INTELLIGENT COMPLEMENTARY RIDE-SHARING SYSTEM
(Fare Calculation)

Project ID: CDAP 19-055

Software Requirements Specification

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

Submitted on: 29th of April 2019

Declaration

I hereby declare that this is my own work and this document does not incorporate without acknowledgement 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 acknowledgement is made in the text.

Name	Registration Number	Signature
G.L.S.R. Gunawardena	IT16011380	

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

Supervisor Date

…… ……

Dr. Janaka Wijekoon

Table of Contents

LIST OF TABLES

LIST OF FIGURES

1. Introduction

1.1 Purpose

This document is to provide a detailed description of the “Fare Calculation” component in “Intelligent complementary ride-sharing System”. In here, we mainly discussed the features and behaviours that includes what the functionality of the component, how to achieve the goals (functions) and target audience. In addition to that, this document illustrates user roles who interact with the final product and 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 very useful for the development team in future references.

1.2 Scope

In “Fare calculation”, the Component main goal is to predict the fuel consumption according to the vehicle, display an estimated cost before the ride and real-time fare calculation on the driver’s mobile. To predict the fuel consumption, we conducted a survey and collected some specific information about the vehicle as well as in here; we are going to use an algorithm. OBD II device used for the real-time fare calculation for each passenger.

The “+Go” product mainly based on the mobile application will achieve these goals.

1.3 Definitions, Acronyms and Abbreviations

ICRSS	**Intelligent complementary ride-sharing System**
SRS	**Software requirements specification**
FC	**Fare Calculation**
OBD	**On-board diagnostics**
API	**Application Programming Interface**
NIC	**National Identity Card**
RAM	**Random Access Memory**
GPS	**Global Positioning System**

Table 1.3.1: Definition. Acronyms, and Abbreviation

1.4 Overview

The rest of the document provides a full description of the component, discuss the functional requirements with user interfaces and use case scenarios, specific requirements, proposed product, describe the non-functional requirements, hardware specifications and references.

A full description of the project is to discuss in the first section of the document. In the second section, focus on functional requirements with user interfaces and use case scenarios. Specific requirements, proposed product and non-functional requirements are discussing in chapter three of the document. The final part of the report mentions supporting information such as hardware specifications and references.

2. Overall Descriptions

The main goal of our project is to reduce traffic congestion in urban areas in ICRSS. In this proposed solution, make mobile platform application among office crowd then users can facilitate the share their vehicle. Therefore, traffic congestion will reduce due to less number of vehicle enter into urban areas. The passenger has to pay some amount to the vehicle owner's service that amount will calculate the before the ride as estimated.

The complete overview of the component, "Fare Calculation" is described broadly throughout this section. In this section, discuss the primary goals of the element, how to achieving these goals the connectivity between other modules, all the interfaces required during the implementation etc.

2.1 Function Perspective

During the period of observing the existing product in the similar domain found several existing ridesharing application in the Sri Lanka market [1,2,3]. Some of them are failing due not to fulfil customer needs and requirements. We were able to identify the existing application features and shortcomings that data using for the improvement of our proposed system. Features of existing products and brand new features are combined with our proposed system that will more productive for society. Society can get the maximum benefit from the FC component due to introduce new features that discussed below table.

Features	**UDIO**	**Carpooling.lk**	**RideShare.lk**	**Proposed Solution (ICRSS)**
The system will decide the estimated cost before joining the trip.	✓	✗	✗	✓
Vehicle fuel cost calculated according to the condition of the vehicle. (Engine Capacity, Manufacture year, Registered year, mileage etc.) [4]	✗	✗	✗	✓
Passengers can get off in any place where is the between source and destination because the price will calculate passenger travel distance	✗	✗	✗	✓

Table 2.1.1 - Feature Comparison Table

2.1.1 System Interfaces

"+Go" is an Android-based mobile application that developed using the android studio. MySQL and SQLite databases are using for the retrieve information. In here MySQL used as a cloud database. Some interfaces are designing for the FC component that interface use for synchronizing the communicate with front end and back end.

