used data is stored in-built database to avoid overhead in connecting to the servers
- User will not add vague information to create fake accounts
- The smart phone is switch on throughout the journey as well as have the enough power of battery.
- The user has a valid NIC and license.
- The users will not insert vague ratings or ratings on others

- This application provide its service only in Colombo district area as for the initial stage.
- The Bluetooth connectivity is enabled when the user act as a driver.
- OBD port of the vehicle is supported for the OBD Scanner (OBD II ELM 327).

2.6 Apportioning of Requirements

In the section 1 and section 2 of the SRS, we mainly focused on the primary functions associated with the UPM, DVPRM, OPR and FC. In the first section of the document, we briefly introduce with the system functions. In the next section, detailed explanation of the requirements is provided. In the section 3, specific requirements are presented with OOP methodology. Due to the dependency with OOP, the implementation may differ from normal implementation.

From the first 2 sections, the methodology, interfaces, requirements etc. are discussed. In the 3rd section, they are elaborated in detail. All these requirements are expected to be implemented in the initial phase of the implementation. Additional requirements are solely depend on the time constraint and will not be committed by developers unless there is a special requirement.

3. Specific Requirements

3.1 External interface requirements

3.1.1 User Interfaces

Name of Item	Description of Purpose	Source of input or destination of output	Valid range accuracy and tolerance	Timing	Relationship to other input and output	Data formats
Phone number Field	Phone number is required	Phone keyboard	Any 10 digit valid number	N/A	No	Number
Verify Button	Click to send the verification code	Phone keyboard	N/A	N/A	Redirect to Verify Code	N/A

Table 3.1.1 - Send Verification Code Interface

Name of Item	Description of Purpose	Source of input or destination of output	Valid range accuracy and tolerance	Timing	Relationship to other input and output	Data formats
Email Field	Email is required	Phone keyboard	Any alphanumeric characters with special symbols	N/A	No	String
Password Field	Password is required	Phone keyboard	Any alphanumeric characters with special symbols	N/A	No	String
Login Button	Click to Login to the system	Phone keyboard	N/A	N/A	Redirect to user profile	N/A

Table 3.1.2 - Login Interface

Name of Item	Description of Purpose	Source of input or destination of output	Valid range accuracy and tolerance	Timing	Relationship to other input and output	Data formats
Code Field	Code is required	Phone keyboard	4 digit code	N/A	No	Number
Verify Button	Click to Verify the phone number	Phone keyboard	N/A	N/A	Redirect to user profile	N/A

Table 3.1.3 - Verify the code Interface

Name of Item	Description of Purpose	Source of input or destination of output	Valid range accuracy and tolerance	Timing	Relationship to other input and output	Data formats
Full Name Field	Full name is required	Phone keyboard	Any number of characters	N/A	No	String
Profession Field	Profession is required	Phone keyboard	Select from the given list	N/A	No	String
Email Field	Email is required	Phone keyboard	Any alphanumeric characters with special symbols	N/A	No	String
DOB Field	DOB is required	Phone keyboard	Any date selected from the calendar	N/A	No	Date
Gender Field	Gender is required	Phone keyboard	Select either one of radio buttons	N/A	No	Boolean
Guardian/Spouse Name Field	Name is required	Phone keyboard	Any number of characters	N/A	No	String
Guardian/Spouse phone number Field	Phone number is required	Phone keyboard	Any 10 digit valid number	N/A	No	Number
Save Button	Click to	Phone	N/A	N/A	Redirect to user	N/A

