INTELLIGENT COMPLEMENTARY RIDE-SHARING SYSTEM

(Optimum Path Recognition)

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 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
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 05

LIST OF TABLES 05

1. Introduction 06

1.1 Purpose 06

1.2 Scope 06

1.3 Definitions and Abbreviations 06

1.4 Overview 07

2. Overall Descriptions 07

2.1 Product Perspective 08

2.1.1 System Interfaces 09

2.1.2 User Interfaces 09

2.1.3 Hardware Interfaces 11

2.1.4 Software Interfaces 11

2.1.5 Communication Interfaces 12

2.1.6 Memory Constraints 12

2.1.7 Operations 12

2.1.8 Site Adaptation Requirements 13

2.2 Product Functions 13

2.3 User Characteristics 17

2.4 Constraints 17

2.5 Assumptions and Dependencies 18

2.6 Apportioning of Requirements 18

3. Specific Requirements 18

3.1 External Interface Requirement 18

3.1.1 User Interfaces 19

3.1.2 Hardware Interface 22

3.1.3 Software Interfaces 22

3.1.4 Communication Interfaces 22

3.2 Classes/Objects 23

3.3 Performance Requirements 24

3.4 Design Constraints 24

3.5 Software System Attributes 24

3.5.1 Reliability 24

3.5.2 Availability 25

3.5.3 Security 25

3.5.4 Maintainability 25

3.6 Other Requirements 26

4. Supporting Information 26

4.1 Appendices 26

4.2 References 26

LIST OF FIGURES

Figure 2.1.1 Driver offer a ride 09

Figure 2.1.2 View Booking Summary ……………………………………………………......10

Figure 2.1.3 Passenger finds a ride ………………………………………………………......10

Figure 2.1.4 Report traffic jam and accidents……………………………………………….11

Figure 2.2 Use case diagram…………………………………………………………………..17

Figure 3.2 Class Diagram of OPR ……………………………………………………………23

LIST OF TABLES

Table 2.1.1 - Comparison of Existing Products……………………………………………….08

Table 2.2.1 - Use Case 01……………………………………………………………………….13

Table 2.2.2 - Use Case 02……………………………………………………………………….14

Table 2.2.3 - Use Case 03……………………………………………………………………….14

Table 2.2.4 - Use Case 04……………………………………………………………………......15

Table 2.2.5 - Use Case 05………………………………………………………………………16

Table 3.1.1 – Offer a ride Interface………………………………………………………….....19

Table 3.1.2 – View Booking Summary Interface………………………………………………20

Table 3.1.3– Passenger finds a ride Interface…………………………………………….20

Table 3.1.4 – Report traffic jam and accident Interface………………………………….....21

1. Introduction

1.1 Purpose

The main objective of this SRS document is to outline the requirements and provide a detailed description of the component “Optimum Path Recognition” in “Intelligent Complementary Ride –Sharing System” project. This document provides the software overview of the related component. SRS document includes the functional and non-functional requirements of the component, features and the goals of the component and project approach etc. Therefore, this document will be useful for the users, developers and other stakeholders to get an overall idea about our proposed software product.

1.2 Scope

In this proposed system, Dijkstra’s Algorithm and Travelling Salesman Problem (TSP) Algorithm will be applied to identify the closest path with least traffic, which connect source and destination while minimizing the traffic congestion. Dijkstra Algorithm find the shortest path by considering weight of the edges in a weighted graph, Depending upon the user’s option ,weight of the edges will be vary as distance or time according to the applicable conditions in it. Similarly,Travelling Salesman Problem(TSP) is another algorithm to identify the least weight tour,which cover all the nodes of the graph.According to the algorithm,it will identify a certain unique cycle in that graph and travel through each node of the graph for once with a minimal price [1].

1.3 Definitions and Abbreviations

SRS	**Software requirements specification**
API	**Application Programming Interface**
TSP	**Travelling Salesman Problem**
OPR	**Optimum Path Recognition**

1.4 Overview

The rest of this SRS document includes three sections and appendixes. A detailed description of the project is reviewed in the first section of this document.In second section of the document ,focus to overall overview of the functionality in the component and its interactivity among other components.Apart from that, this section provides functional requirements with user interfaces ,hardware interfaces, system constraints and use case scenarios about the product.The third section provides the specific requirements in detail and non functional requirements.The rest of the sections which organized in this document are hardware specifications and references.