2.1.2 User Interfaces

There are several interfaces have proposed for this process and the main user interfaces are,

Image 2.1.1 Passenger Request to the ride - Passenger

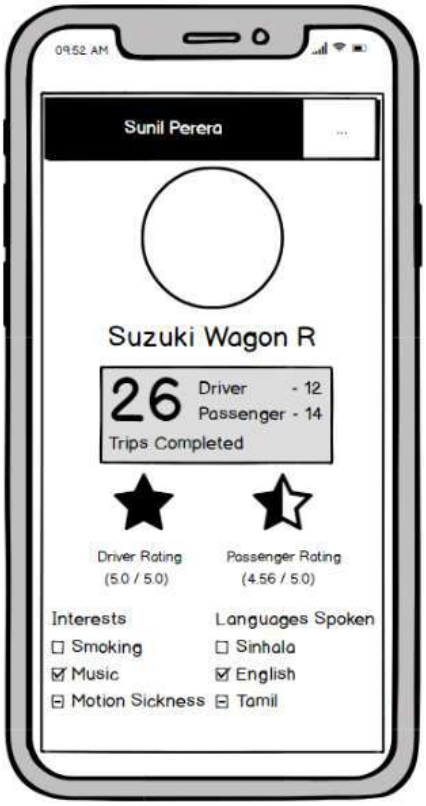

Image 2.1.2 View user details – Driver and Passenger

Image 2.1.3 Driver accepts the request which passenger sent to the ride - Driver

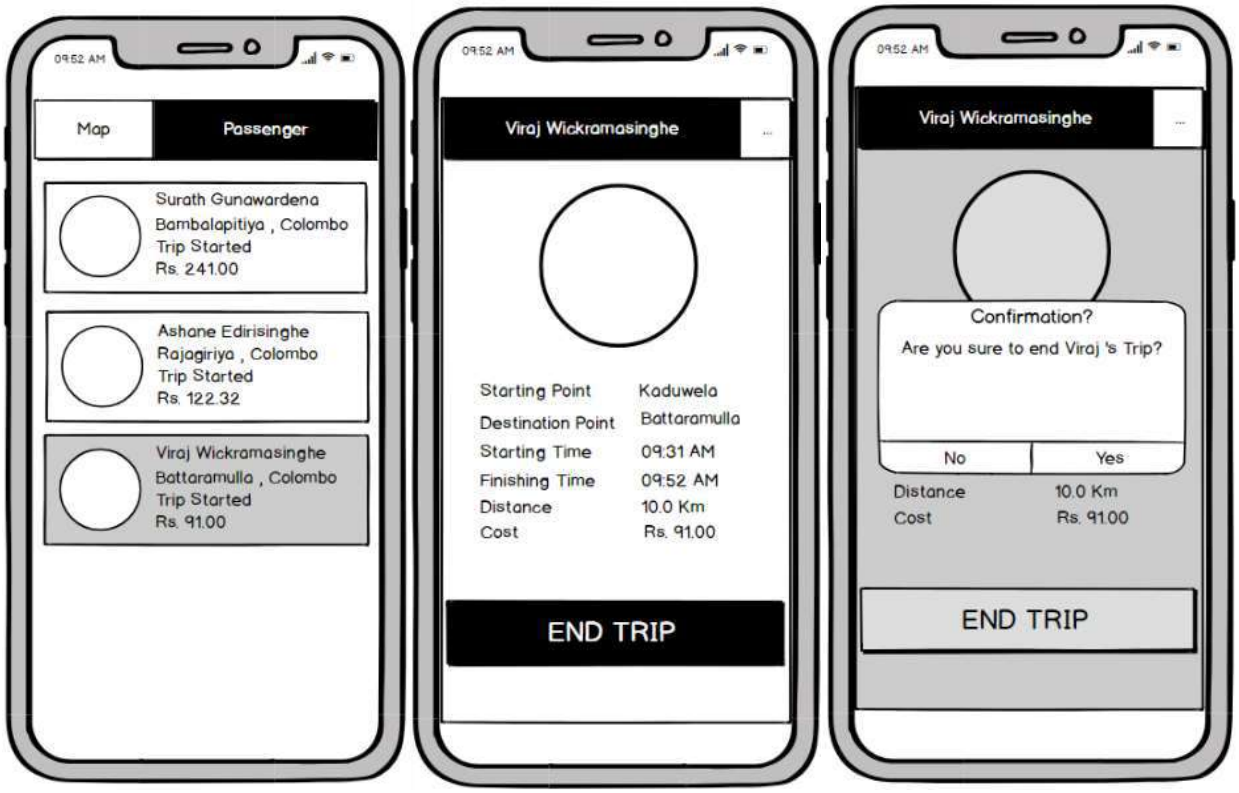

Image 2.1.4 End Trip- Driver

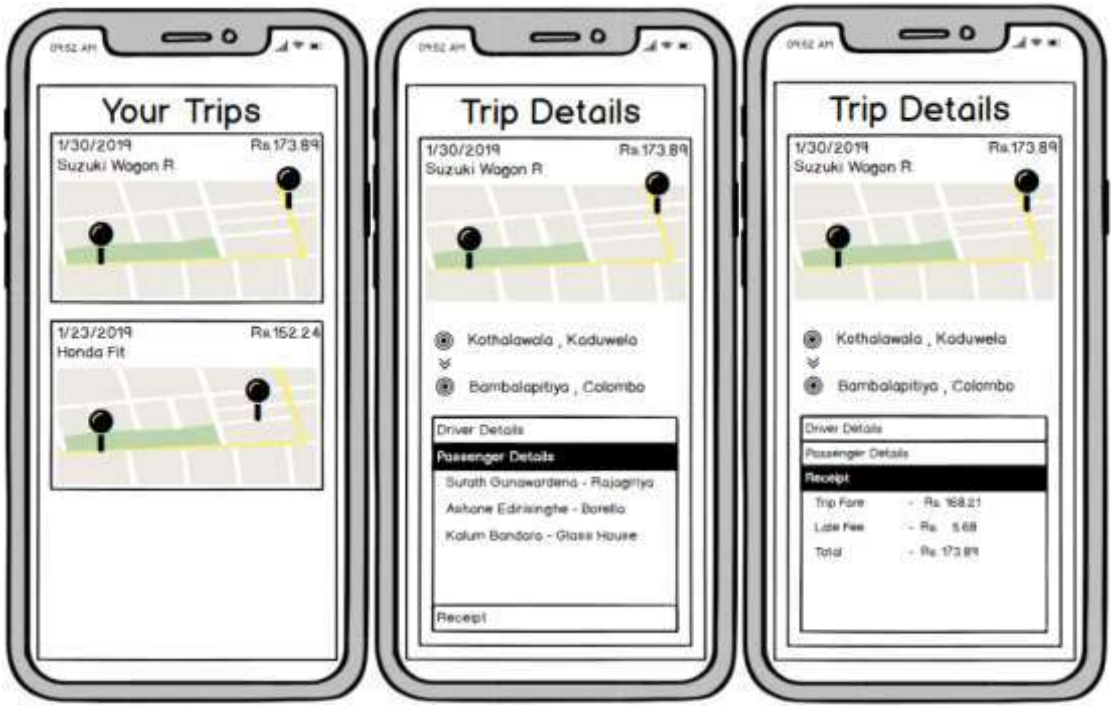

Image 2.1.5 Trip History – Driver/Passenger

2.1.3 Hardware Interfaces

To use this “+Go” product user need to properly plug into OBD adapter into the vehicle’s OBD port before starting the trip and also user needs to have proper Internet facility. In addition to that, the GPS module will be used for the identification of the current location of the passenger when estimating the fare for the journey.

2.1.4 Software Interfaces

The main software interfaces used in the FC component are,

Android Studio	For the development of mobile application
Genymotion Emulator	Emulating purposes
MySQL	Cloud Database Management
SQLite	Embedded Database in android application
Python	To implement the backend algorithms
Express js	For Development of web API

Table 2.1.4.1: Software Interfaces

2.1.5 Communication Interfaces

Internet connection is an essential thing for the communication between the mobile app and web server. As well as, Bluetooth facility will use for the between mobile and OBD Scanner.

2.1.6 Memory Constraints

- Android version should be 6.0 or higher
- 1 GB Ram
- 100 MB Memory space

2.1.7 Operations

The user operations can be categorized by the main four components in the system. 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

2.1.8 Site Adaptation Requirements

English used as a Supported language of the interfaces of the application.