	Login to the system	keyboard			preferences	

Table 3.1.4 - Add details to profile Interface

Name of Item	Description of Purpose	Source of input or destination of output	Valid range accuracy and tolerance	Timing	Relationship to other input and output	Data formats
Gender Preference Field	Gender Preference flag is required	Phone keyboard	Select from 3 radio buttons	N/A	No	String
Language Spoken Field	Language Spoken flag is required	Phone keyboard	Select from 3 radio buttons	N/A	No	String
Smoking Field	Smoking flag is required	Phone keyboard	Select from 2 radio buttons	N/A	No	Boolean
Music Lover Field	Music Lover flag is required	Phone keyboard	Select from 2 radio buttons	N/A	No	Boolean
Motion Sickness Field	Motion Sickness flag is required	Phone keyboard	Select from 2 radio buttons	N/A	No	Boolean
Confirm Button	Click to Login to the system	Phone keyboard	N/A	N/A	Redirect to Add payment details	N/A

Table 3.1.5 - Add details to profile Interface

Name of Item	Description of Purpose	Source of input or destination of output	Valid range accuracy and tolerance	Timing	Relationship to other input and output	Data formats
Credit	Credit Card	Phone	16 digit valid	N/A	No	Number

Card Number Field	Number is required	keyboard	number			
Expiration Date Field	Expiration Date is required	Phone keyboard	4 digit date with month and year	N/A	No	Date
CSV Field	CSV is required	Phone keyboard	3 digit number	N/A	No	Number
Login Button	Click to Login to the system	Phone keyboard	N/A	N/A	Redirect to add vehicle	N/A

Table 3.1.6 - Add payment details Interface

Name of Item	Description of Purpose	Source of input or destination of output	Valid range accuracy and tolerance	Timing	Relationship to other input and output	Data formats
Select Brand Field	Vehicle Brand is required	Phone keyboard	Select any from the list	N/A	No	String
Select Model Field	Model is required	Phone keyboard	Select any from the list	N/A	Yes	String
Vehicle Number Field	Vehicle Number is required	Phone keyboard	Any alphanumeric characters	N/A	No	Number
Manufactured Year Field	Manufactured Year is required	Phone keyboard	Select any from the list	N/A	No	Number
Fuel Type Field	Fuel Type is required	Phone keyboard	Select any from the list	N/A	No	String
Transmission Type Field	Transmission Type is required	Phone keyboard	Select any from the list	N/A	No	String

Registered Year Field	Registered Year is required	Phone keyboard	Select any from the list	N/A	No	int
Engine Capacity Field	Engine Capacity is required	Phone keyboard	Select any from the list	N/A	No	String
Upload images	Click to upload maximum of 6 images	Phone keyboard	Select any gallery	N/A	No	Blob
Login Button	Click to Login to the system	Phone keyboard	N/A	N/A	Redirect to add vehicle	N/A

Table 3.1.7 - Add vehicle details Interface

Name of Item	**Description of Purpose**	**Source of input or destination of output**	**Valid range accuracy and tolerance**	**Timing**	**Relationship to other input and output**
Upload NIC button	Opens an activity to capture the image of NIC	Camera	N/A	N/A	Redirect to the previous page
Verify button	Uploads the image to the server and retrieve the NIC number in the image	Touch screen	N/A	N/A	Show a dialog box with a proper message

Table 3.1.8 - Upload NIC

Name of Item	**Description of Purpose**	**Source of input or destination of output**	**Valid range accuracy and tolerance**	**Timing**	**Relationship to other input and output**
Upload License button	Opens an activity to capture image of License	Camera	N/A	N/A	Redirect to the previous page
Verify button	Uploads the image to the server and retrieve the NIC number and license expiration date in the license image	Touch screen	N/A	N/A	Show a dialog box with proper message

Table 3.1.9 - Upload License

Name of Item	**Description of Purpose**	**Source of input or destination of output**	**Valid range accuracy and tolerance**	**Timing**	**Relationship to other input and output**
Star rating	The stars selected by the user will be used in calculating the rating	Touch screen	N/A	N/A	Redirect to the next page
Submit button	Uploads rating to the web server	Touch screen	N/A	N/A	Show a dialog box with a proper message

Table 3.1.10 - Five-star rating interface by the driver /passenger

Name of Item	**Description of Purpose**	**Source of input or destination of output**	**Valid range accuracy and tolerance**	**Timing**	**Relationship to other input and output**