2. Overall Descriptions

In optimum path recognition, we use both Dijkstra's Algorithm and Travelling Salesman Problem (TSP) algorithm to identify the closest path with the least traffic. Because of that, it helps to minimize traffic congestion in urban areas.Dijkstra algorithm is known to be as the algorithm which show the shortest path.Dijkstra’s algorithm is helpful to identify the optimal path of the tree by considering the root of the tree as the starting point and then expand the tree node-by-node [2].When considering the weighted directed graph,there is a shortest path node.So that node is beginning from the starting point and extend to the earliest smallest point.Therefore,point where entire nodes are adjacent to is called as the smallest point and the length of the arc is known as the chord length [3].

Relevant to the weighted directed graph, vertices of the graph show the cities while edges of the graph emphasize the distances between two cities which cohere by a road.As a result of that,this algorithm will be helpful to identify the closest path between a city and other entire cities [2,4].

Dijkstra's Algorithm provides the optimal path to reach the destination by considering the weight of the edges. According to the user's concern, the weight of the graph gets vary. If the user's concern is the distance factor, the length of the edges will be identified as the weight. Similarly, if the user's concern is about traffic, accidents or road closures, weight becomes the calculated time duration to cover the specified distance.

Travelling Salesman Problem (TSP) Algorithm provides the optimal path to travel through a city, which covers all the user-specified locations. When users set their locations, our supposed system will identify the order of the locations to visit. Then this algorithm finds 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 on the relevant path within a specified time range by uploading the pictures of the particular incident. We are introducing the Genetic algorithm to optimize the traffic.

2.1 Product Perspective

We have identified several ride sharing applications within the country during the initial stage.However most of the applications couldn't able to satisfy the users with their requirements.Since traffic congestion has become a huge concern in the society,we proposed a solution to minimize this problem by implementing a new ride sharing application.Here we have identified a research gap among some ride sharing applications by referring some literature surveys and other related findings.Therefore through our proposed solution we were able to satisfy the users needs by ensuring their reliability,security and some other essential factors.

Following table emphasize a comparison between the existing products and our proposed solution.

Features	**UDIO**	**Carpooling.lk**	**Rideshare.lk**	**+GO(Proposed System)**
Crowdsourcing to improve the optimum path by analyzing more than one algorithm.	✕	✕	✕	✓
Allowing the registered users to enter the live updates by uploading images.	✕	✕	✕	✓

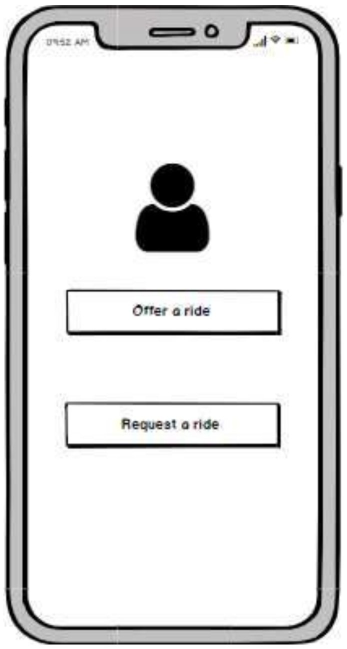
Table 2.1.1 - Comparison of Existing Products

2.1.1 System Interfaces

+GO is a ride sharing mobile application for the working people which will help to minimize the traffic congestion during office hours in urban areas.This mobile application is built on the android platform which will be helpful for developing the application.Mainly firebase can act in two ways as a database server as well as a web server.Therefore when act as a database server ,it will used as a realtime database.Meanwhile it will provide backend as a service.Therefore ,it will be known to be as BaaS.Therefore databases can access user interfaces to maintain and control the database section in an organized manner.