Users have to permit to access the mobile internet connection, GPS location and Bluetooth connection to access the "+Go" Application.

This Application compatible, if and only if android OS version 6.0 or latest version. Otherwise, have to update the OS.

2.2 Product Functions

In this component, there are main three functions. They are,

- Predict the fuel consumption according to the vehicle

  To achieve this, develop an algorithm using several parameters (factors) such as Vehicle Manufacture Year, Registration Year, Engine Capacity, Number of cylinders, Power of the engine and mileage. [4]

Input	Vehicle Manufacture Year, Registration Year, Engine Capacity, Number of cylinders, Power of the engine and mileage
Output	Predicted Fuel Consumption
Process	Input data will be analyzed by an algorithm

Table 2.2.1: Predict the fuel consumption according to the vehicle

- Display estimate cost before the ride

  After the enter source and destination from the passenger, then the application gives the suggested driver list with an estimated cost to reach the destination.

Input	Fuel Consumption, Distance, Estimate travelling time without Traffic, Estimate Travelling time with Traffic, Current Passengers, Current passenger drop off details
Output	Display estimate cost with the suggested driver
Process	Input data will be analysed by an algorithm

Table 2.2.2: Display estimate cost before the ride

- Real-time fare calculation

  Fare calculation update in every 30 seconds for every passenger who is currently travelled. OBD Scanner use for the calculate distance passenger travelled.

Input	Distance, Waiting time, Fuel consumption, Number of passengers, Price of one litre of fuel
Output	Display Real-time fare on driver view
Process	Input data will be analysed by an algorithm