Star rating	The stars selected by the user will be used in calculating the rating	Touch screen	N/A	N/A	Redirect to the next page
Keywords	Keywords selected by user will be updated in the database	Touch screen	N/A	N/A	Show a dialog box with a proper message
Review textbox	Sentiment given by the user will be updated and processed in the server	Keypad	N/A	N/A	Show a dialog box with a proper message

Table 3.1.11 - Low rating interface by the driver

Name of Item	**Description of Purpose**	**Source of input or destination of output**	**Valid range accuracy and tolerance**	**Timing**	**Relationship to other input and output**
Star rating	The stars selected by the user will be used in calculating the rating	Touch screen	N/A	N/A	Redirect to the next page
Vehicle / Driver/ Fellow Passenger button	The user can select those buttons to specify which made them dissatisfied	Touch screen	N/A	N/A	Redirect to the next page
Keywords	Keywords selected by user will be updated in the database	Touch screen	N/A	N/A	Redirect to the next page

Review textbox	Sentiment given by the user will be updated and processed in the server	Keypad	N/A	N/A	Redirect to the next page
Submit button	Submit the review	Touch screen	N/A	N/A	Show a dialog box with a proper message

Table 3.1.12 - Low rating interface by the passenger

Name of Item	**Description of Purpose**	**Source of input or destination of output**	**Valid range accuracy and tolerance**	**Timing**	**Relationship to other input and output**
Enter the destination Text field	To fetch the suitable drivers for the passenger.	Touch screen	N/A	N/A	Redirect to the most suitable driver list
List of suggested driver	To choose a desired driver among the suggested driver list.	Touch screen	N/A	N/A	Redirect to the trip details page
Join a ride Button	To set a maximum time passenger wait for the driver's response for the request and set the pickup points	Touch screen, GPS module, key pad	N/A	N/A	Open a Confirmation pop up window
Confirmation Button	To verify the trip with Terms and conditions	Touch Screen	N/A	N/A	Notification received to the passenger as a “Your request

					successfully sent to the driver. Please wait for his response!”

Table 3.1.13 – Passenger request to the driver

Name of Item	**Description of Purpose**	**Source of input or destination of output**	**Valid range accuracy and tolerance**	**Timing**	**Relationship to other input and output**
Passenger whose status as “Not yet started.”	To Identify the passenger who has not started the trip.	Touch Screen	N/A	N/A	Pop up window and display the details of a selected passenger
Start Trip button	To Start the trip when passenger get into the vehicle	Touch Screen, OBD Scanner	N/A	N/A	Passenger Status will be changed as Trip Started and Sent notification to the passenger

Table 3.1.14 – Start a trip

Name of Item	**Description of Purpose**	**Source of input or destination of output**	**Valid range accuracy and tolerance**	**Timing**	**Relationship to other input and output**
Passenger whose status as “Trip Started.”	To Identify the passenger who has already started the trip.	Touch Screen	N/A	N/A	Redirect to the passenger profile which creates in dynamically

End Trip button	To End the trip when passenger get off the vehicle	Touch Screen, OBD Scanner	N/A	N/A	Pop up a confirmation box.
Verify (Yes) Button	To verify the ending.	Touch Screen	N/A	N/A	Passenger Status will be changed as Trip Ended and Sent notification to the passenger with travelled cost

Table 3.1.15 – End a trip

Name of Item	**Description of Purpose**	**Source of input or destination of output**	**Valid range accuracy and tolerance**	**Timing**	**Relationship to other input and output**
The menu item of view history	To navigate the past trip details	Touch Screen	N/A	N/A	Redirect to the trip history page
Tab Button Called as “Driver”	To filler out the trip user act as a driver	Touch Screen	N/A	N/A	Results of trip user act as a driver
Specific Trip grid	To view in details description about the trip	Touch Screen	N/A	N/A	In here driver can identify who are the passengers of the trip, their destination and how much earn from this ride

