INTELLIGENT COMPLEMENTARY RIDE SHARING SYSTEM

19-055

BSc(Hons) in Information Technology

Specializing in Software Engineering / Information Technology

Department of Software Engineering / Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

August 2019

INTELLIGENT COMPLEMENTARY RIDE SHARING SYSTEM

19-055

Dissertation submitted in partial fulfilment of the requirements for the Special Honors Degree of Bachelor of Science in Information Technology Specializing in Software Engineering / Information Technology

Department of Software Engineering / Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

August 2019

DECLARATION

We declare that this is my own work and this dissertation 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.
Also, we hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other media. We retain the right to use this content in whole or part in future works (such as articles or books).

Signature: _________________ Date:___________________
(V.A.Wickramasinghe)

Signature: _________________ Date:___________________
(A.E.Edirisinghe)

Signature: _________________ Date:___________________
(G.L.S.R. Gunawardena)

Signature: _________________ Date:___________________
(R.M.A.N.Gunathilake)

The above candidates have carried out research for the BSc Dissertation under my supervision.

Signature: _________________ Date:___________________

(Dr. Janaka Wijekoon)

ABSTRACT

Traffic congestion is a significant problem which causes chaos on the road and interruptions to regular tasks of the people. The primary concern is the number of vehicles daily entering the urban cities a high. Therefore, it will give lots of pessimistic effects on social aspects, environment and economically. Since most of the vehicles are coming from the same venue to defined region resulting vehicles are underutilized. Many people have proposed and implemented various methodologies to control this traffic flows. Therefore as a solution, we thought to introduce a new ride-sharing platform:+Go. In the initial stage, +Go platform is mainly focused on the office staff in Sri Lanka. +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. Moreover, our application uses computer vision methodologies to avoid fake registrations. It uses crowdsourcing platform to increase the accuracy of the suggested routes. The passengers, including the driver, have to share the fare for the ride and that is intelligently calculated according to fuel consumption of the vehicle, distance, the time taken to arrive at the destination. 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 ultimate purpose of this document is to provide a detailed description about +Go application and how it effects to reduce traffic congestion.

Keywords - Ride Sharing, Machine Learning, Image Processing, K-Means Clustering, Crowdsourcing

ACKNOWLEDGEMENT

We would like to extend my sincere gratitude to Dr. Janaka Wijekoon for the constant guidance throughout the project as the supervisor, Dr. Dharshana Kasthurirathna for giving constant feedback on our application, and everyone who provided their sensitive data to be used in the training and testing purposes of our application.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

UPM	**User Profile Management**
GDP	**Gross Domestic Product**
OTP	**One-Time Password**
API	**Application Programming Interface**
GPS	**Global Positioning System**
FC	**Fare Calculation**
OBD	**On-board diagnostics**
NIC	**National Identity Card**
ICRSS	**Intelligent Complementary Ride Sharing System**
DVPRM	**Document Validation and Profile Rating Maintenance**
RAM	**Random Access Memory**
REST	**Representational State Transfer**
XML	**Extensible Markup Language**
OPR	**Optimum Path Recognition**
TSP	**Travelling Salesman Problem**

LIST OF APPENDICES

1. INTRODUCTION

1.1 Introduction

According to the Report of World Population Prospects 2017 in the United Nations, by 2030, the Sri Lankan population forecast will be around 21,474,701. Hence, the demand for private transportation has also increased over the last few years[1]. Corresponding to the articles, surveys, and police reports, there is a gradual increase in vehicles that enter to Colombo during rush hours[2]. As a result of that vast traffic congestion has occurred in urban cities like Colombo city during office hours. Because of the increase in traffic congestion, the number of accidents also increases. According to the surveys conducted by the Central Bank, they have identified low occupant vehicles like cars that have been in high quantity and resulted in traffic congestion. Hence, this is a waste of financial resources, moreover polluting the environment severely. By considering some statistics collected from articles and reports, we identified that traffic is a significant concern in urban areas, especially during office hours. It has been stated that there is an Rs.500 million loss due to the daily traffic congestion in Sri Lanka[2]. Due to the heavy traffic in the rush hours, there is a massive loss of time wasted on the road. Therefore we came up with a concept of ridesharing application as a solution for this huge problem. The basic idea behind that was to combine professionals who are traveling to work by their private vehicles. This will lead to building the network among professionals. So this proposed ridesharing application will help to reduce stress and improve productivity while traveling as a passenger.

Hence, when one vehicle carries several numbers of people together, it will reduce the number of cars entering the city. To verify our hypothesis, we surveyed a sample of more than 150 office crowd, and we identified that the majority considered ridesharing as an excellent solution to reduce traffic in urban areas.

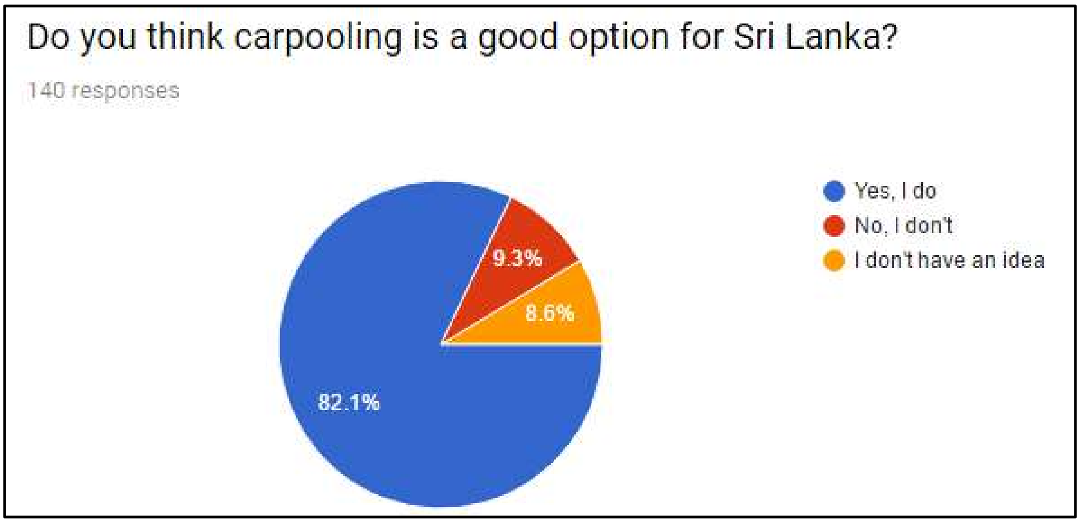


Figure 1.1.1: Suitability of ride-sharing in Sri Lanka

We were able to identify that the majority prefer to have a ride sharing platform in Sri Lanka. Also, the study made us obvious that people tend to use a ride sharing platform mainly to become a solution for traffic congestion.

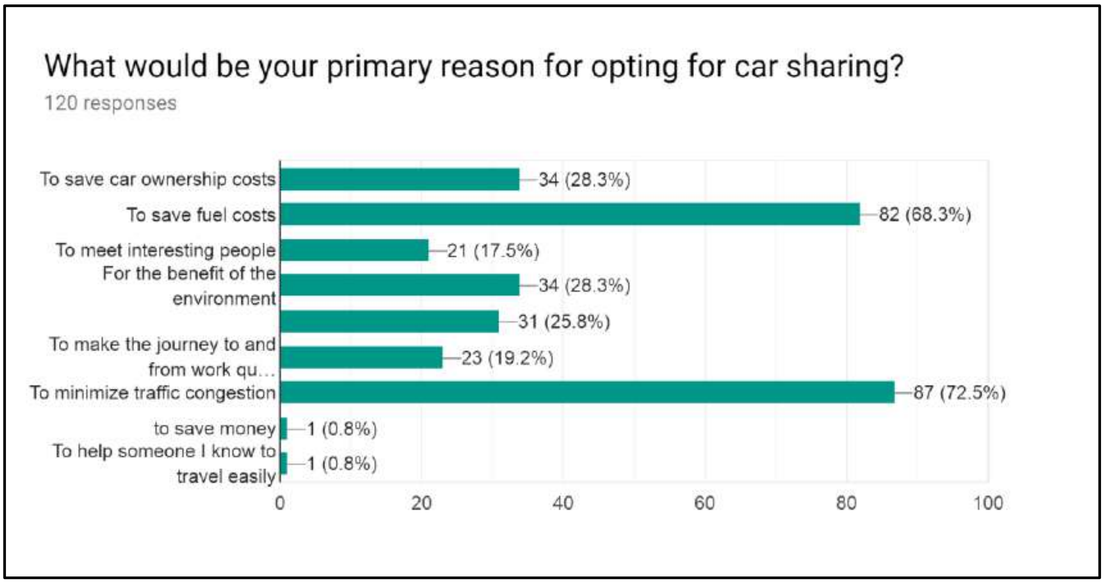


Figure 1.1.2 Reasons for opting Ridesharing

Furthermore, it became clear that though ride sharing is not much common in Sri Lankan context, that people are ready to collaborate in ride-sharing platforms. This was highlighted as 61.4 % of the crowd we surveyed fully interested to it.

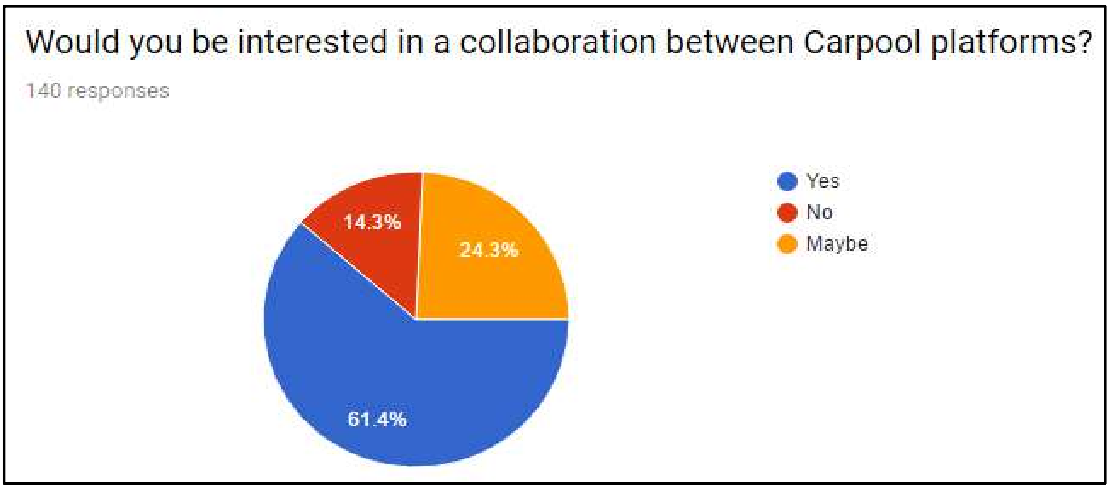


Figure 1.1.3 Interest in Ridesharing

Apart from the traffic minimization, this will provide some other traveling facilities to the users. When single occupants share their ride among other occupants, who go to the same destination, which helps to reduce the number of vehicles on the road which results in reducing the traffic congestion and minimize the emission of carbon dioxide (CO_2) to the environment, consequently, CO_2 emissions (metric tons per capita) in Sri Lanka was reported at 0.88555 in 2014 can be drastically reduced from this[3]. Resultantly it will minimize environmental pollution too.

Furthermore, we surveyed to get the real-time fuel consumption of the vehicles in the urban areas[4]. Our proposed system will predict the ride fare of the passenger before the ride starts and help to reduce the cost of traveling by distributing the total fare among all the passengers.

1.1 Background Literature

Ridesharing is considered as a possible solution to reduce traffic congestion[5]. Factors like cost of travelling, time, distance, ownership of the vehicle, personal preferences influenced in choosing the individual transportation mode[6, 7]. Certain studies have focused on the factors like fuel price, demography and safety measures in selecting the transport mode[8]. Manzini and Pareschi[9] introduced a new Decision Support System (DSS) for ridesharing systems. Passengers were given support to select the best car to be travelled. With our proposed method, we optimize the solutions provided by the previous studies by suggesting only the drivers, which match the preferences and professional level of the passenger.

Swati, Neha, and Ajita[6] did another research to make the ridesharing application user-friendly of both the driver and the passenger. From the point of algorithms, Fagin and Williams(1983) and later Ajtai et al. (1998)[10] proposed the algorithm to get a portion of users that willing to share their cars and from that study, they introduced a new scheduling algorithm. For the process of grouping individuals who matches with each other, was first done through using K-Means Clustering. Abdul and Sebastian[11] proved that the mean accuracy of using K-Means clustering gives you over 78% accuracy depending on the dataset, we are using. They also enhanced some features in the K-Means and proved that the new Enhanced version of the K-Means yield accuracy over 89%. From the study done by Chih-Ping Wei, Yen-Hsien Lee and Che-Ming Hsu to compare the most accurate clustering algorithm, they have proved that K-Means clustering yields precise comparatively results than the other clustering algorithms they used in the study[12]. From the literature, the number of clusters in a dataset is easily calculated using the elbow method from the study of Ketchen. and Shook[13], they have analyzed Elbow method returns accurate value depending on the dataset used. In this study, we will be using Elbow method to calculate the number of clusters and K-Means clustering to cluster the drivers and passengers.

To validate the images of NIC and License, it is needed to proceed with image processing. Valiente, Sadaike, Gutiérrez, Soriano, Bressan, and Ruggiero(2016) explains that uploading the images to a server and do the manipulations there is an optimum method[14]. This image manipulation process could be done by following the steps as loading the image, cropping Image, processing the image and then extracting information from the image[15, 16]. In the way, Chakraborty and Mallik(2013) explain, the colour images have to be converted to grayscale and then to binary using an adaptive threshold for an accurate recognition[17, 18]. According to Mordvintsev and Abid(2017)[19], image processing could be easily performed with the help of OpenCV library. It also provides the capability for face detection using Haar Cascade. With reference to Clark(2018)[20], the Python Imaging Library (PIL) has the ability to add image processing functionalities to the python interpreter, and it could be used in image archives, image display, and image processing. It was also identified that the processed images could be sent for recognition using Tesseract recognition engine which is considered as the best highly portable open source OCR engine currently available[14, 17, 21, 22].

Naive Bayes classifier is a simple method based on the Bayes rule which assumes that the presence of a particular feature independent of the presence of any other feature and contributes independently to the final probability[23, 24]. The literature says that an advantage of Naïve Bayes' is that it only needs a little amount of training data for the estimation of the parameters required in classification[25]. Further, it has been identified that the Naïve Bayes algorithm performs better in sentiment analysis when it is compared with algorithms such as Support Vector Machine (SVM) and Multinomial Logistic Regression[26]. According to the comparison between Naive Bayes, K-Nearest Neighbour and Random Forest Algorithms did by Baid, Gupta, and Chaplot(2017); suggests that the Naïve Bayes algorithm gave the best accuracy with the accuracy of 81.4%, while others give the accuracy of 55.3% and 78.65% respectively[27]. Further, Vidushi and Sodhi(2017) have emphasized that the Naïve Bayes algorithm outperforms the KNN algorithm[28][29]. By considering all the facts in the literature, we decided to use Naive Bayes in analyzing the reviews given by the users in our ride-sharing platform.

Since the general objective of this research is to minimize traffic congestion in urban areas, the identification of the optimum path is very significant. Dijkstra algorithm is known as the shortest path algorithm. Dijkstra Algorithm is used to identify the shortest path of the tree by considering the root of the tree as the starting point and then expand the tree node-by-node[30]. According to the weighted directed graph, there is the shortest path node. For the matter of fact, that node is starting from the starting point and extend to the earliest smallest point. In here, the point where entire nodes are adjacent to, is known as the smallest point, and the length of the arc is called the chord length[31]. According to this weighted directed graph, vertices of the graph emphasize the cities while the edges of the graph show the distances between two cities which adhere by a road. Therefore, Dijkstra's algorithm can be used to identify the closest path between a city and other entire cities[32, 33].

Similarly, Travelling Salesperson Problem (TSP) is another algorithm to identify the least weight tour, which covers 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[34]. Some algorithms have been developed to solve TSP. Adewole(2011) have been proposed Genetic Algorithm for solving Traveling Salesman Problem[35]. From the study of Dweepna Garg and Saurabh

Shah(2011), they used some other method to solve Travelling Salesperson Problem, which is known as Ant Colony Optimization[36]. Tunon and Lopez (2005) proposed the algorithm of Branch and Bound to solve Travelling Salesman Problems[37]. Later on, Pragya, Dutta and Pratyush(2015) succeeded to identify another method to solve the Travelling Salesperson Problem, which is known as Dimensional Ant Colony Optimization (DACO)[38]. Find the shortest route by using the Travelling Salesman Problem based on Dijkstra's algorithm is more accurate since the algorithm extricates the distance from Google Maps [39, 40].

The fare calculation is considered as the business logic of this system. In a previous study done by Zoepf, Chen, Adu, and Pozo(2018), it was identified that the fare has to be calculated according to the vehicle type along with a minimum cost. Furthermore, it is based on time and distance added to the fare as dynamic factors. The driver has to bear all the expenses associated with the vehicle operation including depreciation, insurance, maintenance, repairs, and fuel which varies from driver to driver, or from vehicle to vehicle[41]. From the study of Santos and Xavier(2013), riders can decide how much the passenger is willing to pay for the ride. Then the system computes the cost based on the current fuel price and fuel consumption for the ride. Next, the system suggests a driver according to the willingness to pay the amount[42]. According to a study of Riquelme, Banerjee, and Johari, they identified two methodology for pricing .those methodologies are called as static and dynamic pricing. The static pricing method assigns a fixed amount for all the drivers who are in the platform. Hence, the price does not change based on the instantaneously available service capacity. These parameters are slightly different across the day (Even most taxicab services price in the evenings are different from time to time). Importance of the static price is that it doesn’t affect to the instantaneous state, but it will only react to the course changes. They have derived an equation for dynamic pricing where their equation enters the list of available drivers, base price, distance and time and as a result, the dynamic price is calculated[43]. Considering the above literature regarding the fare calculation, there is no fixed way to calculate the fare for a single passenger. In our proposed solution, we will be using Multiple Linear Regression method to predict the fuel consumption, which results in estimating the fare for the ride. From the study of Aleksandar, Silvana and Valentina(2015), they have proved Mean Absolute Percentage Error(MAPE) of using Multiple Linear Regression in their study is 3.0730601 using a trained set of data. From that we have identified the error percentage of using regression model in critical analysis is comparatively lower than the other regression models and we'll be using it for our fare calculation procedure to produce an accurate result [44].