2.1.2 User Interfaces

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



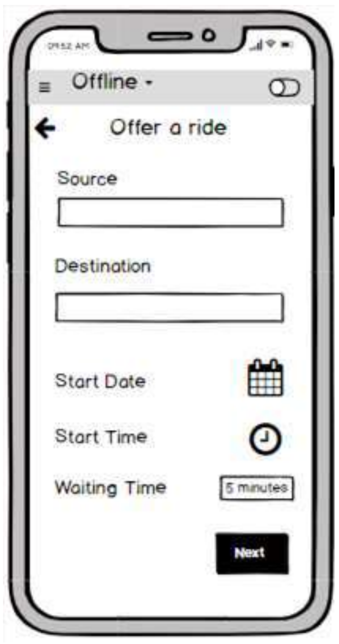


Figure 2.1.1 Driver offer a ride

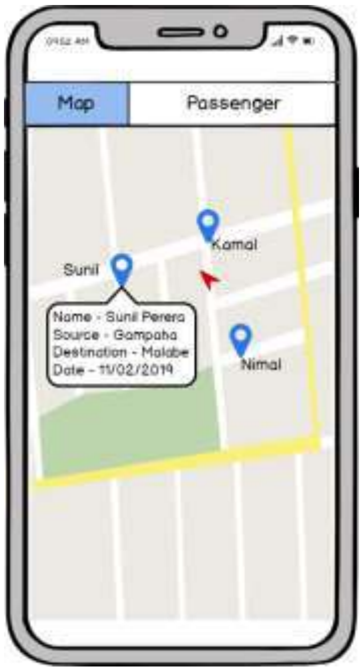


Figure 2.1.2 View Booking Summary

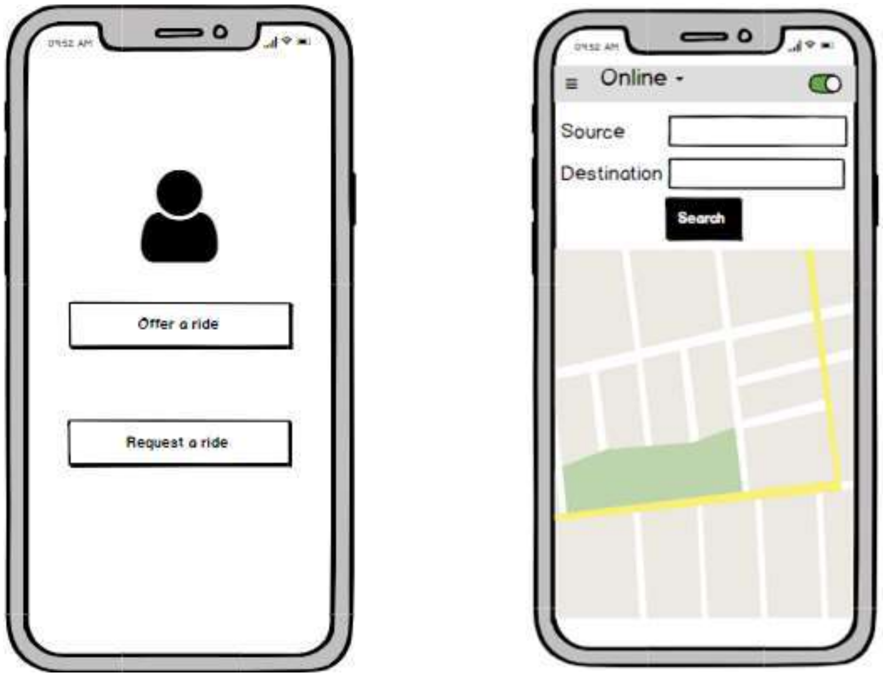


Figure 2.1.3 Passenger finds a ride

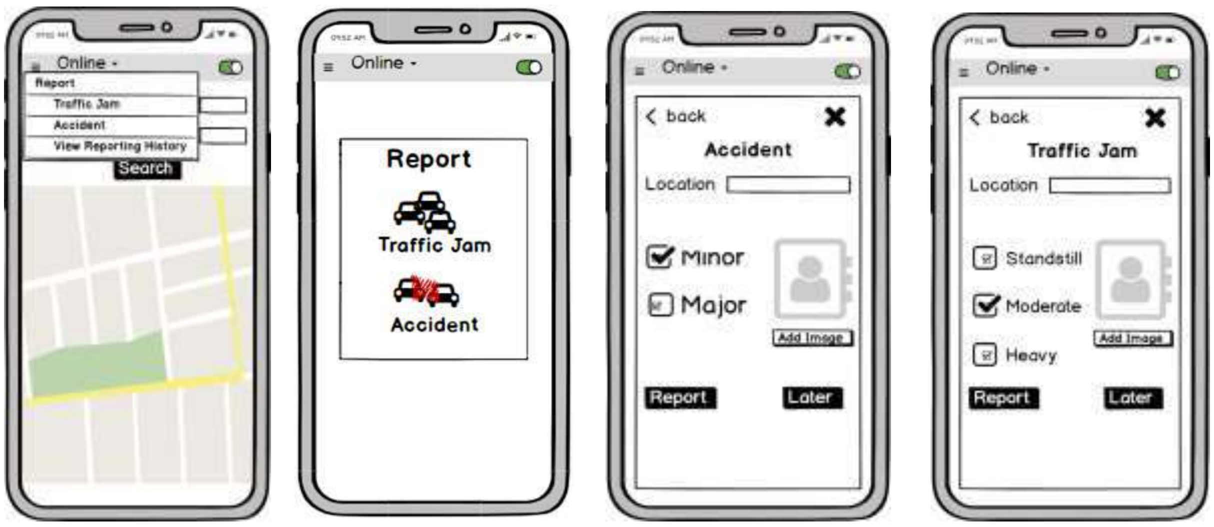


Figure 2.1.4   Report traffic jam and accidents

2.1.3 Hardware Interfaces

Mobile application is the solution in this proposed system.Therefore it will use only a small amount of hardware.Proposed system will have a mobile phone which should be complement with internet connectivity ,GPS and allow camera to capture the pictures using mobile phone.

2.1.4 Software Interfaces

- Android Studio – Used for developing the mobile application
- Firebase - Realtime Database
- Leaflet - Used as an open source javascript library for emphasizing inter maps
- Geolocation API - identify the current location/place on the map.
- Distance matrix API - identify the distance among cities on the map.
- Geocoding API - identify the addresses and convert them into geographic coordinates
- MySQL - Used as backend database server
- SQLite - Used as an in built data storage
- Python - Using for run the background algorithm
- Express.js - Using for development of web API

2.1.5 Communication Interfaces

Proposed system consist with four different components.Therefore communication between each an every component is very essential to make a huge interconnectivity while sharing the functionalities.Internet connection provided by modem will be used for the communication between mobile application and the web server.Better internet connection is required when loading the maps.

2.1.6 Memory Constraints

The android mobile application is needed,

- 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

The user operations can be categorized by the main four component in the system.All the operations belong to the "Optimum Path Recognition " are listed as follow.

1. 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.
2. 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.
3. Drivers can view all booked summary details of each passenger and display their drop offs in a map.
4. Drivers can add/report the traffic jam and accidents by entering the corresponding information with an image of the incident.
5. 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 will be used as the language in the interfaces of the application and for the notifications.Internet facility should be available for an effective communication.All user details and other related information will be stored by using a database.Firebase realtime database will be used to store the real time informations.

2.2 Product Functions

Use Case ID	OPR001
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 2.2.1 - Use Case 01

Use Case ID	OPR002
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 2.2.2 - Use Case 02

Use Case ID	OPR003
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 2.2.3 - Use Case 03

Use Case ID	OPR004
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 2.2.4 - Use Case 04

Use Case ID	OPR005
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 2.2.5 - Use Case 05