Table 3.1.16 – View trip history as a driver

Name of Item	**Description of Purpose**	**Source of input or destination of output**	**Valid range accuracy and tolerance**	**Timing**	**Relationship to other input and output**
The menu item of view history	To navigate the past trip details	Touch Screen	N/A	N/A	Redirect to the trip history page
Tab Button Called as “Passenger”	To filler out the trip user act as a passenger	Touch Screen	N/A	N/A	Results of trip user act as a passenger
Specific Trip grid	To view in details description about the trip	Touch Screen	N/A	N/A	In here driver can identify who are the co-passengers of the trip, their destination, driver details and fare breakdown of the ride

Table 3.1.17 – View trip history as a passenger

Name of Item	**Description of Purpose**	**Source of input or destination of output**	**Valid range accuracy and tolerance**	**Timing**	**Relationship to other input and output**
Source Field	To enter the source	Keypad	Alphanumeric characters	N/A	No
Destination field	To enter the destination	Keypad	Alphanumeric characters	N/A	No

Start Date	To select the start date	Touch Screen	Valid date range	N/A	No
Start Time	To enter the start time	Keypad	Valid time range	N/A	No
Waiting Time	To enter the waiting time	Keypad	Valid time range	N/A	No
Next Button	Search for the ride requests	Touch Screen	N/A	Response should give within 1-4 seconds	Show a dialog box with a proper message

Table 3.1.18 – Offer a ride Interface

Name of Item	**Description of Purpose**	**Source of input or destination of output**	**Valid range accuracy and tolerance**	**Timing**	**Relationship to other input and output**
Tab Button named as “Map”	To view a map with drop off points of all passengers	Touch Screen	N/A	N/A	Show the map page with drop offs
Drop Off Points	To view a detailed description about the passenger’s ride	Touch Screen	N/A	N/A	Show the drop offs in detail

Table 3.1.19 – View Booking Summary Interface

Name of Item	**Description of Purpose**	**Source of input or destination of output**	**Valid range accuracy and tolerance**	**Timing**	**Relationship to other input and output**
Source Field	To enter the source	Keypad	Alphanumeric characters	N/A	No
Destination Field	To enter the destination	Keypad	Alphanumeric characters	N/A	No
Search Button	Search for a suitable ride while matching drivers profiles	Touch Screen	N/A	Response should give within 1-4 seconds	Should show the driver list with their information

Table 3.1.20 – Passenger finds a ride Interface

Name of Item	**Description of Purpose**	**Source of input or destination of output**	**Valid range accuracy and tolerance**	**Timing**	**Relationship to other input and output**
Location field	To enter the location	Keypad	Alphanumeric characters	N/A	No
Image View	To add an image of the relevant incident	Touch Screen	Image view	N/A	No

Incident type	To select the incident type	Touch Screen	Checkbox	N/A	No
Report Button	Add the incident information	Touch Screen	N/A	Response should give within 1-4 seconds	Show a dialog box with proper message as all the reporting incidents like traffic jam and accidents are saved respectively

Table 3.1.21 – Report traffic jam and accident Interface

3.1.2 Hardware Interfaces

1. Smart Phone
   a. CPU: 1.6GHz or higher
   b. Storage: 200MB or higher
   c. RAM: 2GB or higher
   d. OS: Android API level 23 - Lollipop
   e. Front Camera: 8 MP
   f. Back Camera: 13 MP
   g. GPS Module

2. Server
   a. CPU: 2.6GHz Intel Core i7 (6th Gen or Higher)
   b. RAM: 16GB DDR3 or higher
   c. Storage: 1TB HDD
   d. OS: Linux (Preferably Ubuntu 16.04)

3.1.3 Software Interfaces

a. Android Studio 5.6
b. Visual Studio Code
c. Android Emulator
d. Firebase Interface
e. SQLite Browser
f. Java SE 8
g. Leaflet V 1.3.4 upwards
h. Google API

Web API's are developed to connect with the web server, python scripts and functions like image uploading. OpenCV, tesseract used in image processing and text identification.