1.2 Research Gap

There are several ride sharing applications in the market. However, according to the literature survey and other findings, we identified a research gap that they do not address major factors, which need to be addressed using a ride-sharing platform to ensure the reliability and the experience of the user. The following table implies a comparison of features between existing products and our proposed solution.

Table 1.3.1 - Comparison of Existing Products

Feature	Uber	UDIO	Carpooling.lk	RideShare.lk	+Go
System focused mainly on office staff	✗	✗	✗	✗	✓
Matching the passengers' profile with the suitable drivers	✗	✗	✗	✗	✓
Suggestion of drivers per passenger interests	✗	✗	✗	✗	✓
Consider gender preference when registering to provide high security	✗	✓	✗	✗	✓
Validating the user by processing and comparing the images of both NIC and license in real time	✗	✗	✗	✗	✓
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 fare before requested the trip.	✓	✓	✗	✗	✓
Vehicle fuel cost calculated according to the condition of the vehicle.	✗	✗	✗	✗	✓
Dynamic fare calculation using OBD II adapter	✗	✗	✗	✗	✓
Crowdsourcing to improve the optimum path by analyzing with multiple algorithms	✗	✗	✗	✗	✓

Our application only focused on professionals and no any other ride-sharing application use this concept. Also, existing solutions in Sri Lanka, suggest any driver to a passenger, and we have taken a step forward in making a new advanced method to suggest drivers who match with the job profile, and the preferences of the passengers. This reduces the number of dissatisfactions from the point of both the driver and the passenger. Another important factor to be mentioned is that our application has the option for passenger's guardian or spouse to track the trip history of the fellow passenger and report any driver whom they think suspicious, and those drivers won't be suggested that passenger again. Out of all the features introduced in UPM, only UDIO has the feature to enable and disable the gender preference, and it is taken as a measure to ensure security on the point of the user.

Validating the user by NIC and license will reduce the fake registrations, and this will eliminate manual validations in the system. Our application allows users to write their own reviews at the end of trip, those sentiments gets analyzed using machine learning algorithm and a proper rating will be given to that sentiment. Hence, our solution maintains separate ratings for user's behavior and vehicle. At the end of the journey, users can rate the driver, vehicle and the co-passengers separately. Also the passengers are given the choice of blocking the drivers,so that they won't be suggested in future sessions.

1.3 Research Problem

According to the statistics from the Road Development Authority, there is the number of private vehicles entering the Colombo Municipal Council (CMC) is increasing over the past three decades[45]. Due to that reason, traffic congestions are increasing as well as the numbers of accidents are increasing. As a result of the traffic congestion, people are getting delay to reach their destinations, increased fuel wastage and monetary losses. That impacted the whole development process; hence, the government has been lost over Rs.500 million daily [2].

To reduce traffic congestion, people need public transportation instead of private vehicle because of the huge number of crowd travel at once. There are some issues in Public Transportation, such as not comfortable because it is overcrowded, too expensive if we get on air-conditioned buses and so on[45]. Therefore, we thought of introducing a ride-sharing app, which could become a **solution to traffic congestion**. The basic idea behind that was to combine professionals who are travelling to work by their private vehicles. Ability to reduce the number of vehicles using the ride-sharing application because of one vehicle carries several people together.

1.4 Research Objectives

1.4.1 Main objectives

The main objective of our study was to develop a solution which can minimize traffic congestion during office hours in Sri Lanka. With the application we proposed targeting the professionals, it can reduce the travelling costs and environmental pollution as well. Our objective was to reduce traffic congestion while providing the same comfortability level, expectations and professional level to the users. By introducing a new ride-sharing application to professionals, it is indeed accomplished. Hence, our app has a significant impact on society.

1.4.2 Specific objectives

1.4.2.1 User Profile Management

In UPM, our main objective was to introduce a new ride-matching algorithm to the users. Ride matching algorithm works according to the profession and preferences of the user. Hence, we needed to collect user information like gender preference, smoking condition, like quietness, motion sickness, language preference, vehicle comfortability, and profession for that. After receiving that information, the new ride-matching algorithm was developed using rule-based machine learning and K-Means clustering. In the algorithm, it is needed to get the current location of the driver, and for that polling, the technique was implemented where every 5 seconds, the current position of the users were recorded in the database. Also, a new algorithm was developed to find the reported drivers based on the actions taken by the spouse or the guardian and removed them from the suggested list. Implemented the OTP system for the mobile number verification of the users to confirm the identity validation.

1.5.2.2 Document Validation

The specific objective of document validation is to validate the driving license and NIC cards, and identify the NIC number and expiration dates using an image processing algorithm and minimize the risk of fake profiles getting registered in the system. In this we identify and check the compatibility of driving license and National Identity Card (NIC) with relevant to the NIC number and also consider the most significant components in them to verify their validity.

1.5.2.3 Profile Rating Maintenance

The objective of profile rating maintenance is to identify the response of the drivers and passengers concerning other passengers who joined the trip and rate the people correspondingly. To achieve this, system has to detect some keywords, which were selected by the users regarding their experiences in the ride sharing. Further this system classifies the reviews given by the users by using a sentiment analysis algorithm (Naïve Bayes). A feature of blocking drivers also have given for the passengers to provide the freedom to filter the suggestions of our application. Hence, this component does help to ensure the security of the passengers and drivers as well.

1.5.2.4 Optimum Path Recognition

The main objective of optimum path recognition is to identify the closest path. In this proposed system, Dijkstra’s Algorithm and the Travelling Salesman Problem (TSP) Algorithm will be applied to perceive the closest path with least traffic, which connect source and destination while minimizing the traffic congestion. Dijkstra Algorithm find the shortest path by considering the 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. Since we enable the registered users to enter the live updates by uploading images, it will be more helpful to implement crowdsourcing to predict the most efficient route.

1.5.2.5 Fare Calculation

The Specific objective of fare calculation is to give reasonable cost to both drivers and passengers. While the ride with passengers meet the several segments. If new passenger joins the trip or passenger end up the trip, new segment will be created. Each Segment calculate fare for the passenger that calculation depends on cost for fuel consumption within the segment, the number of current passengers, and the current price of the fuel in Sri Lanka. Used OBD II adapter to get fuel consumption of the vehicle. Equation:2 Calculate the fare for the Segment. Final fare of the ride is calculated by adding the fare of all the segments which passenger has travelled.

2. RESEARCH METHODOLOGY

2.1 Methodology

In our system, we have divided the functionalities mainly into 4 parts including UPM, DVPRM, FC and OPR. In the fig 2.1.1, overall high level view of the components are discussed.

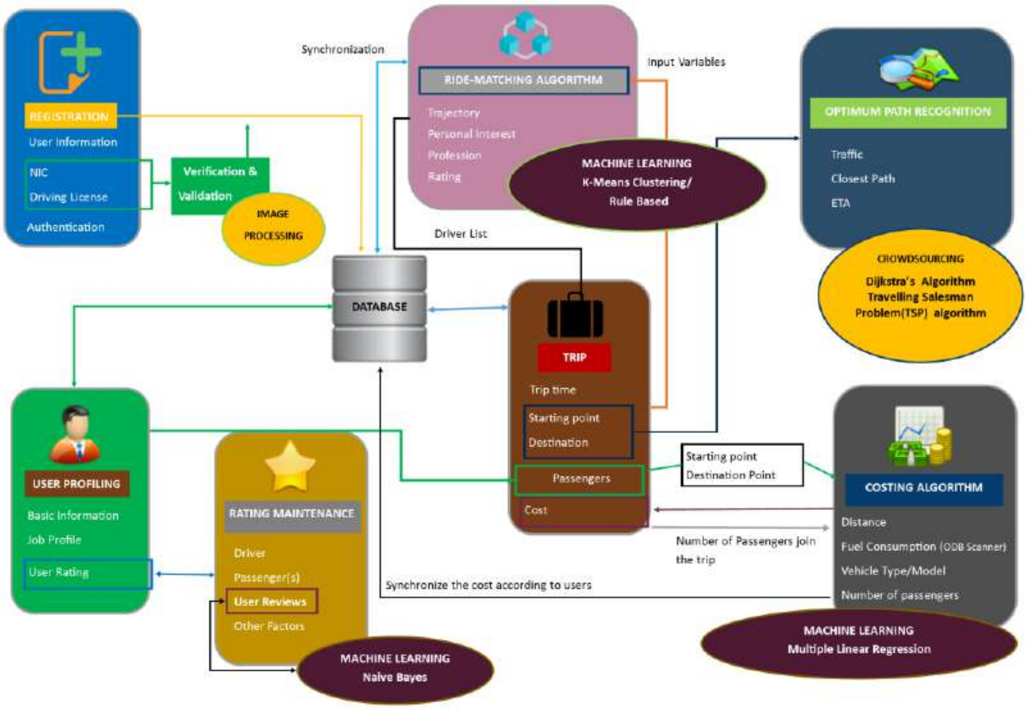


Figure 2.1.1 Overall System Diagram of +Go

2.1.1 User Profile Management

User Profiling Management is considered as the core functionality of the +Go application. The main idea behind the UPM is to suggest the most suitable drivers for a passenger. In our study, we have divided the UPM into 2 phases.

- User Profiling
- Ride matching algorithm

In User Profiling, we collect the basic information, job profile, preferences like gender preference, smoking preference, and language preference, like quietness, motion sickness and ratings. At the initial stage, we verify the mobile phone number of the user to ensure the trustworthiness in the application. Users who are unable to verify the phone number or creating an account with the same phone twice are not allowed to proceed further in our app. It always communicates with the rating class to keep a track on the ratings, and the passenger details are taken from the trip class. For each registered user, a separate record is maintained in the database. All the results are sent to the database and that information is used in the Ride Matching algorithm.

Initially, identified several important activities that need to be carried out in the UPM. For frequently, UPM is used by the user (driver or passenger) as well as a spouse or the guardian. All the activities carried out by those users are given from the below use case diagram.

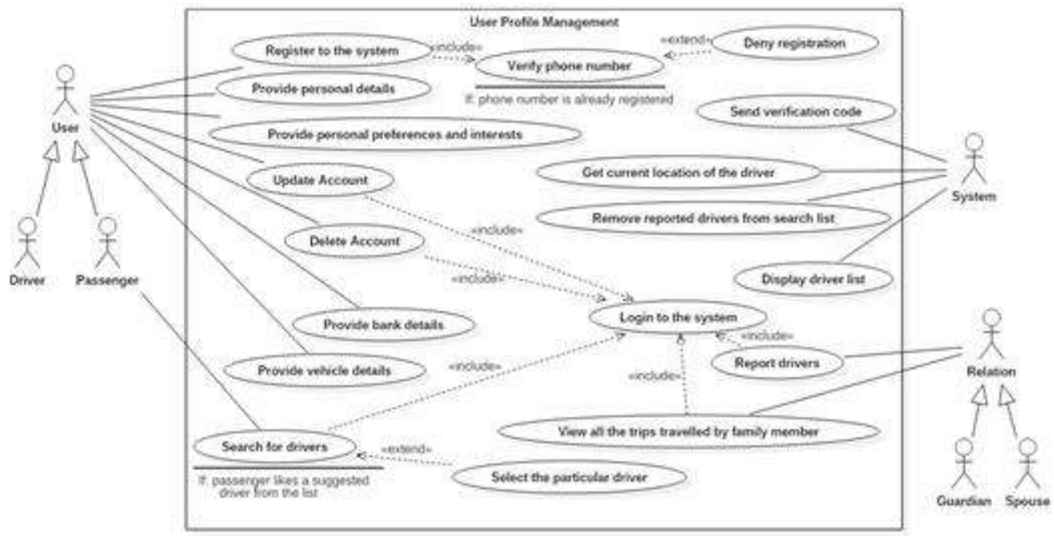


Figure 2.1.2 Use case diagram of UPM

We introduced a new algorithm to suggest the most suitable driver list to the passengers upon searching for the destination. Ride Matching algorithm always synchronizes with the database to keep a track on the latest data to be processed. Primary inputs to the algorithm are the personal details, trajectory, time, date, personal preferences, ratings and more importantly, the profession. Initially, we have defined a set of rules using rule-based machine learning, to select the drivers who match with the preferences of the passengers.

After filtering out the drivers, the selected list is sent to the clustering. It is essential to highlight that we have used the polling technique to update the location of users at every 5 seconds into the real-time database. The purpose of it was to identify the exact location of the drivers before suggesting them to the passenger.

Figure 2.1.2 Polling the location

In here, we have defined a model to identify the drivers who are in the range of 5 Km from the starting location of the passenger. Drivers who are beyond those limits are neglected automatically.

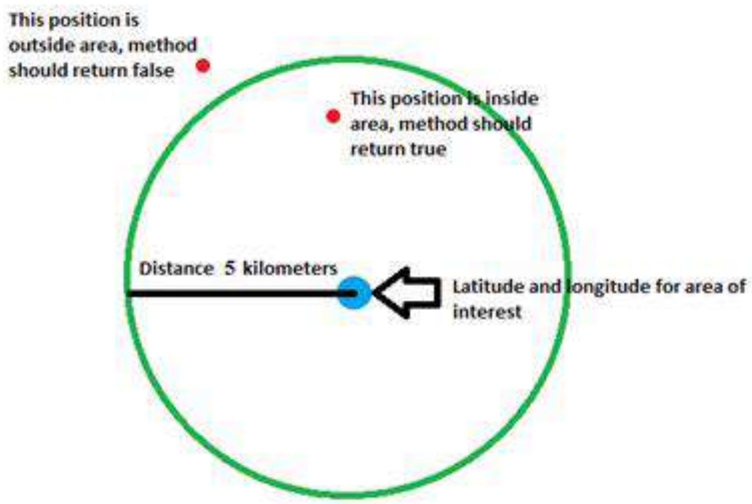


Figure 2.1.3 Distance Comparison

When all the above requirements are fulfilled, those drivers are used for the clustering. Clustering is used mainly to group the drivers based on the professional level and age. In here, we suggest only the drivers whose professional level equal to or higher than the level of the passenger. To identify the professional level, we have defined a set of levels in the backend of our application. For clustering purposes, we decided to use K-Means clustering, which is simple but gives a more accurate result than other clustering algorithms for our datasets. To identify the number of clusters in the given context, we used the Elbow method. Technologies chosen were decided after a systematic study about the existing products and literature. From the elbow method, the number of clusters was identified as three according to the below diagram.

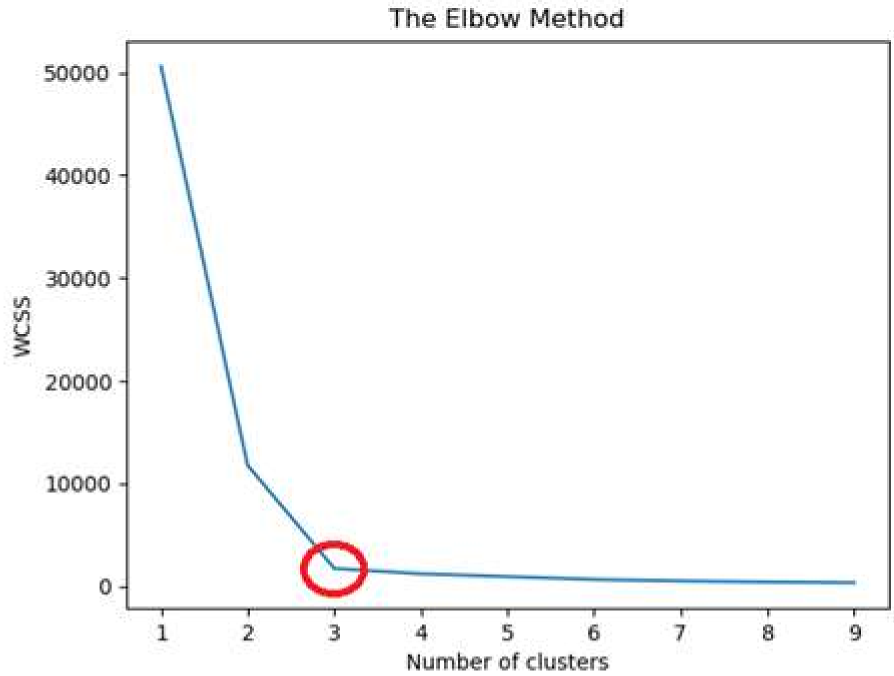


Figure 2.1.4 Identifying the number of clusters

For the clustering Profession was chosen as the X-axis and Age were selected as the Y-axis. The main reason for selecting the Profession as the Y-axis is that, we cluster according to the profession. After clustering, the cluster which includes that relevant passenger is taken out. In UPM, spouse or the guardian too can check the trip history of the passenger.

If they feel any driver is suspicious, they can directly report that driver through the interface given to them. Hence, we remove those particular drivers from the suggested list of that specific passenger from future suggestions. Below diagram would demonstrate how suitable drivers are selected for a specific passenger upon searching for the destination. It explains the rules defined when filtering the drivers like within the 5 Km range to passenger's starting point etc.

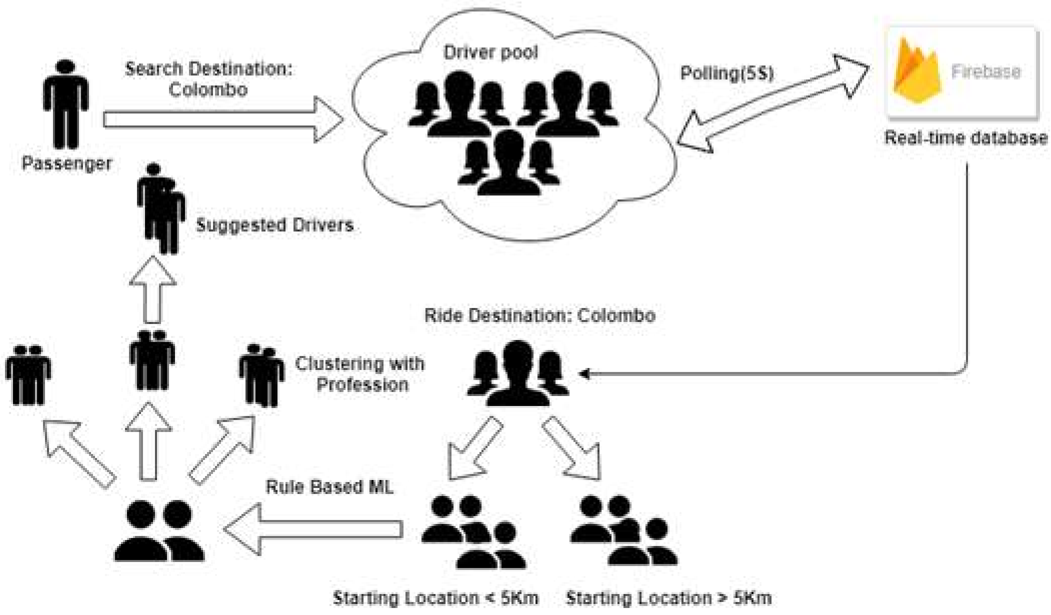


Figure 2.1.5 Flow of the Ride matching algorithm

The system diagram which summarizes the whole architecture of the proposed Intelligent Complementary Ride-Sharing System; specifically the Document validation and Profile Rating Maintenance component can be identified as follows.