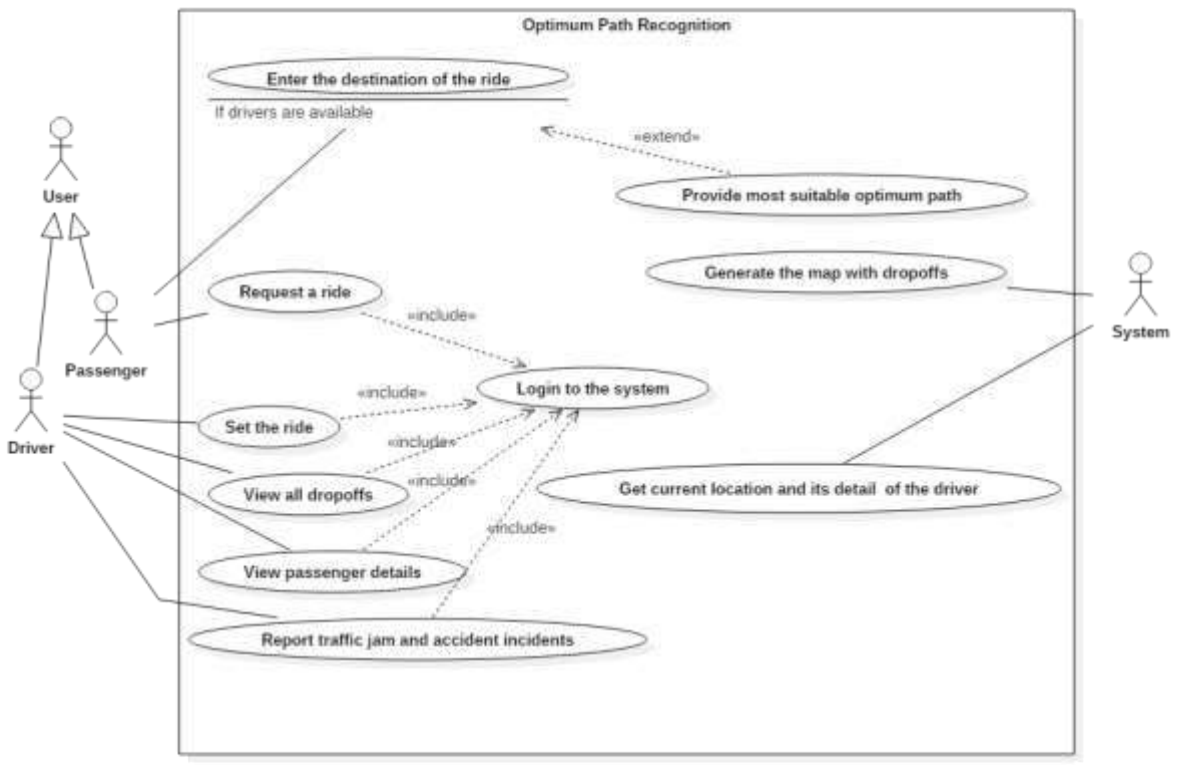


Figure 2.2Use case diagram

2.3 User Characteristics

This proposed product is mainly focused on working people. Therefore the most significant type of users for this component are mobile application users. Users should have a knowledge about the phones and English Language Skill in order to interact with the mobile application.

2.4 Constraints

- Users must have smartphones with a proper internet connection to load the results within an organized map.
- Android will be used for the mobile application development.
- Firebase used for the database configuration.
- User should be an authenticated user.
- Server should be able to handle multiple requests at a certain period of time without any constraint.
- User should have English Language knowledge.

2.5 Assumptions and Dependencies

- +Go mobile application is compatible with all the smartphones.
- Whenever the mobile application is running mobile data or Wi-Fi should be available.
- This application provide its service only in Colombo district area.

2.6 Apportioning of Requirements

Section 1 and 2 describe the primary requirements of this proposed project.Section 3 referred to as functional requirements and those requirements are used while designing and implementation processes. When considering all of these requirements ,they are meant to be consistent.In the meantime defects can be identified during the inconsistency period.First phase of the project will include all the significant requirements that mentioned in the section 3 ,which will helpful to create an organized inceptive plan.Once the significant requirements are developed and tested completely ,other remaining requirements will be used next.However we will be able to build a satisfied project by using these requirements .