3.1.4. Communication Interfaces

The process of communication happens through internet connectivity. To get the current location of the driver firebase communication is established via internet connectivity. It is preferred to have at least a 3G connection for the best data synchronization. As well as, Bluetooth facility will use for the between mobile and OBD scanner to read some values from the engine of the vehicle.

3.2. Classes/Objects

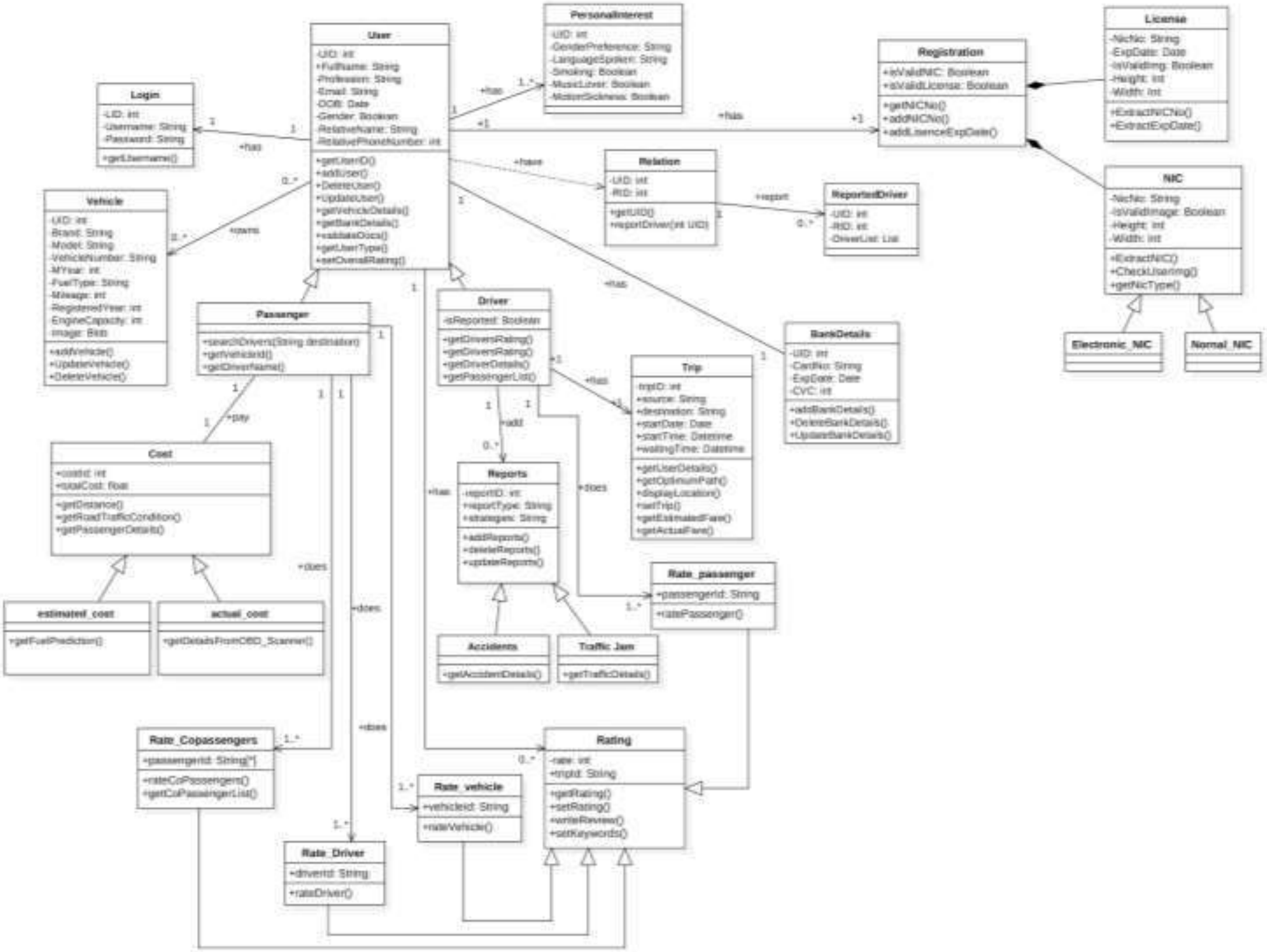


Image 3.2 Class diagram of +Go Application

(*for clearer image: Click Here)