In the phase of user registration, both the passenger and the driver is considered as similar user roles. The user is asked to upload the front image of national identity (Electronic or non-electronic) card in Sri Lanka and after successful extraction of data from the image, asked to upload the front image of the license card. Hence the image is verified and necessary information is extracted; also the NIC numbers extracted from both are compared to make sure that they belong to the same user. At instances of failing to process the images uploaded, user is asked to re-enter a clear image, or else the user does not get the chance to proceed. This is done to avoid any spammers getting register to the system.

In our system, users can behave as a passenger or as a driver at different times. At the completion of the trip session, the passenger is asked to rate his/her experience.

He/She can simply rate with 5 is everything is good; the Driver, Vehicle and the Co-Passengers will get the default rating 5. The passenger can give some compliments to the driver as well. If the rating is below five, the passenger is asked to specify which made them the journey uncomfortable. Further, they are allowed to write their own

review as well and the system will identify the user experience accordingly. Drivers to get the chance to rate and review on the passengers at the end of each trip session.
The overall overview of the functionality of this component would be much clearer from the following use case diagram.

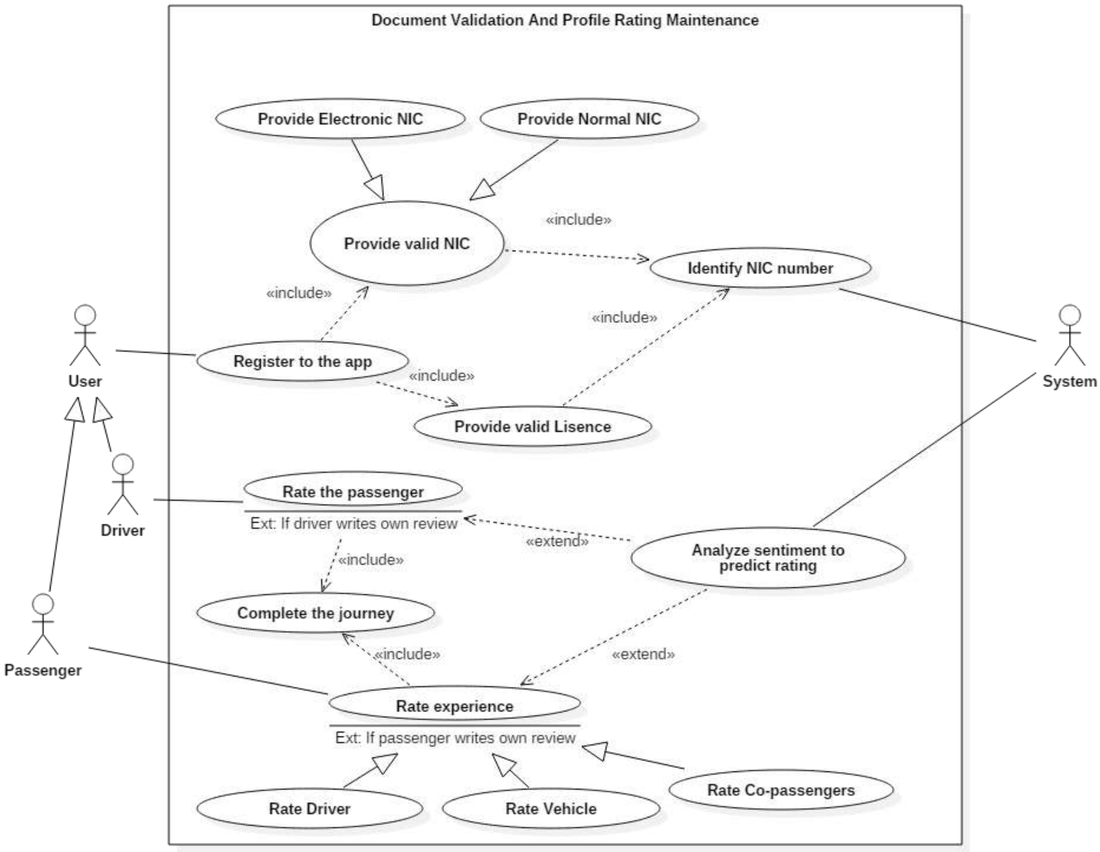


Figure 2.1.2 Use Case Diagram of DVPRM

The detailed description of how the process is continued as described below.

2.1.2 Document Validation

The total process of validating the NIC card and license card can be segmented as follows. In the implementation of this component we used Php(v7.2.19), Python(v3.6.8), Android(v6.0 or higher). Also, the libraries OpenCV(v4.1.0), Tesseract(v4.0.0-beta.1) does help a lot in the image processing and text extraction functionalities respectively.

1) Loading Image

Using the android mobile application, the users are given the chance to capture the image or select an image from the gallery. Once the user selects and upload, the image gets saved in the web server with a unique id. For this implementation of image uploading, we have used PHP as the programming language.

2) Pre-processing Image

The accuracy of the extracted text from images highly depended on the pre-processings done one the image. Both NIC and License card proceed through a few common steps of preprocessing as well. Those images are first subjected to identify a valid photograph of a human face. This has been achieved with the haar cascade algorithm; openCV library connected with python does help to achieve this easily. To use the haar cascade algorithm, it is needed an XML file which has a proper configuration on face detection. In the development, we used the file provided by Intel as an open-source resource[46].

After successful face detection, the images are subjected to colour change functionality in order to eliminate the unwanted colours in the image which cause distractions in text extraction. In this process, we changed the darker colour pixels to red and lighter colours to white with the expectation of having proper segmentation of text from the rest of the image. The RGB values used in the above process were different from each other methods used in images of the license, electronic NIC and non-electronic NIC. And those RGB colour values were decided based on the results of the test images.

In the process of license images, the images had to move through a process of enhancing the images by sharpening, adjusting brightness and contrast, before the colour-changing process mentioned above, as they contained too much color complexities with respect to NIC card images.

Next, the images are converted to grayscale to reduce the complexity of the image which has to be processed. Erosion and dilation are performed on the image to reduce the noises of the image, and also Gaussian blur filter is applied to make the image more suitable. Finally, the image is converted into a binary image as we identified that they give results with better accuracy.

Figure 2.1.1.1 - Raw, grayscale and binary image parts of non-electronic NIC respectively[47]

3) Content Extraction

After the above steps have been completed, the images are suitable to be used in text extraction with the help of Tesseract library. In order to minimize the time taken to extract the final output, we run the text extraction algorithm on raw, the colour changed and fully processed image stages separately, merely because some images do not need to be pre-processed for text extraction as they are high in quality.

The text extraction algorithm runs initially on the uncropped image as some users upload images with a very little amount of unwanted background. If it fails to extract the required text, then the images are cropped by considering in extracting needed section as a ratio of the given image. After successful extraction of text in the image, the extracted text is filtered for unwanted characters and letters. In the context of non-electronic identity card or license images with 9 digit nic number; they are converted to the format of 12 digit, which is the standard of the nic number in the electronic identity card.

If the images uploaded by the user are not clear enough to be processed with the algorithm, they are asked to upload a much clearer image again. At the instances where the extracted characters are partially correct, using the mobile platform the user is allowed to make changes to 18% of the extracted characters. But we implemented this feature as a unique way.

As an example, if only two digits of the NIC number extracted at the process of text extraction in the national identity card, is different from the actual NIC number of the user, they are given to change those two digits. After the user makes the changes, that changed number is saved in the application, and then at the process of identifying NIC number in the licence, the users are not given the chance to edit the extracted text; but compared with the NIC number recorrected by the user.

2.1.3 Profile rating maintenance

The rating of the user profiles directly affect in grouping the users of same type. So in our application, maintaining a better rating is important to have a comfortable experience. In the implementation of this component we have used Python(v3.6.8), Android(v6.0 or higher), node js(v8.10.0), MySQL.

This rating activity can be done by the users either as a passenger or as a driver.

If it is a passenger,at the end of the session, the android mobile platform does direct them to the interface of rating; which lets them to rate the overall experience. They can either select a rating or simply skip that process. If the rating given is 5; means that fully comfortable, they are given the opportunity to write their compliments to the driver. So that these compliments are displayed in the driver's profile.

At the instances where the rating is below 5; which means there is a kind of dissatisfaction, the passenger is asked whether the driver, the vehicle ,the co-passengers, or all of them caused for the dissatisfaction. Once selected the category of dissatisfaction, they are given a set of keywords which explain their situation the best. They also can write their own reviews if the keywords provided are not explaining their situation. The passengers are given the chance even to block the driver in future suggestions.

If the rater is a driver, he is also given the chance to rate the passenger at the time the session ends. He can rate the passenger, and he too gets a set of keywords to describe the passenger and also to write a review if the keywords are not satisfying.

The reviews given in the above mentioned situations are analysed with the naive bayes algorithm, which is a popular machine learning algorithms used in sentiment analysis.

The sentiment analysis procedure in this component can be described as below.

1) Preparing the dataset

To prepare the dataset, we have collected reviews given by different users on popular taxi applications in the world. Also the vague reviews were removed in order to increase the accuracy of the algorithm.

2) Training the dataset

For the process of training, the dataset collected is processed through an algorithm to eliminate the unbiased words and stopwords. Word stemming was practiced in this process in order to categorize the words with the same root. Finally a dictionary of words with relevant ratings is made from the dataset. In this process, several libraries connected with python was used; nltk was very useful especially in collecting stop words and word stemming.

3) Analysing the ratings

When analysing the rating of the given review, the given sentiment is subjected to stopword, unbiased word removal and also word stemming is applied on it. Next the words in the sentence are segmented and ran through an algorithm to calculate the probability of those words falling into the rating of 1,2,3,4,5 with relevant to the trained dataset; with the concept of Naive bayes algorithm. Finally the rating category with highest calculated probability is identified as the rating of the provided sentiment.

Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 2.1.2 - Concept of Naive Bayes algorithm used [48]

At the completion of this process to improve the accuracy of the rating generated, we have implemented a mechanism to get the median of the rating provided by user and the rating calculated from the naive bayes algorithm.

For the implementation of web api through python, we used Flask library(v1.0.3). So that both the document analysis and rating calculation methods were exported as REST endpoints.

Fare calculation plays a vital role in distributing fare among the passengers. There were two types of fare calculations. First one was estimated fare calculation which is predicted before starting the ride, and the other one is the actual fare calculated using the statistics collected, and it will be notified to passengers via a push-up notification at the end of the riding session. Fuel consumption varies from vehicle to vehicle. So, the project surveyed to collect information about the cars which were travelling within the Colombo area. Statistics like manufacturer year, engine capacity were collected during the survey.

Fuel type, engine power (kW), transmission type, travelled mileage and average fuel consumption. Afterwards a multiple linear regression model was created to predict the fuel consumption according to the vehicle condition. Number of kilometers get according to the starting point and destination of the passenger using Google distance matrix API.

It is identified that it is reasonable to use the following properties for an estimated fare calculation: (i) the price of a liter of fuel in Sri Lanka market(c); (ii) distance(d); (iii) number of passenger/s joined with the ride(p); (iv) fuel consumption(n) in km/l. Equation (1) is derived as a result of the study to calculate the estimated fare of a user.

$$Estimated\ fare\ Calculation = \frac{\left(\frac{c}{n}\right) * d}{p}$$

Equation 1 Estimated fare Calculation

Need to share some important notice both drivers and passengers. So, it was decided to push notification better than text messaging or call mechanism. After in-depth studied, we choose firebase cloud messaging (FCM) Service. Essential to follow several steps to achieve this target.

I) User authenticates to a firebase app

After the user authenticates returned in the information of users as a callback function to the client device.

```
W/BiChannelGoogleApi: [FirebaseAuth: ] getGoogleApiForMethod() returned Gms: com.google.firebase.auth.api.internal.zzal@20ff8cc
D/FirebaseAuth: Notifying id token listeners about user ( F14Z9n4DubRpPEi6qgYaivDLqGj2 ).
    Notifying auth state listeners about user ( F14Z9n4DubRpPEi6qgYaivDLqGj2 ).
D/FirebaseApp: Notifying auth state listeners.
    Notified 1 auth state listeners.
```

Figure 2.1.3 - information of users as a callback function to the client device

The returned callback contains the user id (UID), which is the unique id across all providers, and it never changes for a specific authenticated user.

II) Generate device registration token

Device registration token is essential to send push notifications as well as that token should be saved on firebase real-time database if user authentication success because of that data save under the user UID which generated in the authentication process.

That generated token should save under the userId in MySQL database because we need to send notification from device to another device (From passenger to driver / From driver to passenger).

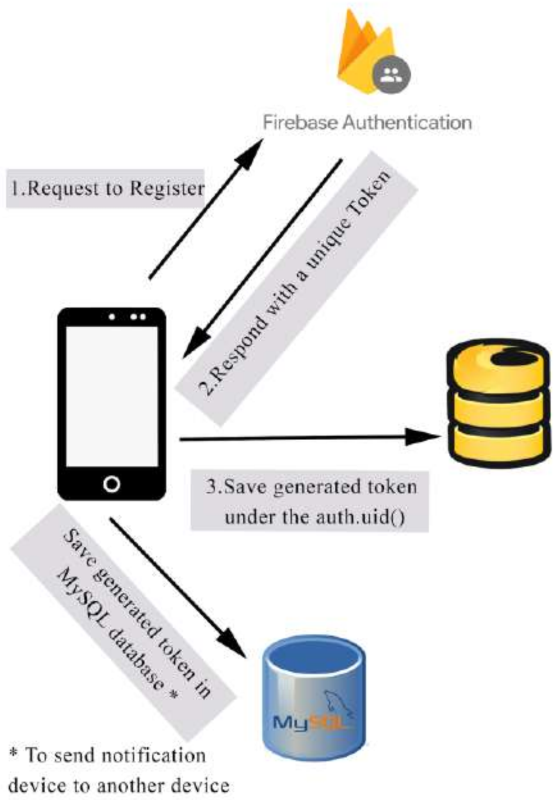


Figure 2.1.4 -Client app registration process [49]

Figure 2.1.5 - Device token save along with user authenticate email

Initially, developed REST API using NodeJS to send notification alone with device token. Unable to send notification without the same network, due to service running on the localhost. To break that limitation, we deployed our REST API in firebase, after that firebase notification service works on ideally without any disturbances.

The trip requested process as follow,

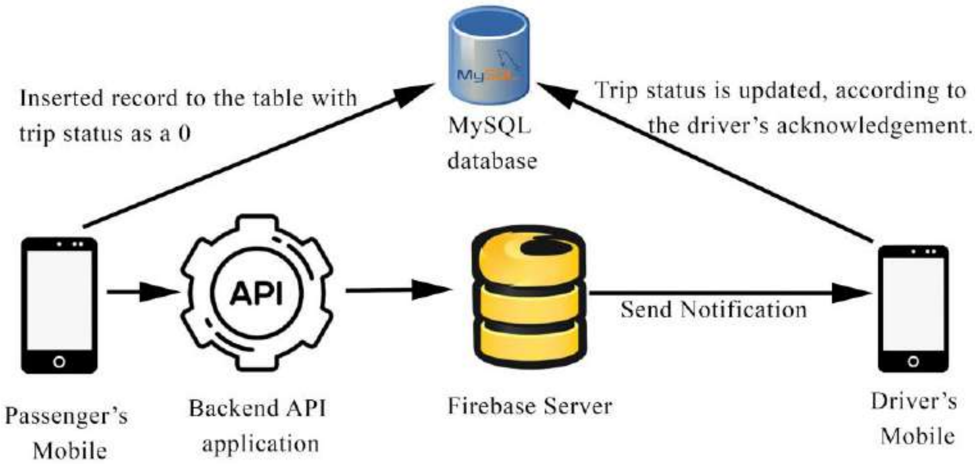


Figure 2.1.6 - Trip requested process

The static pricing methods assign a fixed price for all the users who are on the platform, and it's obviously not fair depending on the distance and time they travel. Therefore, introduced new fare equation to calculate the fare of the ride. Equation (2) calculates for each segment created throughout the ride. If new passenger joins the trip or passenger end up the trip, new segment will be created. Segment denotes i_n ($n \in Z+$), segment i_1 start fuel consumption is f_{i-1} , as well as segment i_1 end fuel consumption is f_i .

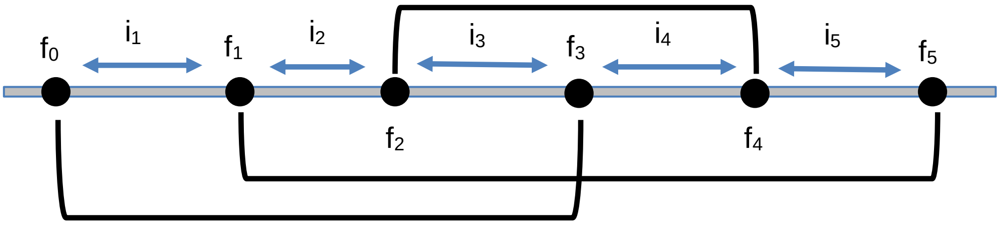


Figure 2.1.7 - Graphical representation how segment divide

Inputs of the equation are the cost for fuel consumption within the segment (f), the number of current passengers (p), and the current price of the fuel in Sri Lanka (price).

$$C\ i\ (\text{Total fare for the i th segment}) = \frac{\text{f}*\text{price}}{\sum \text{p}}$$

Equation 2 Total fare for i^{th} segment

With the fare of one segment is calculated, we add all the segments together to find out the total fare of the ride using (3).

$$\text{Total Fare of the Ride} = \sum_{i=Start\ point}^{end\ point} (\text{C i})$$

Equation 3 Total fare of the ride

Real-time fuel consumption gets from the OBD II Scanner that device based on ELM327 microcontroller, which has a low power CMOS design and RS232 serial interface. Based on the terminal type, it includes an RS232 to Bluetooth, USB or WIFI converter module inside the adaptor. In the mobile phone applicable only Bluetooth and Wi-Fi among them Bluetooth is better because since it was less expensive and typically has a lower power consumption[50] .