3. Specific requirements

This section describes about all the functional and quality requirements of the optimum path recognition component.

3.1 External interface Requirement

This section contains all the inputs and outputs related to the component in a descriptive way.Other than that, it describes about software,hardware,communication interfaces and user interfaces too.

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**
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 1-4seconds	Show a dialog box with a proper message

Table 3.1.1 – 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 3.1.2 – 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 3.1.3– 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 are saved respectively

Table 3.1.4 – Report traffic jam and accident Interface

3.1.2 Hardware Interfaces

1. Smart Phone
    1. CPU: 1.6GHz or higher
    2. Storage: 200MB or higher
    3. RAM: 2GB or higher
    4. OS: Android API level 23 - Lollipop
    5. Front Camera: 8 MP
    6. Back Camera: 13 MP
    7. GPS Module

2. Server
    1. CPU: 2.6GHz Intel Core i7 (6th Gen or Higher)
    2. RAM: 16GB DDR3 or higher
    3. Storage: 1TB HDD
    4. 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

A better internet connection should be presence when developing this proposed mobile application.In this component ,communication will be occur through internet connection.For retrieving some realtime informations,firebase connectivity should be strong.

3.2Classes/Objects

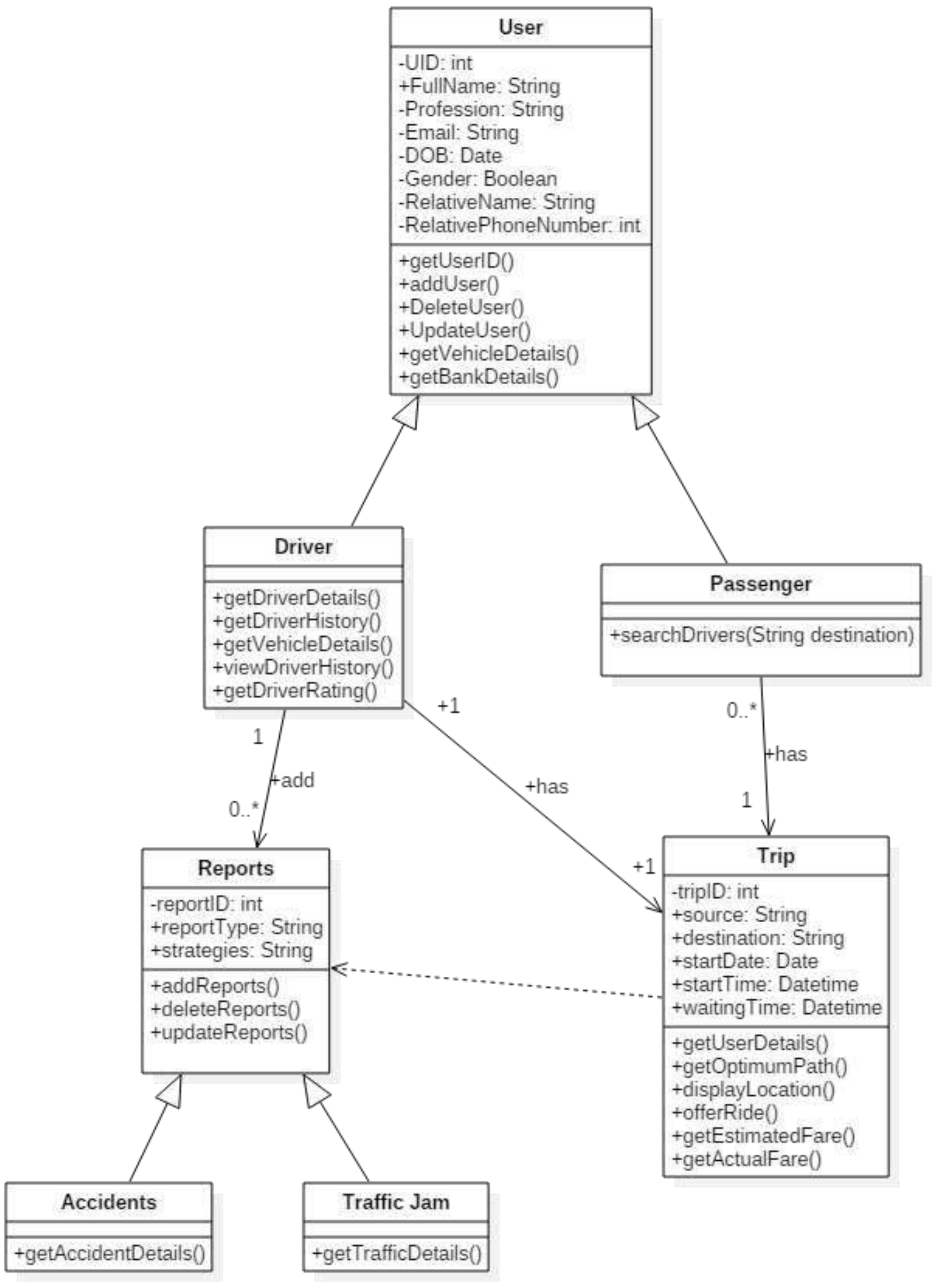


Figure 3.2 Class Diagram of OPR

3.3 Performance requirements

- Since the performance of the proposed product depend on performance of the mobile,it should have a proper internet connectivity.
- Backend should have an ability to handle multiple requests in a particular time period.
- It would be more effective if the user interface screens respond within 5 seconds.
- Meanwhile system will handle the accuracy of the information generated by the system.

3.4 Design constraints

According to this research component in the proposed product, developers should concern about some essential factors.Since this is used as an mobile application ,we should consider about the mobile application designing.As this is mainly focused in travelling system on office crowd among professionals,so it will be a huge diversity of audience.Because of these primary users ,user interfaces of the mobile application should be simple and attractive.It will be more user friendly when the interfaces are in light color.As a result of that people will be able to interact with this proposed product easily.

3.5 Software system attributes

In this section, it will describe about some attributes which provide through this system.The explanations are as follows.

3.5.1 Reliability

Reliability of the proposed product is more significant,since it should have the ability to perform its functions with less failures. It would be more reliable , if the system have backups to prevent occurrences of the failures.