3.3 Performance requirements

The system will be based on the android platform and it should have a proper internet connectivity to maintain the quick responses and the stability of this application. Depending upon the stability of the internet connection, initial loading time of the mobile application will be varied as it requires to gather real time data of the users. All the processing tasks can be enhanced by using a mobile phone with minimum 2GM RAM and 1.6 GHz CPU for the execution. Also, it should presence a good internet connection and Bluetooth connection between the mobile application and the OBD Scanner. Therefore it will help to increase the overall performance of the application. In here, backend should have an ability to handle multiple requests within a given period of time. Since the databases are stored in servers, the system should allow multiple users to do their task concurrently without any interruptions.

3.4 Design constraints

There are several design constraints that need to be considered in designing the “+Go” mobile application.

- Sessions need to be maintained for each and every individual user
- Loading time of the mobile application should be less than few seconds
- Application and database should be available 24x7
- User Interfaces should be easy to use and understand
- Performance, reliability and security must be achieved in the application.

3.5 Software system attributes

3.5.1 Reliability

Reliability is a vital thing in any system. When executing the system functionality, it should work smoothly with minimum failures or without any failure. The mobile application fetches details from the database with a less amount of time with high accuracy and availability. Before data is send to the database, we will check for validation of input fields. The application will notify the users, when entering any invalid inputs. In addition to that, system should be tested using several testing techniques to reduce the possible set of the failures.

3.5.2 Availability

The application should be available 24x7 and server should be up and running always. Users should be able to access the app from anywhere they go using their internet connection. Before the post-release, each component of the **+Go** application is thoroughly tested individually as well system wise to make sure those executes without any bugs. System downtime is also taken into consideration while implementing. In scenarios like users going offline during the trip, which won’t affect the overall functionalities, as the system will be using the cloud for data storage.

3.5.3 Security

Security is one of the major concern when it comes to the systems which deals with sensitive data of users. System should have an ability to outlast the unauthorized users. During the registration process, password is taken from the user and saved it in the database after hashing using the salt method. The users are expected to log in to the system using correct login credentials. The system shall not allow access if the user fails to provide correct login information. The users are identified with the NIC number to eliminate fake registrations. Images of the documents submitted at the registration will validate and verify the users. All the databases are created with admin privileges and no any other external party is allowed to access them. Security of the proposed system will be maximized by maintaining and controlling the server side and also the real-time database security will be provided by the Firebase database.

3.5.4 Maintainability

Arrival of new requirements in order to improve the functionality and performance of the application without any failures can be called as the maintainability of the system. This proposed system uses modularization technique for the algorithms to reduce the coupling between each module while enhancing the cohesion among them. As most of the algorithms are used in backend, backend server will help to increase the maintainability of the system. The application is coded according to the classes and objects while serving the updates of the components in future. In case of a failure or bugs during runtime will be fixed with instant consequences.

3.6. Other Requirements

- The mobile should be with enough memory, RAM, and battery power.
- Application and database should be available 24x7
- Use open source technologies.
- Usage of a reliable web server

4. Supporting Information

4.1 Appendices

Throughout the SRS, we have elaborated the solution that we presented to overcome the traffic congestion in morning hours. For that, all these four components play major roles in the implementation.

4.2. References

[1] Urban Urban Transport Master Plan for Colombo Metropolitan Region and Suburbs. (2014). [ebook] Colombo ,Sri Lanka: Ministry of Transport, pp.25,26. Available at: http://www.transport.gov.lk/web/images/stories/CoMTrans%20Final%20Report_S.pdf [Accessed 13 Jan. 2019].