The overall overview of the fare calculation component that can able to get clear idea from the following Use case diagram.

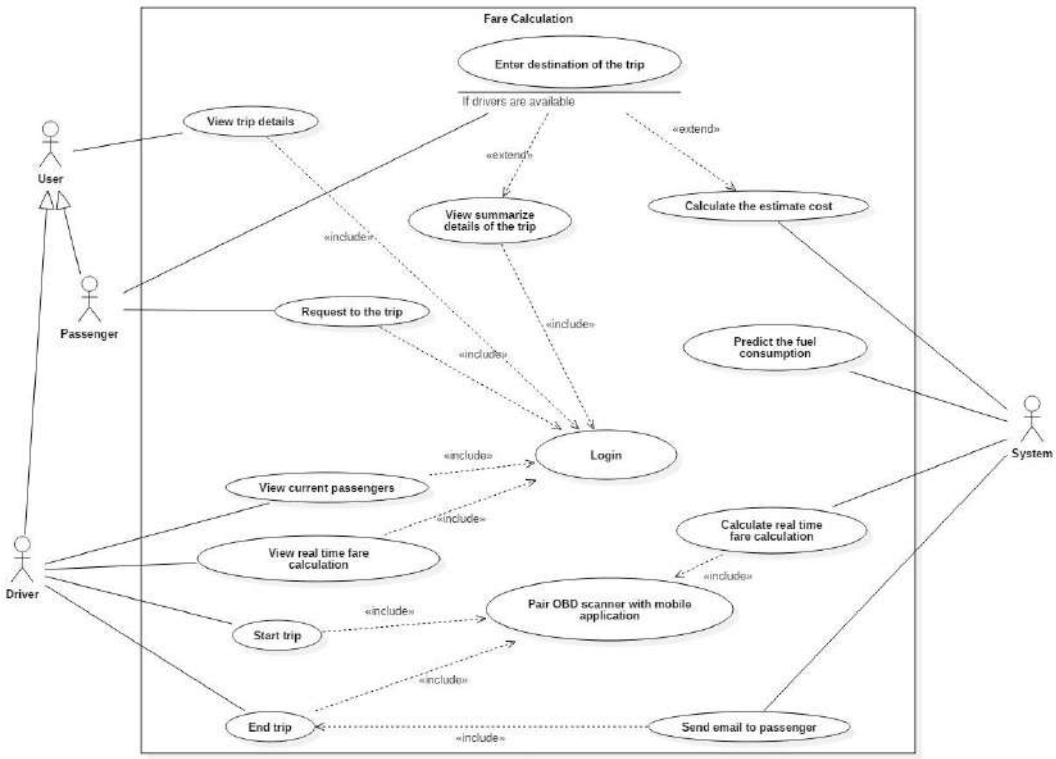


Figure 2.1.8 - Use Case of FC

2.1.5 Optimum Path Recognition

In optimum path recognition, we use both Dijkstra's Algorithm and the 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'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. Also, AWS is used to store the images separately to increase efficiency.

The following use case diagram will show the overall functionality of the optimum path recognition.

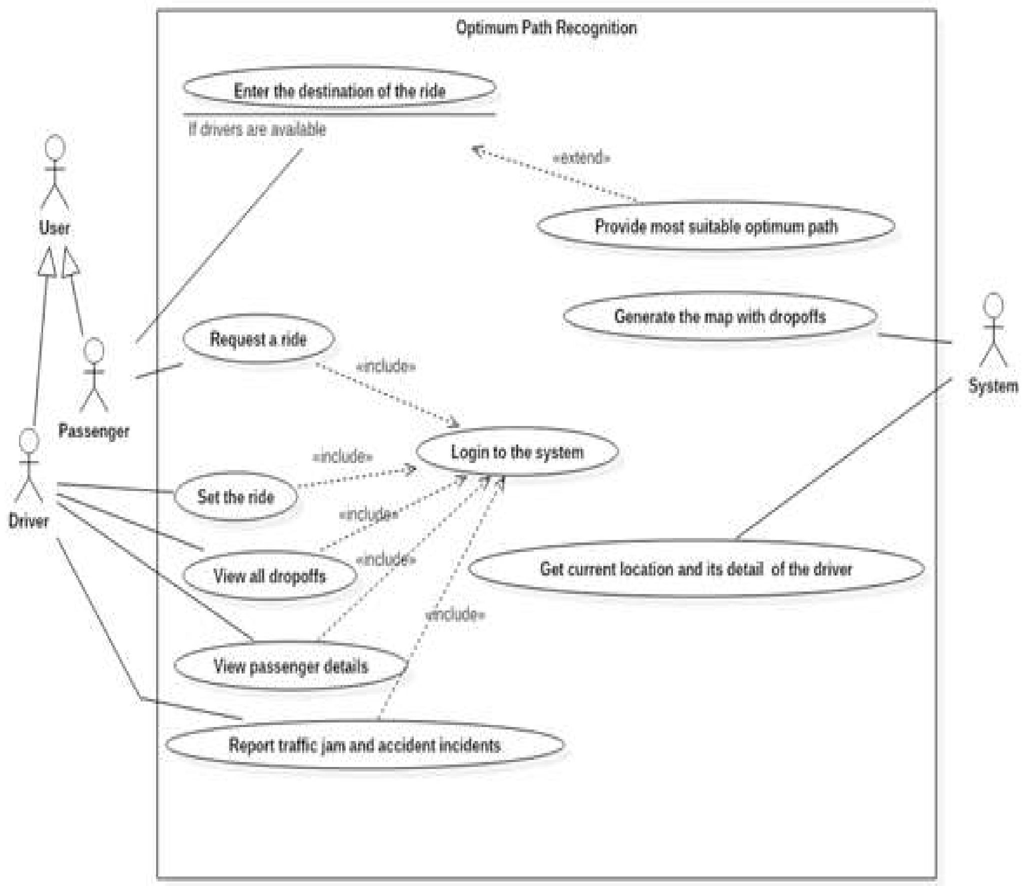


Figure 2.1.9 - Use Case of OPR

2.2 Commercialization Aspects of the Product

In the business point of view, it is essential to market the product among the people. For that, we should have a comprehensive plan addressing every part of the commercialization. The speciality of our product from other ride-sharing applications is that we only focused on the professionals. Hence, the passengers are fully satisfied with the drivers suggested to them with our ride-matching mechanism. Currently, the +Go application covers only the Colombo area, and we'll be planning to spread it across the country in a few months of time. Our app is absolutely free to download and use. We would charge only 10% of the total fare spend by a particular user as our revenue. Currently, all the services in the application are free, and we'll be introducing new value-added services in the future for a small monthly subscription fee. Also, we planned to introduce dedicated parking slots for +Go users with the help of the government. Meanwhile, we planned introduced new loyalty schemes and promotional codes for the top-rated users to encourage them to use our application more. Currently, only credit/debit card payment method is integrated, and in the future, we planned to integrate other payment gateways like internet banking, PayPal, and EZ cash for the easiness of the user.

Before releasing it to the market officially, we planned to release a beta version which can be given to several professionals and test for the applicability. After successfully testing the beta version, we planned to release to the market officially. Currently, our application can only be used by Android users, and if the iOS market urges for the product, we'll be developing for iOS as well. Our product will be mainly promoted via social media platforms as well as advertisements. Current commercialization plan is given below where after the official release, depending on the reaction of the users, our concept may be merged with an existing company or will be starting a startup business with several other members and licensed it for future work.

2.3 Testing and Implementation

2.3.1 Implementation

2.3.1.1 Hardware Interfaces

As the solution is a mobile application, this will be using a small 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.

2.3.1.2 Software Interfaces

The primary software interfaces used in the application are,

- For the application development: Android
- 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

2.3.1.3 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

All the backend servers and database are deployed in Amazon AWS and Google Cloud.

2.3.2 Testing

For any kind of application, testing plays a significant role in the success of the application. Quality testing procedures can reduce the number of bugs in the application and identifying them before the release is essential as well as cost-effective. Therefore, we used the V model in the testing phase, where each component after the completion is tested individually before the integration. It is more effective and easy to fix bugs in the unit testing level rather than solving them after the integration. After each module is tested thoroughly, they were integrated and tested once again as a whole. After finishing all the components, the full system test was carried out as an alpha testing. As this is a mobile application, both the backend and front end need to be tested thoroughly. The backend was tested with all the possible cases by inputting to the ride-matching algorithm as given below.

A	B	C	D	E	F	G	H	I	J	K
Smoking	Music_Lover	Motion_Sickness	Like_Quietness			Gender_Preference		Checked		
Yes	Yes	Yes	Yes		Male	Female	No	Tested	Tested	Tested
Yes	Yes	Yes	No		Male	Female	No	Tested	Tested	Tested
Yes	Yes	No	Yes		Male	Female	No	Tested	Tested	Tested
Yes	Yes	No	No		Male	Female	No	Tested	Tested	Tested
Yes	No	Yes	Yes		Male	Female	No	Tested	Tested	Tested
Yes	No	Yes	No		Male	Female	No	Tested	Tested	Tested
Yes	No	No	Yes		Male	Female	No	Tested	Tested	Tested
Yes	No	No	No		Male	Female	No	Tested	Tested	Tested
No	Yes	Yes	Yes		Male	Female	No	Tested	Tested	Tested
No	Yes	Yes	No		Male	Female	No	Tested	Tested	Tested
No	Yes	No	Yes		Male	Female	No	Tested	Tested	Tested
No	Yes	No	No		Male	Female	No	Tested	Tested	Tested
No	No	Yes	Yes		Male	Female	No	Tested	Tested	Tested
No	No	Yes	No		Male	Female	No	Tested	Tested	Tested
No	No	No	Yes		Male	Female	No	Tested	Tested	Tested
No	No	No	No		Male	Female	No	Tested	Tested	Tested

Figure 2.3.2.1 Backend Testing Plan

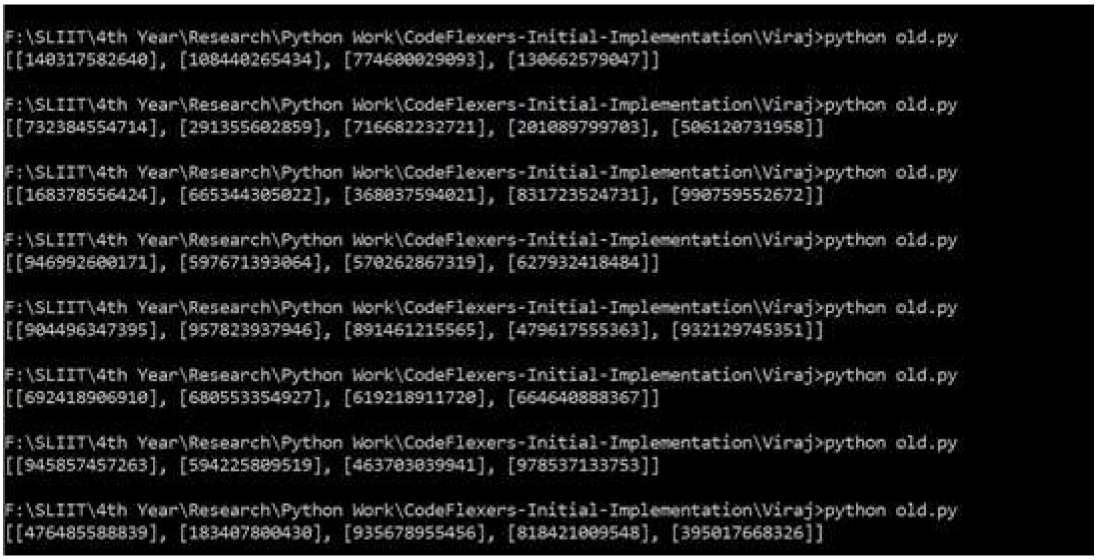


Figure 2.3.2.2 Backend Testing Results

Each and every API is tested individually using Postman; an HTTP test client. All the GET, POST, PUT and DELETE requests are tested with sample data for the verification. Below figures show some of the tested scenarios in Postman Client.

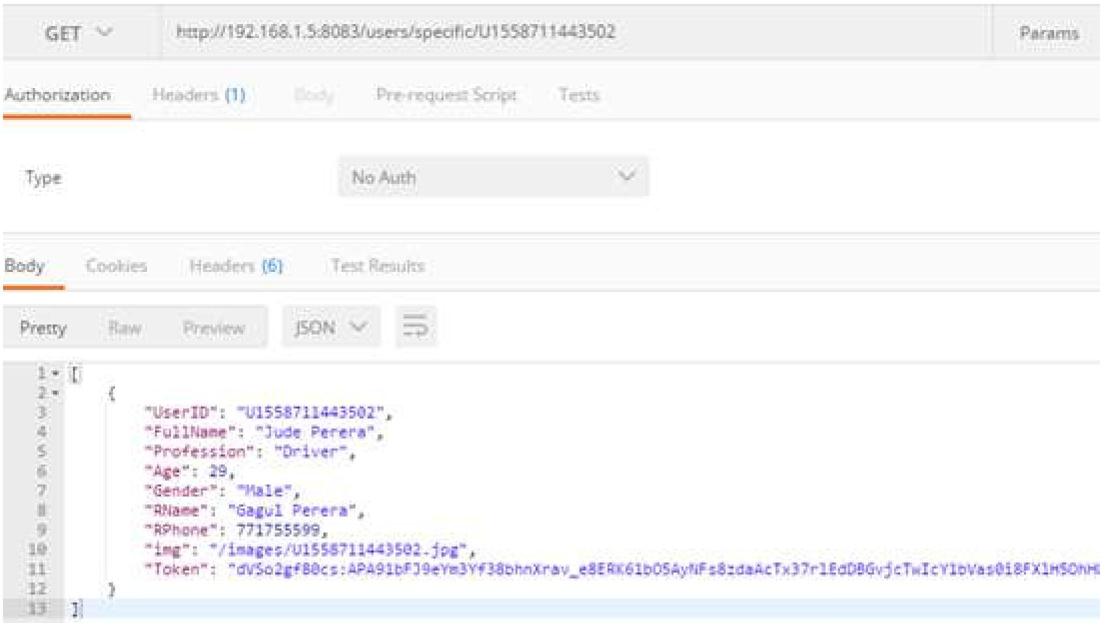


Figure 2.3.2.3 GET request testing

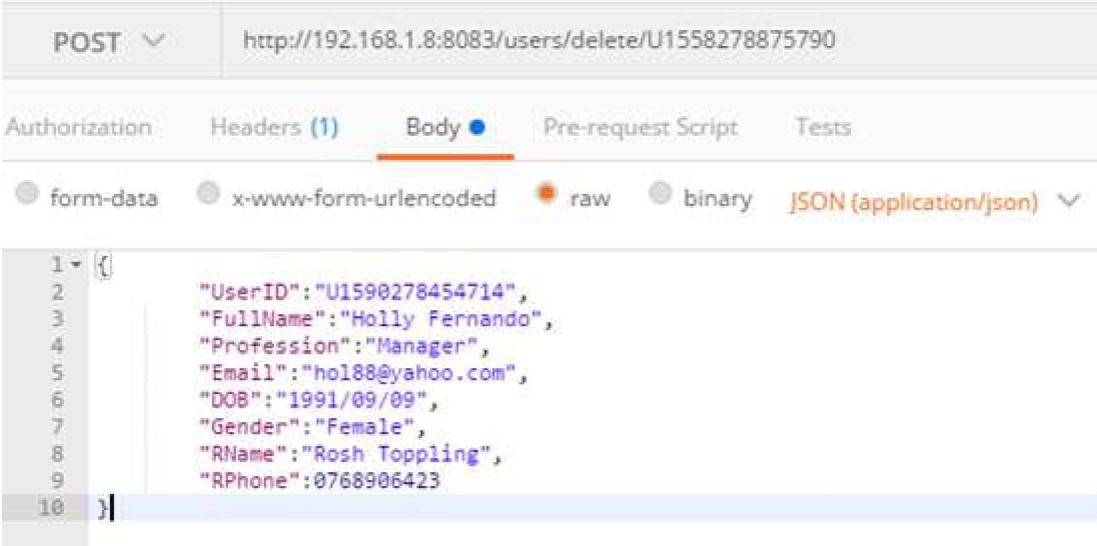


Figure 2.3.2.4 POST request testing

Interfaces were tested using a comprehensively prepared set of test cases. In here, expected output was validated against the actual output to check whether the application works as desired. Frontend testing was done through the manual process and checked against any unexpected errors and bugs.

Table 2.3.2.1 Test Case 01

Test Case ID	TC01
Test Case Description	Validate User Credentials
Pre-Condition	Need to be a registered user
Test Procedure	• Provide the username • Provide the password • Click on the Login button
Test Input	• Username: Jude • Password: jude@123
Expected Output	User should be validated and direct in the user profile
Actual Output	User logged into the system as expected

Table 2.3.2.2 Test Case 02

Test Case ID	TC02
Test Case Description	OTP Verification
Pre-Condition	User should be a new user, and the mobile phone number is not used to register for an account in our application
Test Procedure	• Provide the mobile phone number • Click Send button • Enter the Verification Code • Click Verify button
Test Input	• Mobile Number: 0766823421 • Verification Code: 7293
Expected Output	The system should verify the mobile phone number
Actual Output	Error message with the inability to verify the mobile number

Table 2.3.2.3 Test Case 03

Test Case ID	TC03
Test Case Description	Provide the basic Information
Pre-Condition	User should be a registered user
Test Procedure	• Provide all the information required
Test Input	• All the information need to be submitted
Expected Output	Redirect to the preferences UI
Actual Output	Successfully submitted the details and redirected to the preferences

Table 2.3.2.4 Test Case 04

Test Case ID	TC04
Test Case Description	Provide Credit Card Information
Pre-Condition	User should be a registered user
Test Procedure	• Provide the Credit Card Number • Provide CVC • Provide Expiration Date • Click Submit
Test Input	• Credit Card Number:451234127810xxxx • CVC:8xx • Expiration Date: 03/23
Expected Output	Credit Card Validated successfully message
Actual Output	Success message

Table 2.3.2.5 Test Case 05

Test Case ID	TC05
Test Case Description	Requesting a ride and no drivers available
Pre-Condition	User should be a passenger and wants to request a ride
Test Procedure	• Provide the destination • Click Search Drivers • Click on the relevant driver
Test Input	• Destination: Colombo
Expected Output	Set of drivers who travelled to Colombo
Actual Output	No available drivers at that moment

Table 2.3.2.6 Test Case 06

Test Case ID	TC06
Test Case Description	Requesting a ride and drivers available
Pre-Condition	User should be a passenger and wants to request a ride
Test Procedure	• Provide the destination • Click Search Drivers • Click on the relevant driver
Test Input	• Destination: Colombo
Expected Output	Set of drivers who travelled to Colombo
Actual Output	List of drivers who travelled to Colombo

Table 2.3.2.7 Test Case 07

Test Case ID	TC07
Test Case Description	Update personal details
Pre-Condition	User should be a registered user and have a completed profile
Test Procedure	• Click on the side pane • Select Your profile • Change the details • Click the Update button
Test Input	• Provide information for all the fields
Expected Output	Updated successfully toast message
Actual Output	Updated successfully toast message and redirect to the map activity

For the testing of the accuracy of NIC and license, we collected nearly 100 different images from people and tested for the accuracy of text extraction. We also did come adjustments of brightness and contrast in images and tested for their accuracy as well.