Table 2.2.3: Real-time fare calculation

Use case Diagram

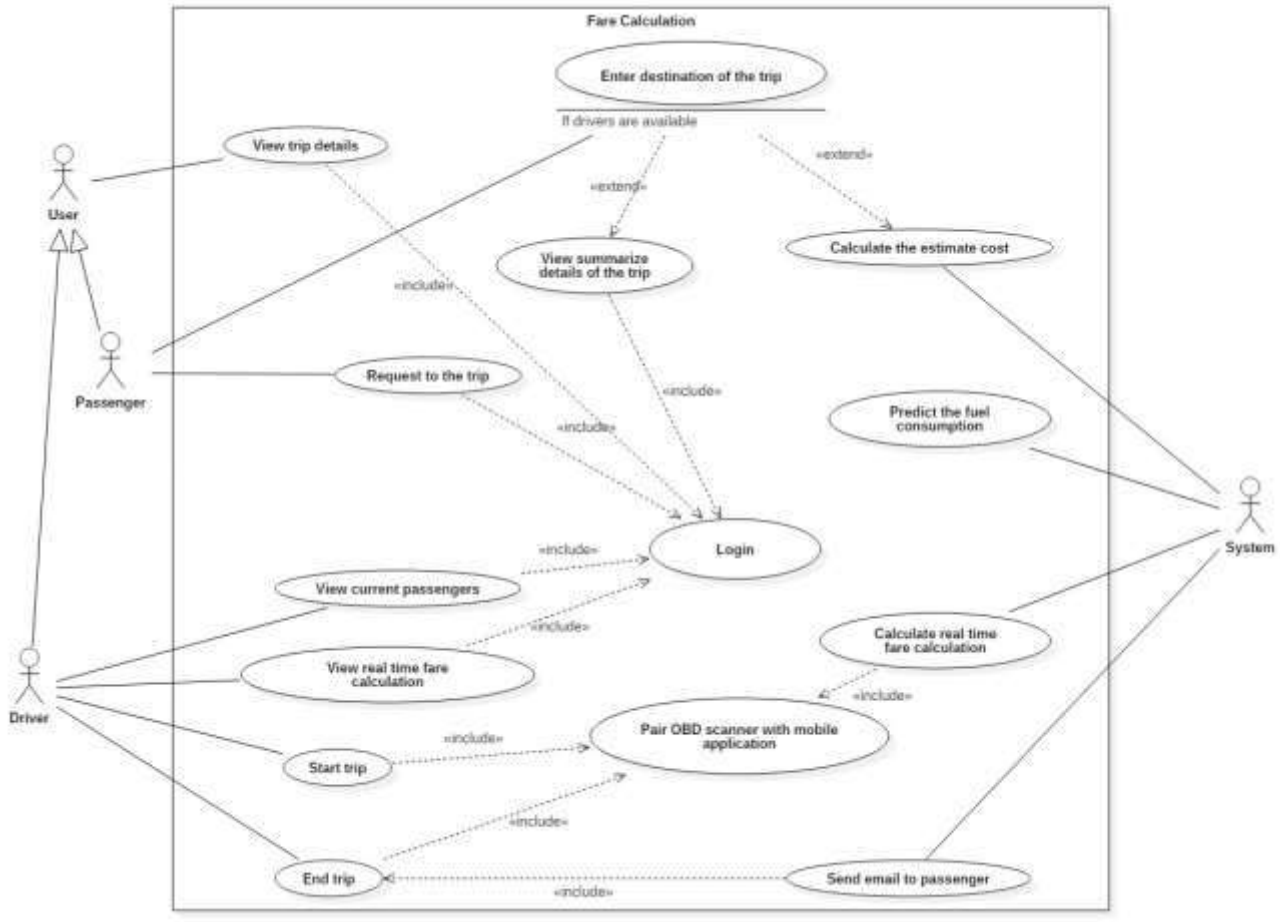

Image 2.2 Use Case Diagram

Use Case ID	FC001
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 2.2.4 - Use Case 01

Use Case ID	FC002
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 2.2.5 - Use Case 02

Use Case ID	FC003
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 2.2.6 - Use Case 03

Use Case ID	FC004
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 2.2.7 - Use Case 04

Use Case ID	FC005
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 seconds 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 2.2.8 - Use Case 05

2.3 User Characteristics

"+Go" application is developing for traffic minimization in urban areas. "+Go" is mainly focused on the office crowd who is daily travel to the urban areas. User should have understandability of how to use smartphone/application and knowledge of the English language.

2.4 Constraints

- User smartphone should be Android OS
- A Smartphone required with marshmallows version or latest version
- Should be a stable mobile internet connectivity
- Should not be an issue in Bluetooth accessibility in smartphone

2.5 Assumptions and Dependencies

- The smartphone is switch on throughout the journey as well as it has considerable battery life.
- The smartphone is on throughout the trip with stable internet connectivity.
- The Bluetooth connectivity is enabled when the user acts as a driver.
- GPS module is enabled in the mobile phone
- OBD port of the vehicle is supported for the OBD Scanner (OBD II ELM 327)
- "+Go" Service provided only within the Colombo district

2.6 Apportioning of Requirements

This main requirement of the FC component in this system is to calculate the fare for the individual passengers who join the trip. Section one and two describe the primary functions of the FC. In the first section, the full description of the project. In the second section, focus on functional requirements with user interfaces and use case scenarios. Specific requirements are presenting with the OOP methodology of the proposed product, and non-functional requirements are discussing in chapter three of the document.

3. Specific requirements

3.1 External interfaces

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
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 3.1.1.1 – 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 3.1.1.2 – 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 3.1.1.3 – 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 3.1.1.4 – 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 3.1.1.5 – View trip history as a passenger

3.1.2 Hardware Interfaces Integrations

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
    h. Bluetooth 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

3.1.4. Communication Interfaces

In FC, the process of communication happens through internet connectivity. It is preferred to have at least a 3G connection for the best data synchronization. The 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