[2][online] Available at: http://www.ft.lk/motor/Innovative-car-pooling-app-UDIO-goes-live/55-645554
[Accessed 18 Jan. 2019].

[3] [online] Available at: https://www.newsfirst.lk/2017/03/16/rs-500m-loss-incurred-daily-traffic-congestion-transport-authorities-observe/ [Accessed 30 Dec. 2018].

[4]"Carpooling", Carpooling.lk, 2019. [Online]. Available: http://carpooling.lk/
[Accessed: 03- Feb- 2019].

[5]"Carpool for better commute and quality of life", RideShare.lk, 2019. [Online]. Available: http://rideshare.lk/. [Accessed: 22- Feb- 2019].

[6] C. Patel, A. Patel and D. Patel, “Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study,” International Journal of Computer Applications (0975 – 8887), vol. 55, no. 10, October 2012.

[7] R. Smith and Google Inc, “An overview of the Tesseract OCR Engine,” Proc. 9th IEEE Intl. Conf. on Document Analysis and Recognition (ICDAR), 2007, pp. 629-633.

[8] V. U. Ramya and K. T. Rao, “Sentiment Analysis of Movie Review using Machine Learning Techniques,” International Journal of Engineering & Technology, vol. 7, pp. 676-681, 2018.

[9] R. Valiente, M. T. Sadaike, J. C. Gutiérrez, D. F. Soriano, G. Bresson and W. V. Ruggiero, “A Process for Text Recognition of Generic Identification Documents Over

Cloud Computing,” 2016.

[10] Llombart, Òscar Romero, “Using Machine Learning Techniques for Sentiment Analysis,” 2017.

[11]"National identity card (Sri Lanka)", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/National_identity_card_(Sri_Lanka). [Accessed: 20- Feb- 2019].

[12]"About RideShare.lk", *RideShare.lk*, 2019. [Online]. Available: http://www.rideshare.lk/en/about/. [Accessed: 22- Feb - 2019].

[13] "Carpooling", *Carpooling.lk*, 2019. [Online]. Available: http://carpooling.lk/. [Accessed: 20- Feb - 2019].

[14] "Rs. 500m loss incurred daily from traffic congestion - transport authorities observe - Sri Lanka Latest News", *Sri Lanka News - Newsfirst*, 2019. [Online]. Available: https://www.newsfirst.lk/2017/03/16/rs-500m-loss-incurred-daily-traffic-congestion-transport-authorities-observe/. [Accessed: 24- Feb - 2019].

[15]"Innovative car pooling app UDIO goes live", *Ft.lk*, 2019. [Online]. Available: http://www.ft.lk/motor/Innovative-car-pooling-app-UDIO-goes-live/55-645554. [Accessed: 20- Feb - 2019].

[16] Ojaswa S. , Darka M. , Francois A. and Girija D., 2005.TRAVELING SALESPERSON APPROXIMATION ALGORITHM FOR REAL ROAD NETWORKS Technical Report 388, Department of Geomatics Engineering, University of Calgary 2500 University Drive NW, Calgary, AB, Canada, T2N 1N4

[17] Vaibhavi Patel and Prof. ChitraBaggar, “A Survey Paper of Bellman-Ford Algorithm and Dijkstra Algorithm For Finding Shortest Path in GIS Application,” International Journal of P2P Network Trends Technology(IJPTT), vol. 5, February 2014

[18] Dechuan Kong ,Yunjuan Liang, Xiaoqin Ma, Lijun Zhang “Improvement and Realization of Dijkstra Algorithm in GIS of Depot”

[19] Ni Kai, Zhang Yaoting, Ma Yuepeng, "Shortest Path Analysis Based on Dijkstra's Algorithm in Emergency Response System," Indonesian Journal of Electrical Engineering and COmputer Science, vol. 12, no. 5,pp 133-139, 2015