For the testing purposes of accuracy of sentiments, we prepared another dataset with some of the sentiments used in the training dataset and also some other new reviews taken from popular applications. Then we tested the dataset by comparing the rating generated from the algorithm to the actual ratings given by the users.

For the testing purposes of functionalities of android application, we used several test cases. A few of them are as follows.

Table 2.3.2.8 Test Case 08

Test Case ID	TC08
Test Case Description	Non electronic NIC card validation
Pre-Condition	User has been successful in previous steps User own a valid NIC card
Test Procedure	• Choose to enter non electronic • Select Upload image button • Select image of NIC from gallery or capture it from the camera • Click button with upload icon • Click Verify button
Test Input	• Valid image of non-electronic NIC card
Expected Output	NIC number is extracted and button to proceed appear
Actual Output	NIC number correctly identified in the textbox and button to proceed appear

Table 2.3.2.9 Test Case 09

Test Case ID	TC09
Test Case Description	Licence Card Verification
Pre-Condition	User has been successful in NIC verification
Test Procedure	• Select the upload image button • Upload image from gallery or capture image • Click button with upload icon • Click verify button
Test Input	• A clear image of License card
Expected Output	Application should extract the NIC number and expiry date from the image
Actual Output	NIC number and expiry date was displayed on a text box and a button appeared to proceed.

Table 2.3.2.10 Test Case 10

Test Case ID	TC10
Test Case Description	Rating given as a passenger
Pre-Condition	The user has completed the trip session as a passenger
Test Procedure	• Provide a rating below 5 • Select vehicle out of the three

	categories ● Click other ● Write a review in the text box ● Click done
Test Input	● Input rating: 4 ● Review text: “The vehicle was not comfortable as expected”
Expected Output	In the category tab with “Vehicle” appears an icon of a yellow star.
Actual Output	Yellow star appeared after the test procedure was finished.

Table 2.3.2.11 Test Case 11

Test Case ID	TC11
Test Case Description	Rating as a driver
Pre-Condition	The user has completed the trip session as a driver
Test Procedure	● Provide a rating below 5 ● Select a keyword
Test Input	● Input Rating : 3 ● Keyword : Professionalism
Expected Output	Close the mobile interface of rating.
Actual Output	The rating mobile interface was closed

Interfaces relevant to OPR were tested using several test cases as shown below.

Table 2.3.2.12 Test Case 12

Test Case ID	TC12
Test Case Description	Inserting the trip ride information
Pre-Condition	User should be a registered user(driver)
Test Procedure	• Enter the source • Enter the destination • Enter the start date • Enter the start time • Enter the waiting time • Click Submit button
Test Input	• Source : Malabe • Destination : Kollupitiya • Start Date : 09/08/2019 • Start Time : 8.30am • Waiting Time : 5 minutes

Expected Output	Displaying a message after submitting the data.
Actual Output	Submitted the details successfully with a success message

Table 2.3.2.13 Test Case 13

Test Case ID	TC13
Test Case Description	Request a ride
Pre-Condition	User should be a registered user(passenger)
Test Procedure	• Enter the source • Enter the destination • Click Accept Route button
Test Input	• Source : Malabe • Destination : Kollupitiya
Expected Output	Display optimum route on the map

Actual Output	Show optimum path on map and display the success message after submitting the details successfully.

Table 2.3.2.14 Test Case 14

Test Case ID	TC14
Test Case Description	Insert traffic jam information
Pre-Condition	User should be a registered user(driver)
Test Procedure	• Enter the traffic location • Enter the situation type • Upload the image of the incident by clicking upload button • Click Report button
Test Input	• Location :Thalahena • Situation : Heavy • Thalahen_traffic.png

Expected Output	Displaying a message after submitting the data.
Actual Output	Submitted the details successfully with a success message

Table 2.3.2.15 Test Case 15

Test Case ID	TC15
Test Case Description	Insert accident information
Pre-Condition	User should be a registered user(driver)
Test Procedure	• Enter the accident location • Enter the situation type • Upload the image of the incident by clicking upload button • Click Report button

Test Input	• Location :Battaramulla • Situation : Major • Battaramulla _accident.png
Expected Output	Displaying a message after submitting the data.
Actual Output	Submitted the details successfully with a success message

Interfaces relevant to FC were tested using several test cases as shown below.

Table 2.3.2.16 Test Case 16

Test Case ID	TC16
Test Case Description	To calculate estimate fare calculation
Pre-Condition	Need to be a registered user
Test Procedure	• Provide starting point and destination • Select the driver from suggested driver list

Test Input	Start point: Kelaniya Destination: Kadawatha Current passenger: 1 Price of a liter of fuel: Rs.136.00 Vehicle Information Manufacture year :2017 Registered Year :2018 no of cylinders :3 Fuel type: petrol – Hybrid Engine capacity: 658 cc Engine Power: 38 kW Mileage 7233
Expected Output	Rs.30.28
Actual Output	Rs.30.28

Table 2.3.2.17 Test Case 17

Test Case ID	TC17
Test Case Description	Request to the ride
Pre-Condition	Need to be a registered user
Test Procedure	• Provide the destination • Click search driver • Click on the relevant driver • Click on the “Join a Ride” button

Test Input	• Driver's device token: X • Passenger's device token : Y • Passenger Name: Jude • passengerId: U1558711443502 • driverId: U1558711443513 • tripId: O1558711443513 • Start point: Kelaniya • Destination: Kadawatha
Expected Output	Display Message as "Your request message has been sent"
Actual Output	Success Message

Table 2.3.2.18 Test Case 03

Test Case ID	TC18
Test Case Description	Accept of the trip
Pre-Condition	• Need to be a registered user • Passenger need to request to the trip • OBD II device pair with driver's mobile

Test Procedure	• Driver provide the response for the request
Test Input	• Passenger's device token : Y
Expected Output	• Driver's mobile redirect to the driver dashboard • Passenger received the notification as "Driver will arrive soon. Please wait until knocks at your place"
Actual Output	Received the notification the system as expected

3. RESULTS AND DISCUSSION

3.1 Results

3.1.1 Test results

3.1.1.1 Matching the most suitable driver

The lack of flexibility provided by the existing applications in suggesting suitable drivers to the passengers quoted as one of the primary reasons for the people to hesitate in using ride-sharing platforms[51]. Our dynamic approach has performed particularly well in giving out more optimized results to the expected user. For the algorithm, the comprehensively prepared dataset was used both for development and testing purposes. Below figures show a part of the dataset used for the development.

1	UID	Profession	Rating	Age	Profession_Category	Language_Spoken	Gender_Preference	Smoking	Music_Lover	Motion_Sickness	Like_Quietness
2	U8450302883366	Clerk	5	37	25	Sinhala	Male	Yes	Yes	Yes	Yes
3	U788539641894	Driver	2.4	44	20	Sinhala	Male	Yes	Yes	Yes	Yes
4	U2879353312224	Body Guard	4.3	32	17	Sinhala	Male	Yes	Yes	Yes	Yes
5	U3650469397532	Security Officer	2.1	31	15	Sinhala	Male	Yes	Yes	Yes	Yes
6	U1558711443588	Clerk	2.6	31	20	Sinhala	Female	Yes	Yes	Yes	Yes
7	U1558711443504	Body Guard	3.9	33	17	Sinhala	Female	Yes	Yes	Yes	Yes
8	U1558711443505	Security Officer	4.9	35	15	Sinhala	Female	Yes	Yes	Yes	Yes
9	U1558711443503	Clerk	3.8	39	25	Sinhala	Female	Yes	Yes	Yes	Yes
10	U1558711443502	Driver	5	32	25	Sinhala	No	Yes	Yes	Yes	Yes

Figure 3.1.1.1 Dataset Collected

Ride matching algorithm is an important finding in suggesting the most suitable driver list with recorded accuracy of 98%. Despite the accuracy, the number of parameters used to determine the suitable drivers directly proportional to the accuracy of the algorithm produced.

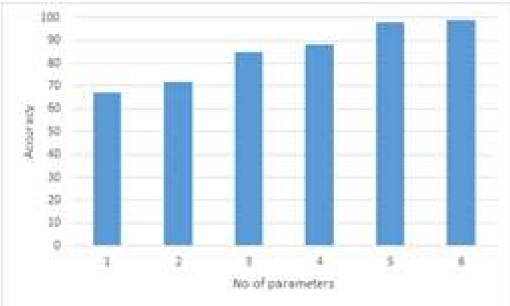
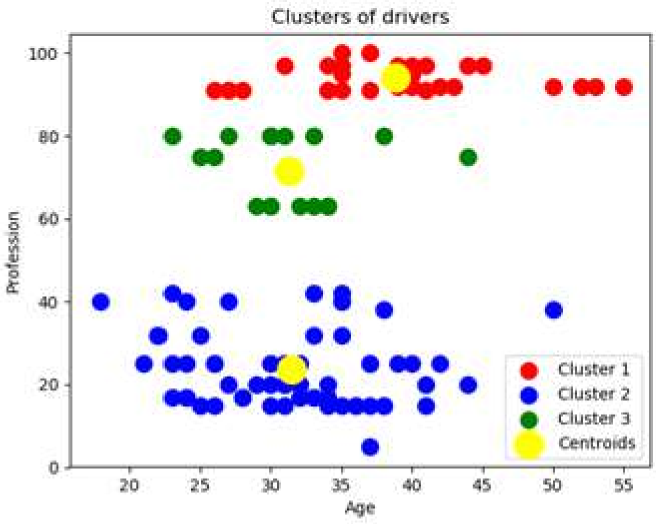


Figure 3.1.1.2 Accuracy vs Parameter count

In the above figure, it is clear that, when the input parameters are high, the accuracy of the results produced is too get increases.

A similar set of results were obtained by Ghoseiri, Haghani, and Hamedi (2011) matching routes based on the age, smoking condition, gender and occupancy preference. However, the accuracy of the results obtained from the study was not optimized like our study and Jung, Jayakrishnan, and Park (2013) used the same algorithm and provided a solution for the problem associated with optimization. However, in line with the ideas of C. C. Tao and Chen (2008), it can be concluded that this ride-matching algorithm can be more optimized using greedy heuristics. One concern about the findings was that when the number of parameters increases the complexity of the algorithm gets increases. This causes the algorithm to process slower than usual processing speed (0.86s). Because of this potential limitation, we apply smart data structures rather than moving into sophisticated optimization approaches.



Aligning with fig. 4(X axis-Age and Y axis-Profession), the selected set of drivers are scattered into several clusters using our ride-matching algorithm. The particular cluster which belongs to the particular passenger is extracted out and displayed.

3.1.1.2 Document Validation

After the completion in implementation of the application and the algorithms, the need was to verify whether the application is suitable to be used. For that we needed to verify that the functions are working and the algorithms are giving the most accurate expected results.

To test whether the algorithm of text extraction works correctly, we proceeded with API testing. A sample result obtained by processing the image in [003] can be identified as below.

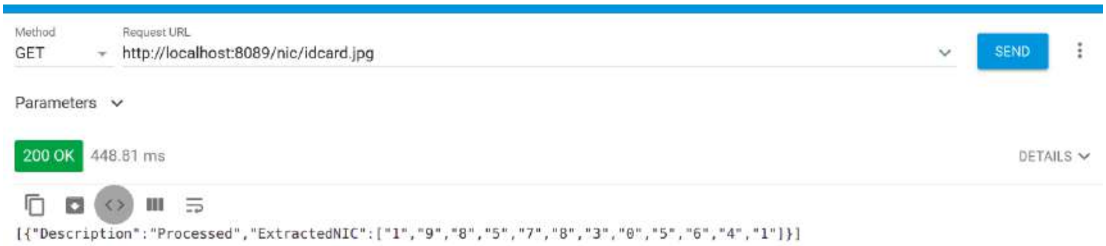


Figure 3.1.1.1 - Api testing to extract NIC number from non electronic NIC

In order to check whether the functionality works fine, we have written logs as follows.

```
 * Running on http://0.0.0.0:8089/ (Press CTRL+C to quit)
idcard
--- Start recognize text from NIC ---
--Started image processing for NIC using face recognition--
Found 1 faces!
Image verified
idcard
857835641
Old NIC contains 9 digits only --> Converted to 12 digit format
['1', '9', '8', '5', '7', '8', '3', '0', '5', '6', '4', '1']
```

Figure 3.1.1.2 - Logs to identify the procedure of text extraction

In all the procedures of text extraction from electronic NIC, non-electronic NIC and license, we did test the results generated as above. With this, we were able to identify that all that image extraction process in all types of documents were successful with the accuracy of more than 83.4%.

3.1.1.3 Rating analysis of sentiments

For the verification of accuracy in sentiment analysis algorithm, here too we did api testing.

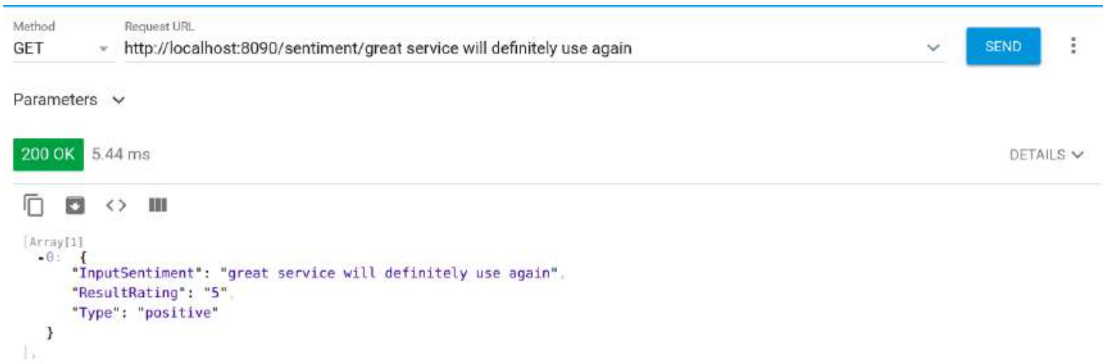


Figure 3.1.1.3 - Testing a positive sentiment

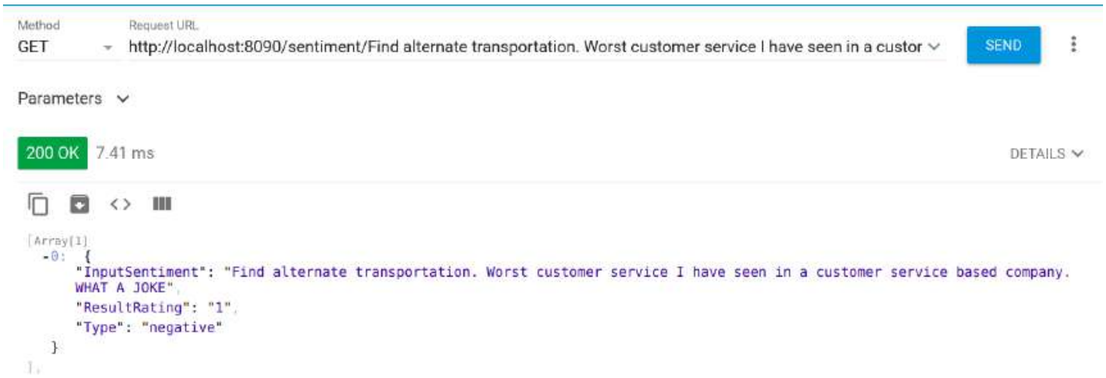


Figure 3.1.1.4 - Testing a negative sentiment

In order to verify that the naive bayes algorithm does work correctly in returning the most suitable rating value, we have written the logs as follows on each request.

```
 * Running on http://0.0.0.0:8090/ (Press CTRL+C to quit)
Find alternative transportation.Worst customer service I have seen in a customer servi
ce based company.What A JOKE
{'1': 1.1093358191643472e-12, '2': 5.369405677046016e-13, '3': 2.2283732129587245e-13,
 '4': 1.4578808367417437e-15, '5': 1.4515662002965292e-15}
negative
1
127.0.0.1 - - [10/Aug/2019 16:06:04] "GET /sentiment/Find%20alternative%20transportati
on.Worst%20customer%20service%20I%20have%20seen%20in%20a%20customer%20service%20based%
20company.What%20A%20JOKE???? HTTP/1.1" 200 -
```

Figure 3.1.1.5 - Logs to identify the sentiment analysis function

Further with the expectation of retrieving the accuracy of the algorithm, we prepared a dataset with 1000 randomly collected sentiments from popular applications in the world. We were able to identify the accuracy of the tested dataset as follows.