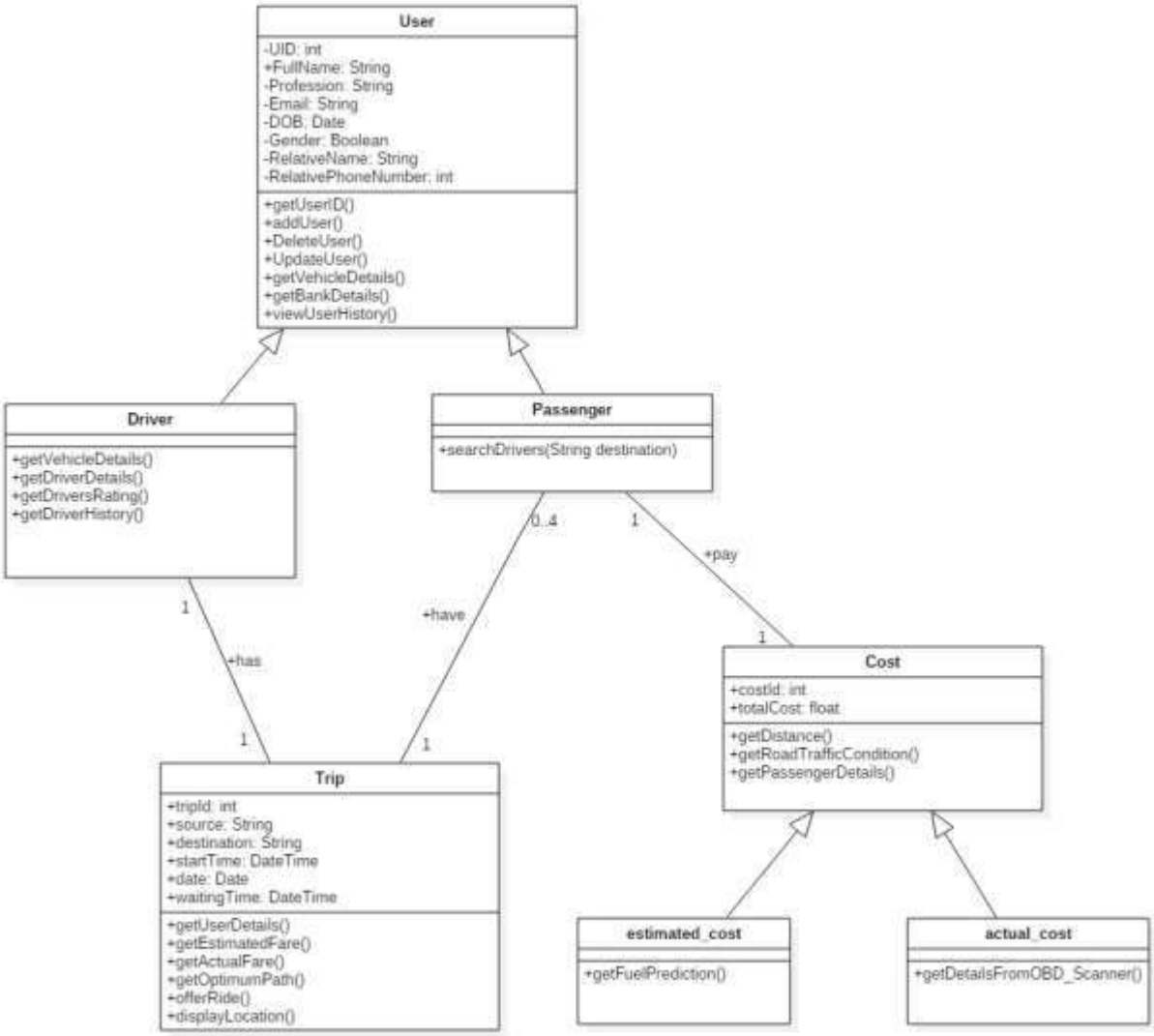

Image 3.2. Class Diagram

3.3 Performance requirements

Mobile phone's which is an application installed RAM; battery life depends on the performance of the proposed system. As well as has a stable internet connection and good Bluetooth connection between the mobile application and OBD Scanner make a better performance.

3.4 Design constraints

After surveying more than 150 office staff in Sri Lanka, As a result of a survey, we can identify the necessity of the ride-sharing application. Most of them have chosen ridesharing the practical solution for the traffic in the morning and evening time. In here we provide a mobile app which is the functionality of the ridesharing with user-friendly interfaces. As a result of user-friendly users can profoundly engage with this application to use this service.