- The reliability of analyzing and identifying the most suitable optimum path to a certain destination will be acquired by using Dijkstra’s Algorithm and Travelling Salesman Problem(TSP) Algorithm.
- The reliability of providing the information regarding the traffic jam and accident incidents will be obtain by using crowdsourcing.

- Reliability of the proposed system can be improved by dispensing the essential details of various users without any faults.
- All the required information should be validated before appearing in to the databases.

3.5.2 Availability

Availability of the system can be considered as whenever the users have the accessibility to the system .However the system should allow multiple users to engage in work concurrently and should be able to deliver the requested services without any extensive delays.Most of the details related to this component are stored in firebase.Therefore if the registered users reset their mobile phones,users are capable of recover their past data through the firebase database.

3.5.3 Security

Security of the proposed system means providing the conservation to the input data.System should have an ability to outlast the unauthorized users.Therefore all the authenticated users should be capable of using the mobile application.

- Since the users provide their private data to the system,security of the users data should be in excessive process.
- Entering password like features can be stored in the database in an encryption version.
- Security of the proposed system can be maximized by maintaining and controlling the server side.
- Firebase database provides the real time database security.

3.5.4 Maintainability

Whenever the failures occur in the system,it should be able to repair and restore the system to a considerable reputation within a given period of time.Most of the algorithms which are used in this component will be used in backend.Therefore the maintenances can be done in ease by using the backend server.

3.6 Other Requirements

Other than above mentioned requirements,following requirements will be more help helpful when implementing this proposed mobile application.

- User interfaces of the mobile application should be user friendly.
- Servers should be more reliable and strong enough to handle any issues.
- System should be more feasible.
- Interfaces are compared and analyzed with the prototypes.

4. Supporting Information

4.1 Appendices

Throughout the SRS documentation,we have elaborated the solution that we presented to overcome the traffic congestion in urban areas .For that,OPR plays a major role in the implementation.

4.2 References

[1] 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

[2] 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

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

[4] 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