Figure 3.1.1.6 - Sample test results in a table format

```
-----Testing for the exact value-----
No. of correct matches of values   : 803
No. of incorrect matches of values : 197
                                     -----------------------
Accuracy of exact matches [%]      : 80.30000000000001

---Testing for the positive/negative sentiment identification---
No. of correct matches for sentiment classification   : 895
No. of incorrect matches for sentiment classification : 105
                                                        -----------
Accuracy of sentiment classification [%]              : 89.5

---Testing completed---
```

32	31	[(2.300421675404689e-25, '1'), (5.297710550566453e-22, '2'), (2.0334265749792314e-23, '3'), (9.686938931341956e-27, '4'), (2.0854303505461746e-29, '5')]	Negative	2
33	32	[(8.655961698911415e-53, '1'), (1.1214114724350308e-47, '2'), (1.764628300978523e-53, '3'), (2.2430749354496426e-58, '4'), (2.7833310607532477e-63, '5')]	Negative	2
34	33	[(1.346803896751994 4e-27, '1'), (9.277140728966769e-25, '2'), (1.5342126178492234e-25, '3'), (2.1887839907065695e-30, '4'), (1.7028850827575227e-31, '5')]	Negative	2
35	34	[(1.2861666625903779e-36, '1'), (3.388668209410422e-30, '2'), (5.529547691923579e-36, '3'), (4.2887115311757266e-41, '4'), (1.485857554526996e-47, '5')]	Negative	2
36	35	[(2.3355171219255377e-26, '1'), (5.782634626281131e-22, '2'), (1.5755354966838319e-25, '3'), (1.5204456895565209e-25, '4'), (8.243818106186289e-27, '5')]	Negative	2

Figure 3.1.1.7 - Test results of the sentiment analysis algorithm

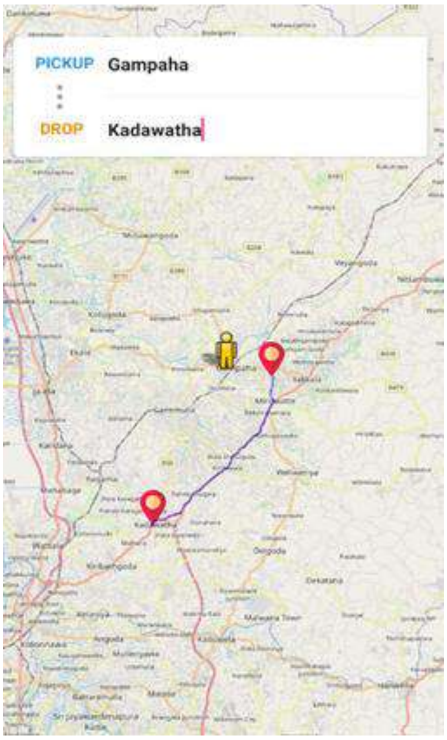
3.1.1.4 Optimum Path Analysis

Different methodologies were used for the verification and validation of our proposed system. Since the system allows multiple users to engage in work concurrently, it should be able to deliver the requested services without any delays. Therefore, the use of Google API for the identification of the optimum path is a virtuous solution. But manipulation of Google API is not the best when handling multiple requests at once. With the introduction of optimum path analysis using the Dijkstra algorithm and Travelling Salesman Problem, we can provide the optimum path considering traffic and distance as parameters. As a result of that, it would be helpful to handle the substantial amount of requests at once.

Source	Destination	Route Distance(km)	
		Proposed system	Google Map
Malabe	Kollupitiya	16.7	16.7
Dehiwala	Rajagiriya	9.37	9.4
Koswatta	Demataagoda	9.9	9.9
Thalahena	Gothatuwa	7.65	7.7

Figure 3.1.1.8 - Distance Comparison

Relevant to the figure 3.1.1.8, proposed system suggest more optimized path at the current moment using crowdsourcing platform and above mentioned algorithms. Therefore, it is not much contradicted from the Google API ,but it will display the less traffic specific path as the output.



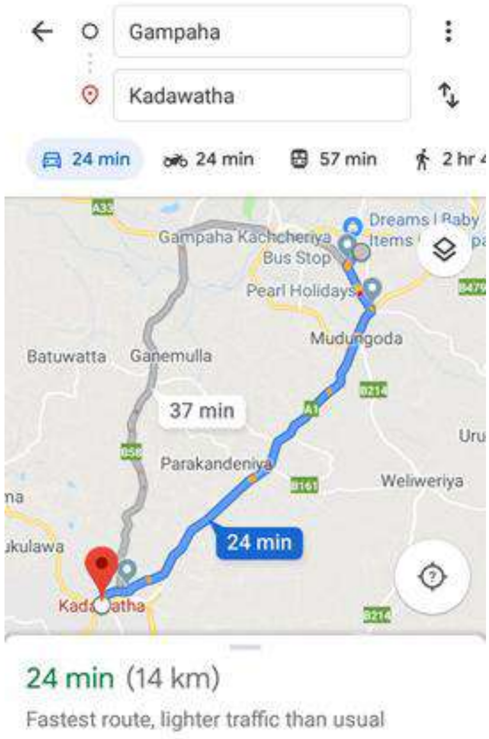


Figure 3.1.9: Optimum path given by the proposed System

Figure 3.1.10:Alternative routes given by Google Map

Preliminarily the proposed system was compared the Google maps results. Following table 3, will show the accuracy level by using the Dijkstra Algorithm and the Travelling Salesman Problem (TSP) algorithm.

	Results given by the Google Map	Proposed System Results(Plus Go)

Dijikstra's Algorithm	When users provide the starting location and ending location,Google map will suggest all the alternative routes on the map. Ex: Figure 3 shows all the alternative routes from Gampaha to Kadawatha as, ● via Colombo - Kandy Rd/Kandy Rd/A1/AH43 ● via Kadawatha - Ganemulla Rd However, optimum path generated by both Google Maps and proposed system is almost similar.	First user enters the source and destination and then optimum path will be displayed on the map by considering the distance factor, time factor, traffic jams or road conditions. Ex: Figure 2 shows the closest path from Gampaha to Kadawatha
Travelling Salesman Problem Algorithm	Relevant to the studies, Google map doesn't use TSP algorithm for the identification of optimum path.	When the user provides the starting location and ending locations, it will generate the closest path to travel by covering all the user specified locations.

These results emphasize that, this is a significant option for path identification and analysis with the combination of our crowdsourcing platform.

3.1.1.5 Fare Calculation

Collected dataset is divided into two parts in the multiple linear regression such as trainee dataset and test dataset. Trainee dataset reserved 80% of the whole dataset to train the linear model and rest of dataset apply for the trained model. After the model, compare the predicted value and the actual value. After the comparison model accuracy is 92.08%. Other accuracy measurements as follow,

Correlation coefficient	0.8232
Mean absolute error	0.726
Root mean squared error	1.198
Relative absolute error	50.1178 %
Root relative squared error	65.5605 %

Table 3.1.1.1 Other accuracy Measurements in Multiple linear regression model

3.1.2 System interfaces

Figure 3.1.1.1 Login Page

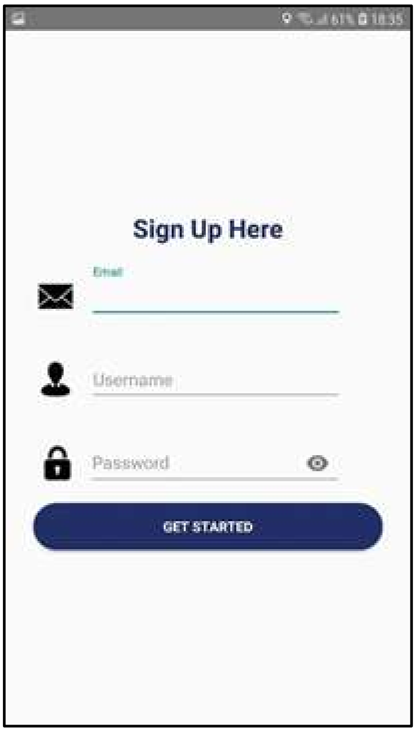


Figure 3.1.1.2 Signup Page

For any user who needs to use our application for the very first time need to go through the registration process. First, the user needs to create a new account to login to the system. From the signup button at the bottom of the login UI, the user can access the signup page. User can provide an email address, unique username and password and create a new account on the application using the UI in fig 3.1.1.2.

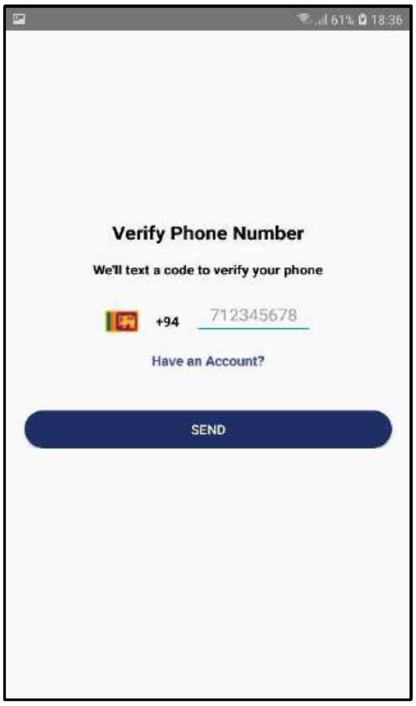


Figure 3.1.1.3 Phone Verification

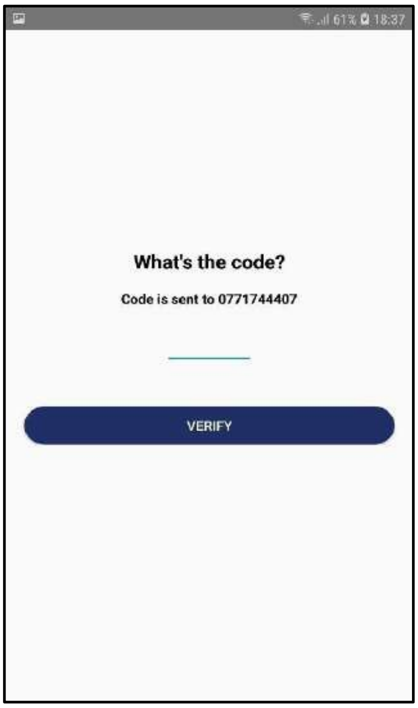


Figure 3.1.1.4 Code Verification

Figure 3.1.1.5 User Profile

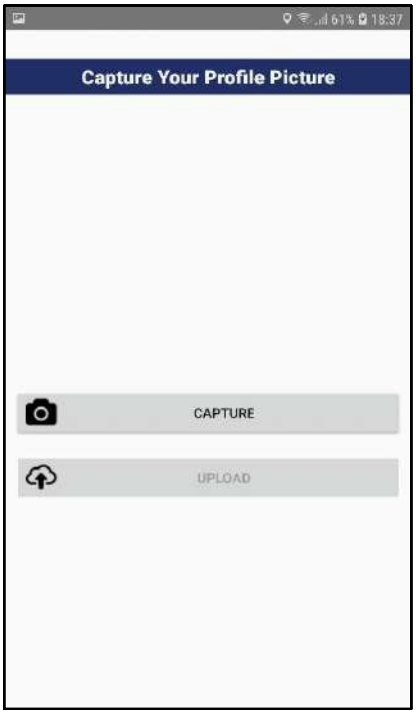


Figure 3.1.1.6 Profile Image

Figure 3.1.1.7 Preferences

Figure 3.1.1.8 Vehicle Details

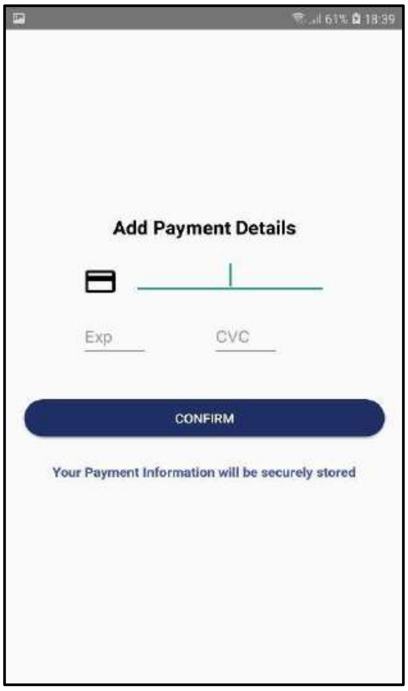


Figure 3.1.1.9 Payment Page

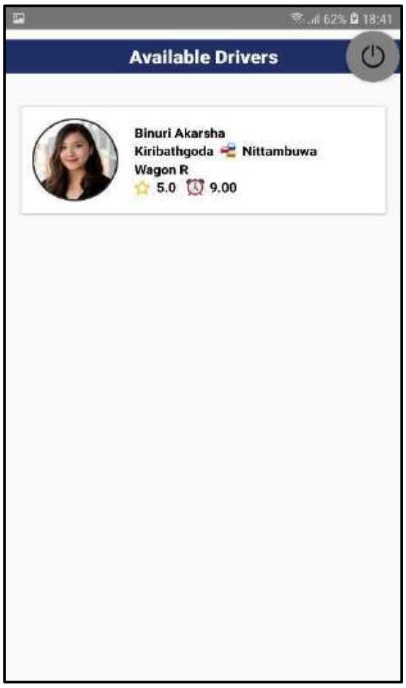


Figure 3.1.1.10 Driver List

After the account is created, the user needs to verify the mobile number as a step taken to prevent any fake profiles created on our platform. When the OTP message is received by the user, the verification code needs to be entered into the UI in fig 3.1.1.4 after the successful verification account is created successfully. Next, the user needs to provide the basic information, as shown in the fig 3.1.1.5. By using the UI in the fig 3.1.1.6, user can take a picture of himself/herself and upload it to the profile. After providing the personal details, preferences of the user is taken using the fig 3.1.1.7 followed by vehicle details in fig 3.1.1.8. From the payment page shown in fig 3.1.1.9, user can add credit card information and set it to get deducted after the ride. Thereby no manual intervention needed at the payment where at the end of the ride, the amount of the ride is automatically deducted from the card. When the passenger searches for a destination and looks for a driver, using our backend algorithm most suitable driver list will be suggested as shown in the fig 3.1.1.10. At the current moment, only one driver is available to the destination of that particular passenger. Hence, there can be many drivers available at a particular time interval or no drivers available at times.

We have designed the mobile interfaces related to DVPRM component. User Interfaces relevant to document validation process are as follows.

Figure 3.1.2.1 - NIC Type selection

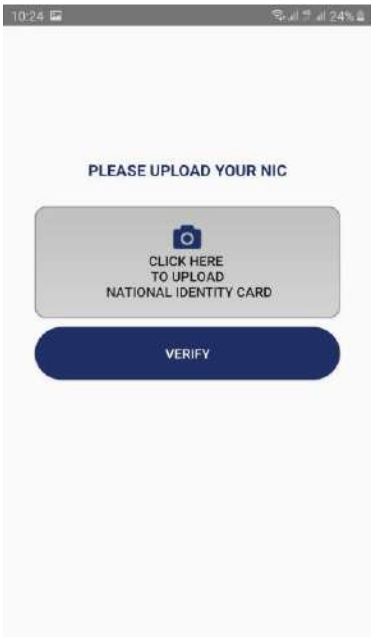


Figure 3.1.2.2 - NIC upload

Figure 3.1.2.3 - Capture or upload

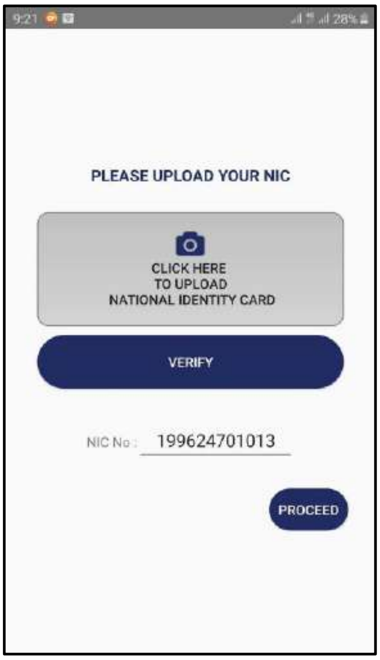


Figure 3.1.2.4 - NIC extraction

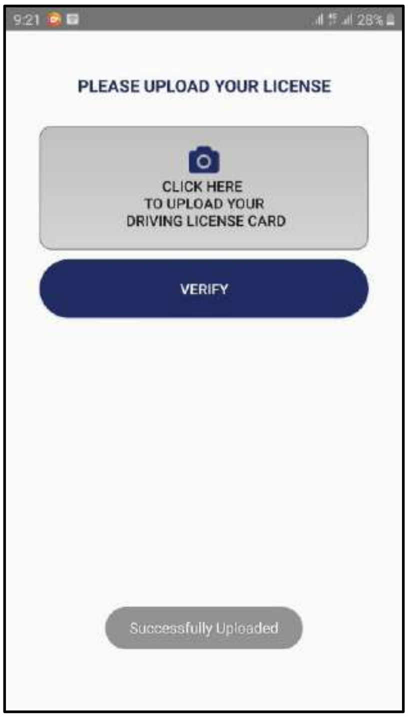


Figure 3.1.2.5 - Successful license upload

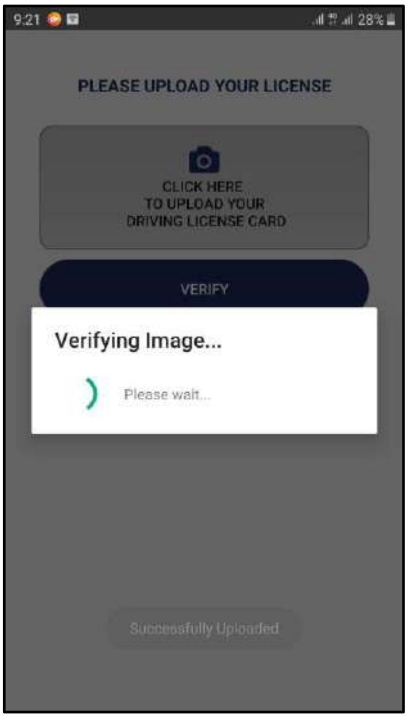


Figure 3.1.2.6 - License verification

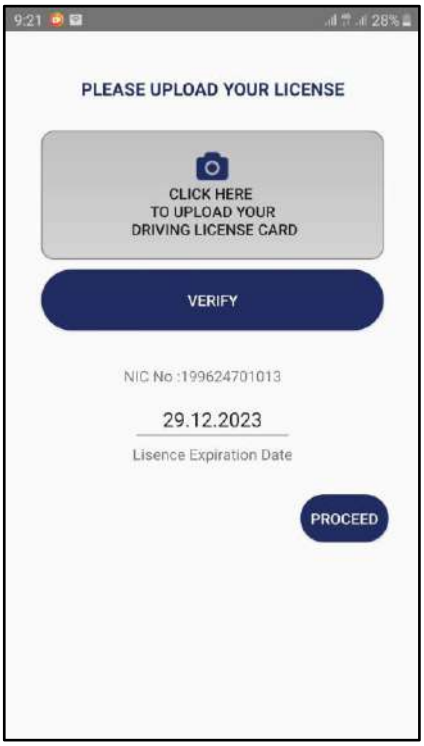


Figure 3.1.2.7 - License Extraction

User Interfaces relevant to the driver in profile rating maintenance.