3.5 Software system attributes

3.5.1 Reliability

Reliability is a vital thing in any system. When executing the system functionality should do the smoothly with any very less amount failure or without failure. When estimating fuel calculation according to the vehicle using multi-valued linear regression that estimation supported data collecting from the survey among the vehicle drivers and owners. Estimated FC will generate to the passenger after search the destination that based fuel estimation which previously mentioned, expected travel distance, current traffic condition, price of a litre of fuel and number of passenger currently joined on the trip that factors can get minimum different with real-time cost calculation. In real-fare calculation, to get actual travelled distance from OBD scanner. This FC component reliability affects to the whole the system.

3.5.2 Availability

Mobile Application and database server should be available 24x7. User can access to "+Go" application, if able to provides internet connection; otherwise, you will not be able to access to the app because application connects with the cloud database system. FC component's function should test whether execute correctly without bugs or not.

3.5.3 Security

We required to collect some sensitive data of the users these are very affecting for their privacy. So we have to apply the hashing method to that information then storing to the database as well as have to consider the database / Server security because of all sensitive data of the users and other required information stores in there. In addition to that, never should not expose privacy details because we are collecting user's NIC image and driving license image when registering.

3.5.4 Maintainability

When implementing the system maintain proper code commenting and indentation that is very easy to identify what developer has done previously and that can use future implementation. Not only that, proper coding standards and naming conventions are very helpfully for every team members. If we follow the above facts, easily maintain / Update our product at any failure as well as add new features to the system.

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

Annex 1: Vehicle Information Data Sheet – Sample

1	Brand Name	Model Name	Manu.Year	Reg.Year	Transmission Ty	Fuel Type	Engine Capacity	No of Cylinders	Milage	Fuel Consumption
2	Suzuki	Wagon R	2018	2018	Automatic	Mild-Hybrid	650	4	7837	17.3
3	Suzuki	Swift	2018	2019	Automatic	Petrol	1197	3	251	12.3
4	Toyota	Axio	2014	2015	Automatic	Petrol	1500	4	38008	18.2
5	Toyota	Prius	2011	2013	Automatic	Petrol	1800	4	138009	20.6
6	Suzuki	Wagon R	2018	2019	Automatic	Mild-Hybrid	650	3	5337	21.4
7	Toyota	Vitz	2017	2018	Automatic	Petrol	1000	3	29000	14
8	Suzuki	Wagon R	2018	2018	Automatic	Mild-Hybrid	650	4	25000	18
9	Toyota	Axio	2015	2017	Automatic	Full-Hybrid	1500	4	30000	25
10	Suzuki	Wagon R	2018	2018	Automatic	Mild-Hybrid	650	3	3373	24.3
11	Suzuki	Wagon R	2018	2018	Automatic	Mild-Hybrid	650	3	8452	24.1
12	Suzuki	Wagon R	2017	2017	Automatic	Mild-Hybrid	650	3	35859	24.8
13	Nissan	X-Trail	2015	2017	Automatic	Full-Hybrid	2000	4	56231	10.8
14	Toyota	Premio	2018	2018	Automatic	Petrol	1500	4	4032	11.3
15	Suzuki	Wagon R	2018	2018	Automatic	Mild-Hybrid	650	3	1683	26
16	Suzuki	Wagon R	2016	2017	Automatic	Mild-Hybrid	650	3	84558	21.8
17	Suzuki	Wagon R	2016	2017	Automatic	Mild-Hybrid	650	3	84558	21.8
18	Nissan	N17	2008	2008	Manual	Petrol	1300	4	60103	13.2
19	Suzuki	Alto	2006	2006	Manual	Petrol	796	3	137373	13.4
20	Suzuki	Wagon R	2015	2017	Automatic	Mild-Hybrid	650	3	19232	17.3
21	Suzuki	Alto	2015	2015	Manual	Petrol	796	3	44087	17.6

Throughout the SRS, we have elaborated the solution that we presented to overcome the traffic congestion in morning hours. For that, FC plays a significant role in the implementation.

4.2 References

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

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

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

[4] Sugathapala, T. (2015). *FUEL ECONOMY OF LIGHT DUTY VEHICLES IN SRI LANKA THE BASELINE*. [ebook] pp.5,6,57. Available at: https://www.globalfueleconomy.org/media/461037/asia_fuel-economy_sri-lanka_baseline.pdf [Accessed 20 Feb. 2019].