Figure 3.1.2.8 - Rate as a driver

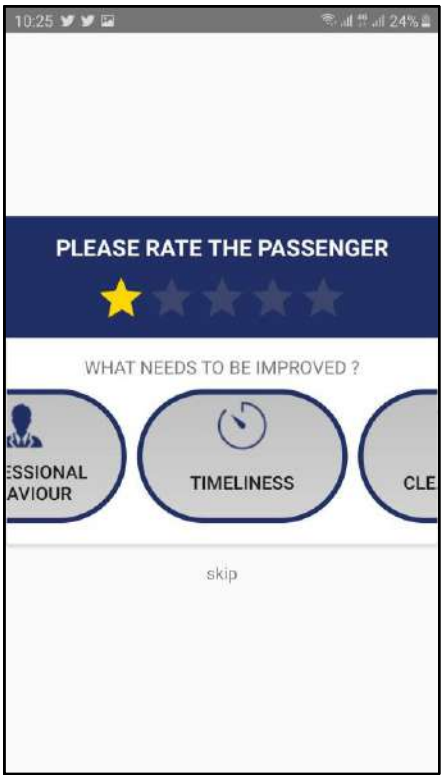


Figure 3.1.2.9 - Keyword selection

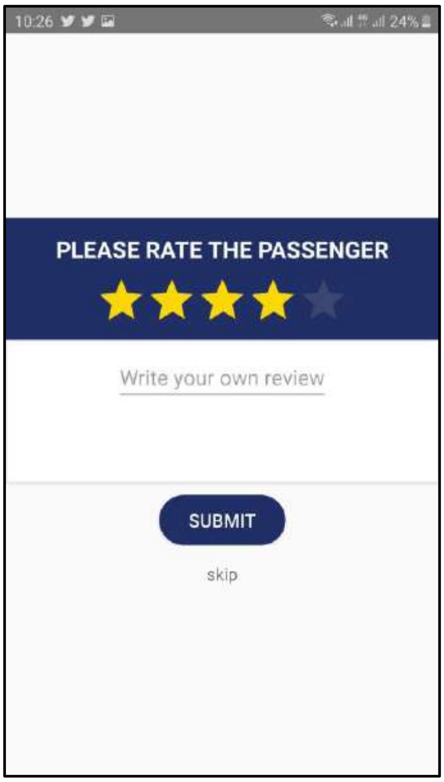


Figure 3.1.2.10 - Review writing

User Interfaces relevant to the passenger in profile rating maintenance.

Figure 3.1.2.11 - Rate as a passenger

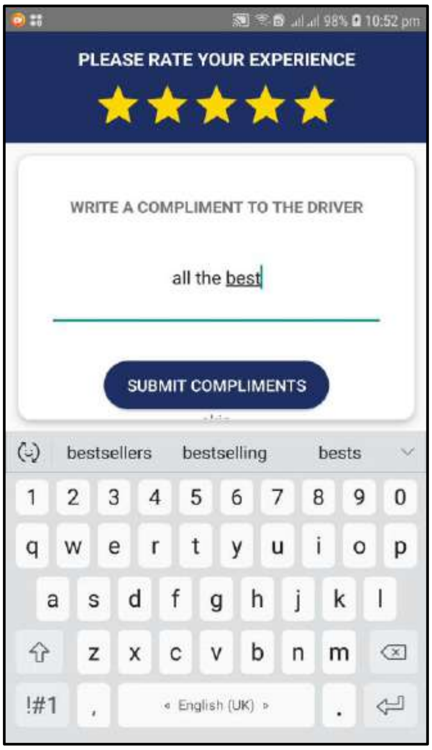


Figure 3.1.2.12 - All star rating

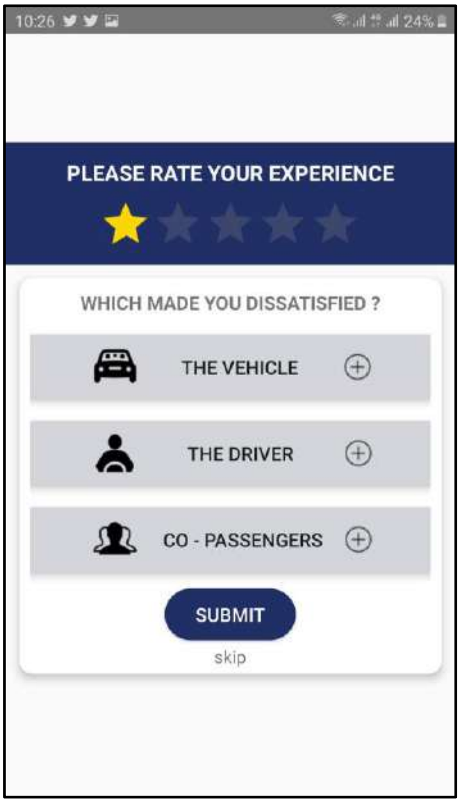


Figure 3.1.2.13 - Low rating

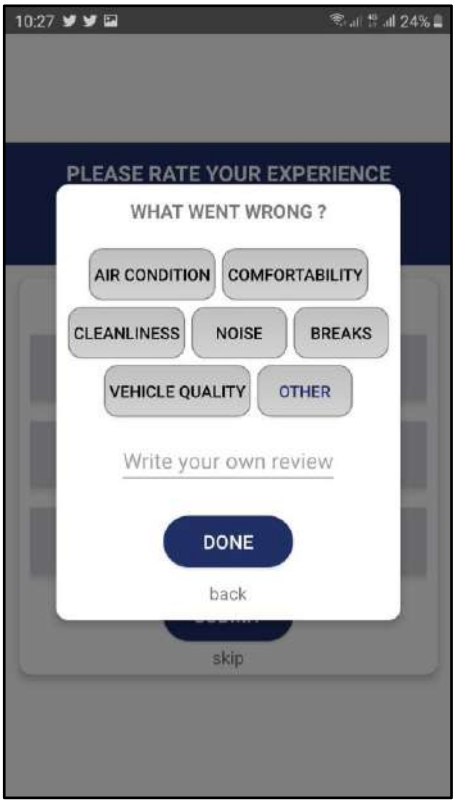


Figure 3.1.2.14 - Keywords to rate vehicle

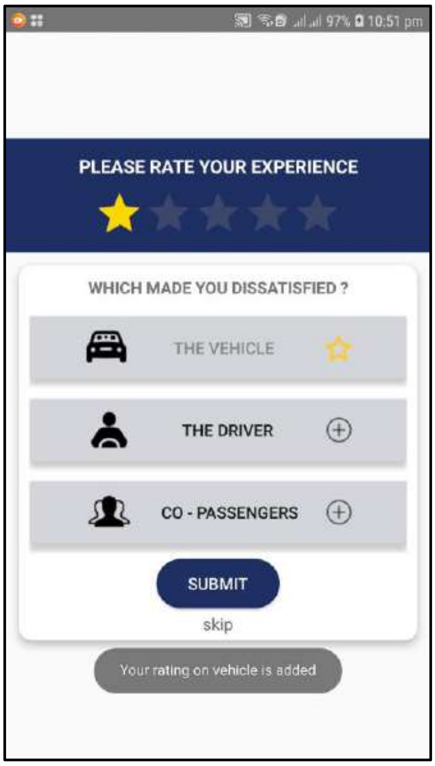


Figure 3.1.2.15 - After rating vehicle

Figure 3.1.2.16 - Keywords to rate driver

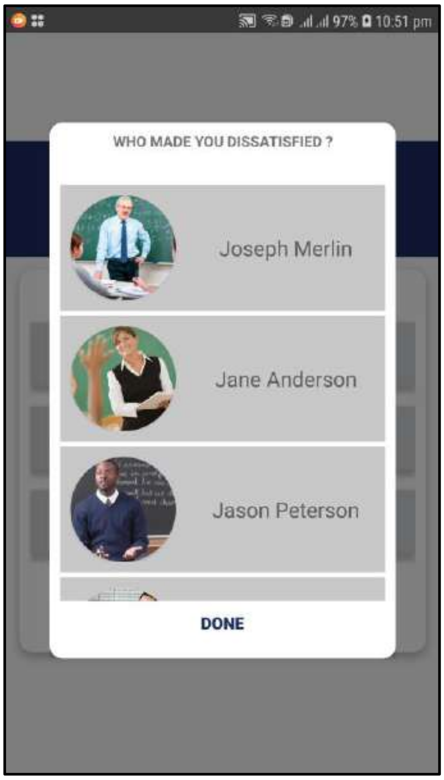


Figure 3.1.2.17 - Select co-passenger to be rated

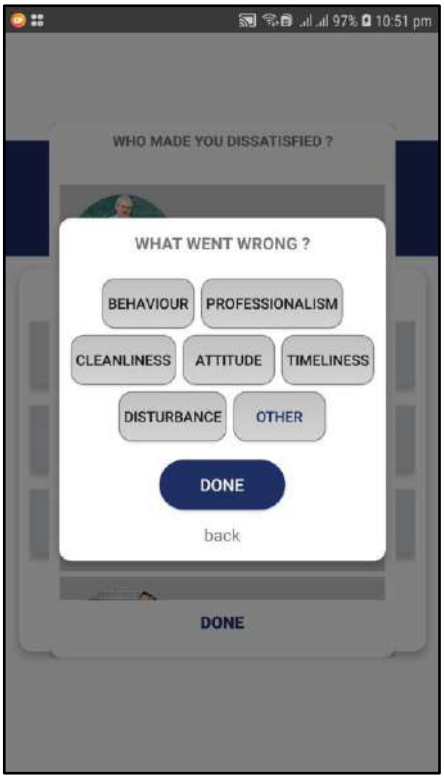
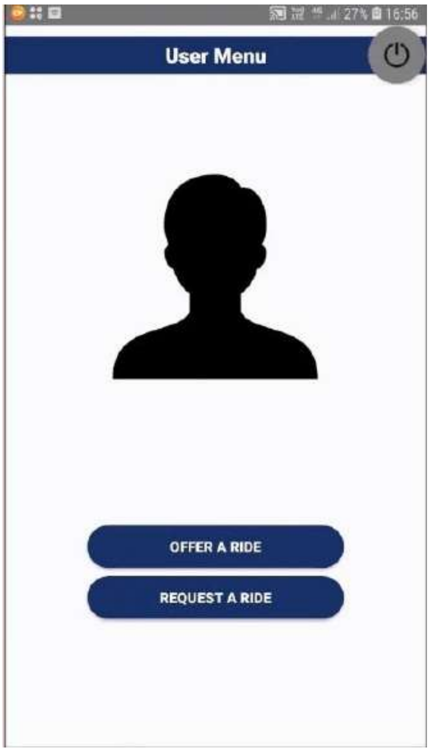


Figure 3.1.2.18 - Keywords to rate co-passenger

We have designed the mobile interfaces corresponding to OPR component. User Interfaces relevant to the optimum path recognition are as follows.

User Interfaces relevant to the driver in OPR.



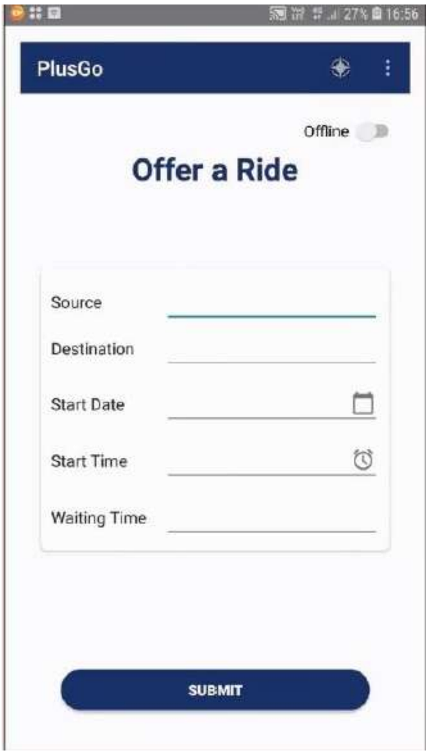


Figure 3.1.3.2 -Driver offer a ride

Figure 3.1.3.1 -Chose a role

Figure 3.1.3.2 -Passenger finds a ride

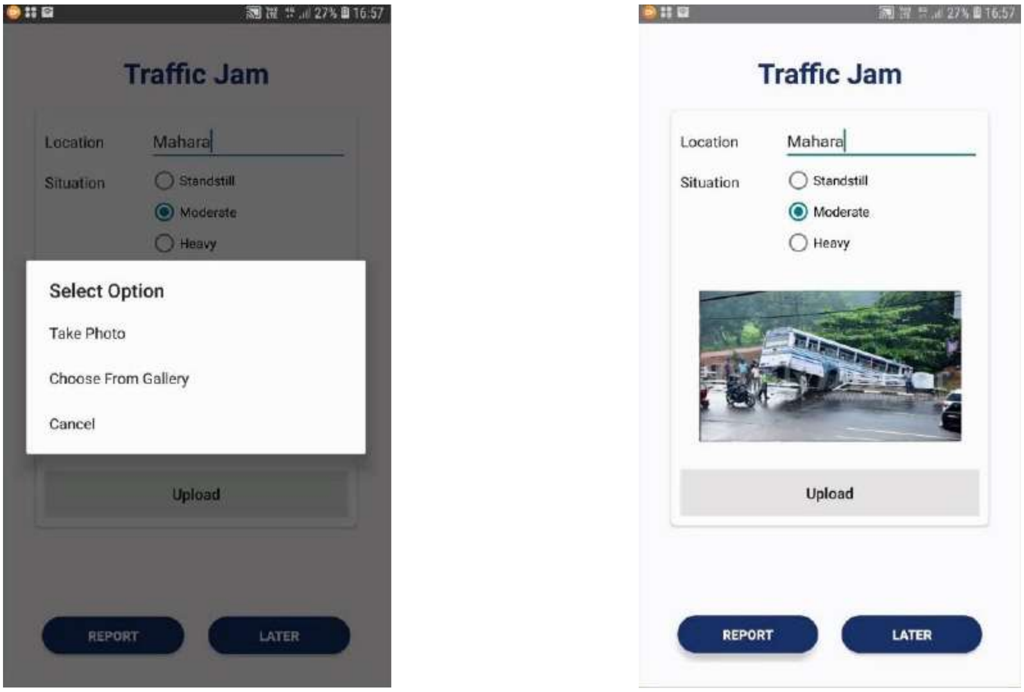


Figure 3.1.3.3 -Report traffic jam and accidents

User Interfaces relevant to the driver in FC.

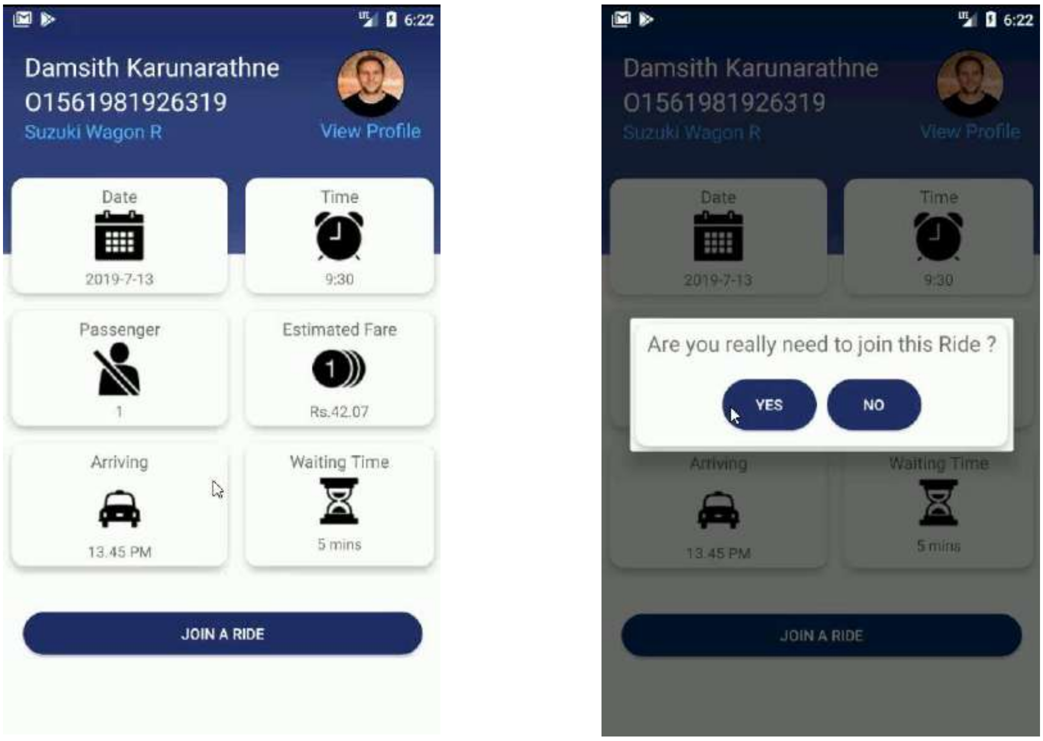


Figure 3.1.4.1 -Trip summary details and request to the ride

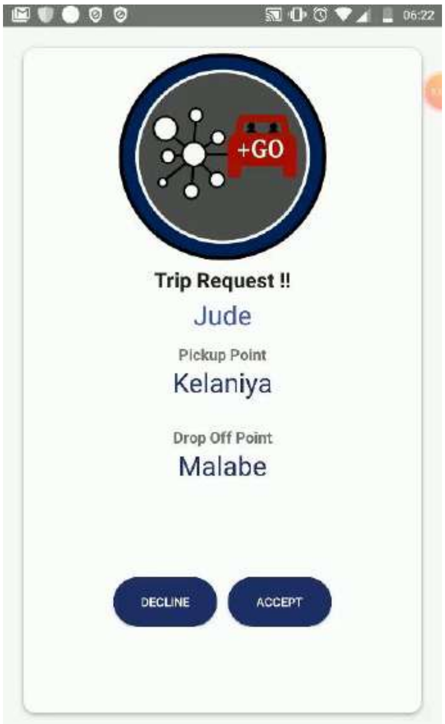


Figure 3.1.4.2 -Trip Requested Notification to the driver

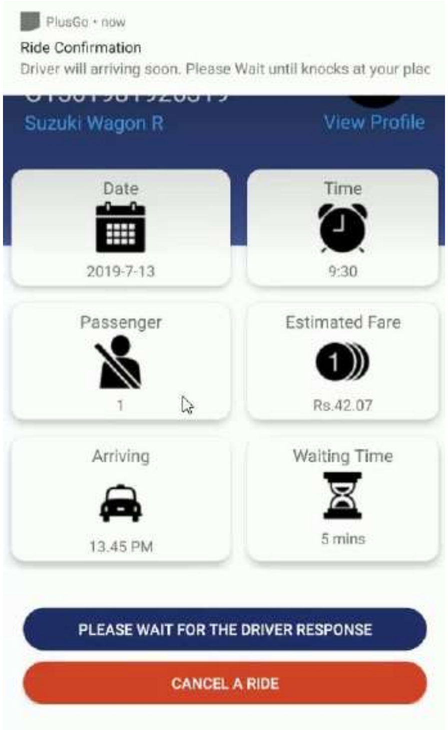


Figure 3.1.4.3 -Notification received

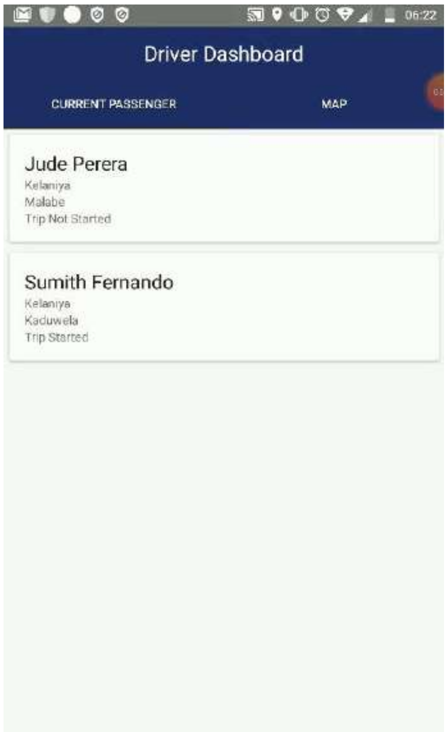


Figure 3.1.4.4 - Main dashboard of the driver

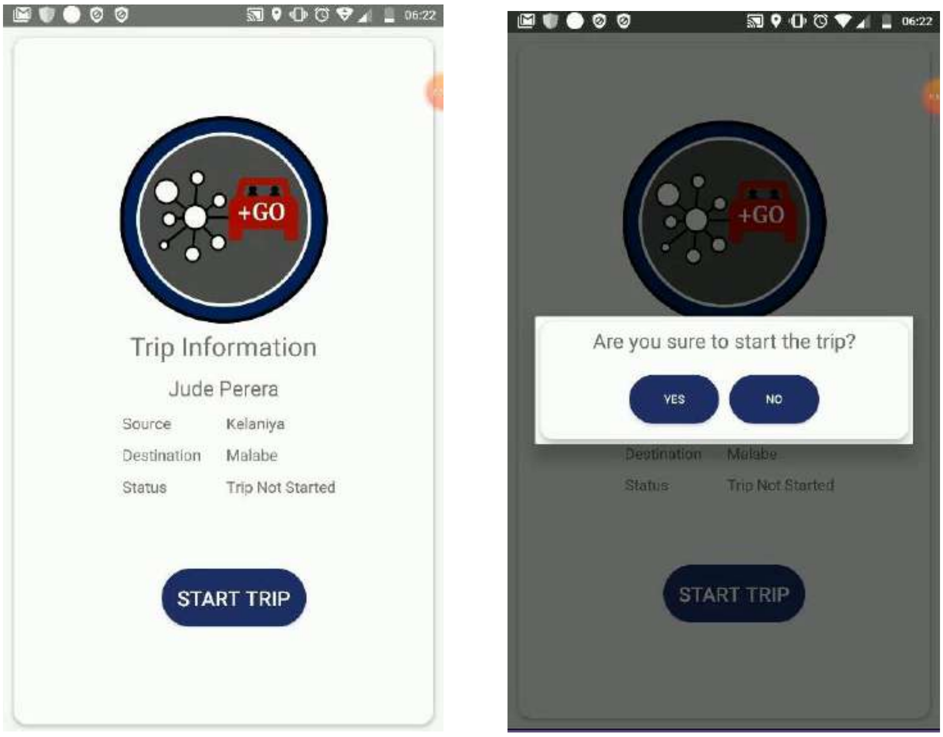


Figure 3.1.4.5 -Trip start

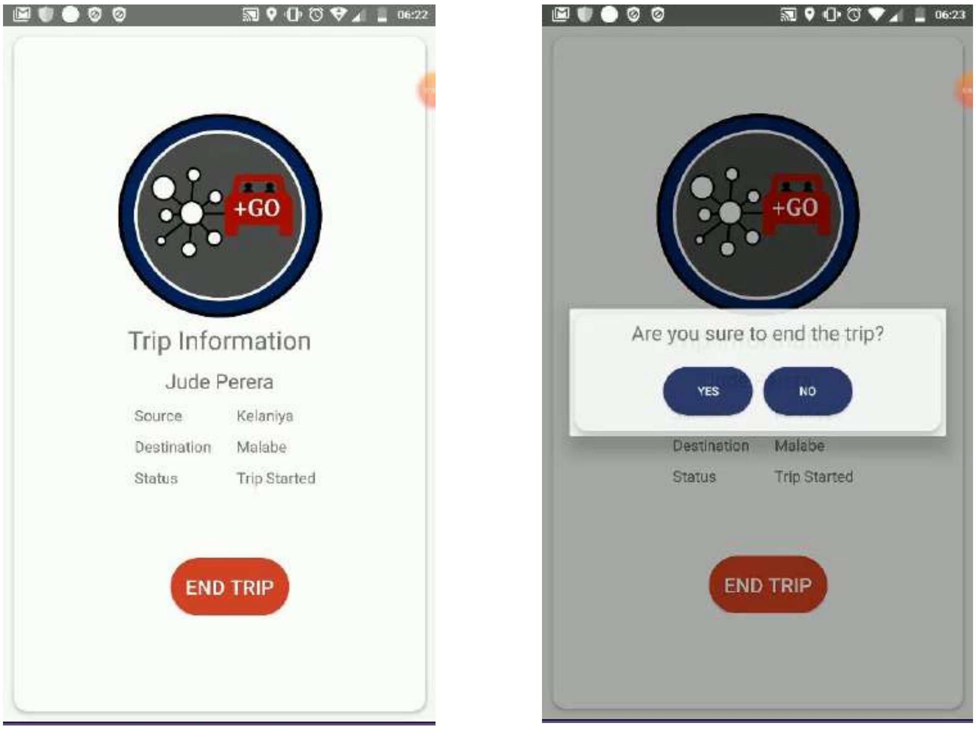


Figure 3.1.4.6 -Trip end

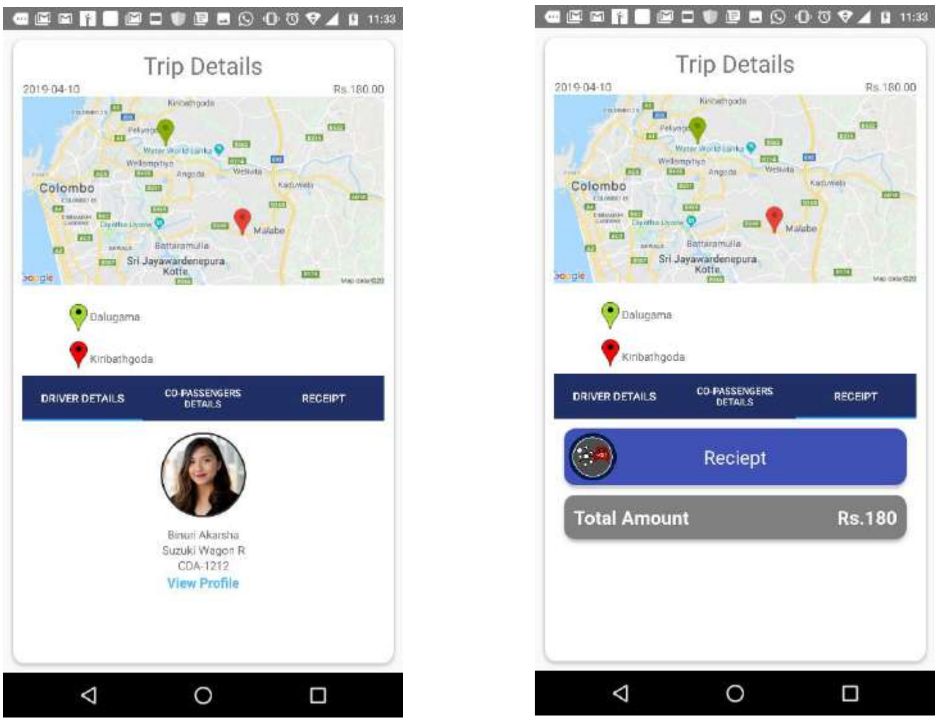


Figure 3.1.4.7 -Trip history details when you selected trip as a passenger

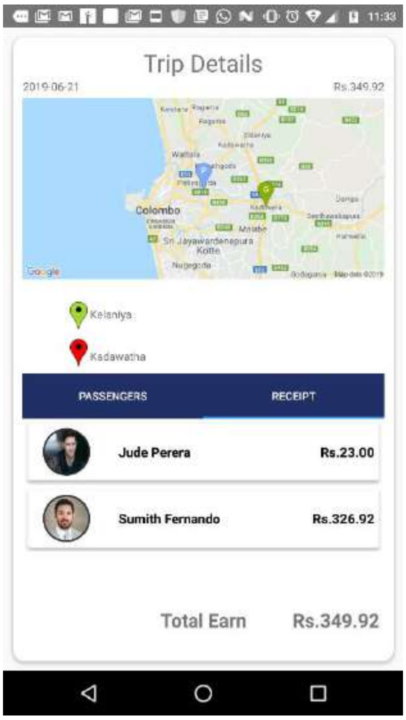


Figure 3.1.4.8 – trip history details when selected a trip as a driver

3.2 Research Findings

+Go application consists of many features and benefits to users who use our application. Throughout this document, the most important parts of the UPM has been discussed. In this section, it elaborates the findings from the study we have done. In the initial stage, we identified that the high number of private vehicles on the road is the major cause of traffic congestion. After conducting our literature survey, we identified that ridesharing is a possible solution to this problem. From the study, we determined that most people like to collaborate with ridesharing platforms, but they found no application in Sri Lanka could cater to their needs. With the solution, we came up. As a result, it reduces traffic congestion, travelling cost as well as environmental pollution catering the users' needs.

Out of the features we included in the application, validation of submitted images of national identity card and also the driving license made us clear that this feature will reduce the man hours wasted on manual verification strategies, and also the registration process does get faster. Also in the stage of development, a suggestion for optimizing the validation process by integrating with a service provided by the government was

proposed. This proposal can be tested for feasibility and adopted in future versions of the application, if suitable.
We also identified that adopting a customized rating system in ride sharing application is a dire need of all the users of existing applications in the world. The sentiment analysis function and allowing the passenger to block the driver in next ride sharing sessions can be considered as a value added features of the platform.
Also the complex logics were developed to run in the server side; made us clear that resource wastage of the mobile phone is minimized and also the effect on battery consumption is less from our application.

We have identified that the reliability of the product is way more important. The reliability of analyzing and identifying the most suitable optimum path to a certain destination will be acquired by using Dijkstra's Algorithm and the Travelling Salesman Problem(TSP) Algorithm. Most of the algorithms which are used in this component will be used in backend. As a result, backend server can be maintained easily. Also, most of the information related to this component are stored in real time database, firebase. Therefore if the registered users reset their mobile phones, users are capable of recovering their past data through the firebase database and meanwhile it will provide the database security too. Security of the proposed system can be maximized by maintaining and controlling the server side.

In fare, calculation component describes the, how to divide reasonable fare calculation under the ridesharing concept. It may be price comparatively lower than other taxi services in Sri Lanka because of that people's interest and try to get experience with service.

To read the fuel consumption of the vehicle, used OBD II ELM 327 adapter. In the mobile phone supports only Bluetooth and Wi-Fi among them, we choose Bluetooth adapter because since it was less expensive and typically has a lower power consumption as well as we can get a high amount of accuracy level from OBD reader.

In the application point of view, we found that the number of parameters included in the algorithm directly affects the accuracy of the results. Also, inbuilt methods in python helped a lot in implementing the algorithms easily. Therefore python is recommended in building complex logics in any study. Hence, with the technologies we used, the consumption of the battery of the mobile phone is very low. Since most of our complex calculations are done on the server-side, there was no overhead in the mobile phone, and as a result, our application consumes a negligible amount of battery power.

Our application increases the visibility of the users who registered to the application by providing comprehensive profile navigation to other users. Reducing traffic congestion reduces the man force as less amount of policemen needed to control the traffic. This helps the government reduce the money spent on labour. As our application is built on top of the android platform, any other android supported platforms can be integrated and can upgrade it to later versions easily.

3.3 Discussion

With the implementation of this proposed system, several new identifications were noted.

The primary focus of UPM is to introduce a new ride-matching algorithm to the users. Though it is stated in many places throughout this document, it is important to highlight that once again here. As mentioned previously, the accuracy of the ride-matching algorithm is recorded as 98%. In the initial stage, the accuracy was recorded around 78%, and several factors were changed including the structure of the algorithm, factors considered in clustering, rules defined and quantifiers used in the algorithm to get a more optimized solution. We gave priority to the clustering using K-Means clustering as it is simple and easy to develop. There are so many other clustering algorithms which can produce higher accuracy rate but complex. But in our context, using K-Means clustering was sufficient.

To identify the number of clusters in the clustering model, we used Elbow method. There are many other methods that are available to find the accurate number of clusters in a model. For example, Average silhouette method and Gap statistic method can yield a more accurate result than the Elbow method. Elbow method uses an iterative approach in finding the number of clusters and in our cases, iterations are set to 300.

For the identification of professional levels, we have defined a set of groups categorizing professions based on the level defined in the International Standard Classification of Occupations. The current version was known as the ISCO-08[52], and we have done some minor changes to it to suit our purpose.

The results of the experiment made it clear that the Tesseract does decent work in extracting text from the colour images. But it does not always perform well in extracting data from the color images. Hence the accuracy of the process was further increased when the images were subjected to noise reduction from erosion and dilation and then converted into grayscale and next into a binary image. Also, it was observed that the processing time of character extraction from grayscale images is reduced by 10% to 50% compared to colour images. This delivers significantly better results because Tesseract works fast with better text extraction accuracy when it comes to grayscale images and binary images. This result ties with previous studies wherein Chakraborty and Mallik (2013) explain, the more accurate recognition is seen when the color image is converted to grayscale. The table 2 below is a sample test result of text extraction from a low contrast image of a license, which was experimented in the analysis.

Image type (Noise removed)	**Property Extracted**	
	NIC number	***Expiry Date***
Raw image	Not Found	Not Found
Grayscale	Found (100%)	Not Found
Binary	Found (91.7%)	Found (100%)

Table 2: Text extraction from different types of same image

An apparent limitation of this image processing algorithm is that it depended heavily on the position of the image taken. Higher the background noise, lower the accuracy of the validation. Though we have taken steps to remove noise from the background after the image is uploaded, it's still not possible to remove all the noise in the image uploaded by the user. This introduces a potential confound in user to upload several versions of the image to get it validated from the system.

When we consider the sentiment analysis technique used in profile rating maintenance, results made it clear that Naive bayes algorithm does a smart job in calculating the probability of falling into a rating category. The approach used suffers from the limitation; accuracy of the test results sometimes get changed when the word density of the dataset does not have considerable amount of matching words to the sentiment provided at testing. In such a situation, the accuracy can be increased by adding similar types of sentiments in the training data set. When the vague sentiments are reduced and more relevant ones are added, the accuracy of the results gets increased.

A Geographic Information System (GIS) integrates hardware, software, and data for capturing, managing, analyzing, and displaying all forms of geographically referenced information. The GIS allows us to view, understand, question, interpret, and visualize data in many ways that reveal relationships, patterns, and trends in the form of maps, globes, reports, and charts [40]. Even there are several methods to identify the optimum path, Dijkstra’s algorithm and TSP algorithm will show the most accurate closest path. When comparing Google Map results and our proposed system results after entering the source and destination locations, Google Map will suggest several alternative routes while proposed system show only the optimum path by using Dijkstra's Algorithm by concerning the complete distance of each road and time to travel to the defined destined region. However, resulted routes from both instances are matched. When compared to Dijkstra's Algorithm, identifying the most suitable solution of a Travelling Salesman Problem is not achievable. A more accurate path can be generated by tracking the current location of the user.

The Primary objective of FC is to give a reasonable cost to both drivers and passengers. For those two types of fare calculation in different places. First one is the predicted fuel consumption that consumption display after selected the driver for the ride. That estimated need to some of the vehicle information (model year, registration year, engine capacity, fuel type, number of cylinders, engine power, mileage), no of current passengers, distance to the destination and current price of a liter of fuel. After surveyed vehicle information to predict fuel consumption according to the vehicle conditions. Initially surveyed with less number of features but it was not a success due to not enough variable with variations with fuel consumption.

OBD II ELM 327 Bluetooth adapter used to calculate the actual cost for the passenger. Initially, planned to read mileage and Instant fuel consumption (L/100km) from the OBD, but that was not a success after testing several vehicles. After, found an alternative solution for it. Idling fuel consumption, and Driving fuel consumption use instead of mileage, and instant fuel consumption.

Vehicle	**Mileage**	**Instant Fuel Consumption (L/100km)**	**Idling Fuel consumption**	**Driving fuel consumption**
Alto LXI	✗	✗	✗	✗
Toyota VIOS	✓	✓	✓	✓
Suzuki Wagon- R	not supported for few vehicles	Supported some of vehicle value not acceptable	✓	✓
Toyota Allion	✓	✓	✓	✓

Toyota Vitz	Supported some of vehicles not supported	✓	✓	✓
SsangYong Kyron	✓	✓	✓	✓

Table 3.3.1 Comparison the features with OBD II using different vehicles

Testing of our application is done thoroughly in both the backend and frontend. As stated before, the backend was tested with some cases which were identified before and compared the results with the expected results. For the frontend, many test cases were executed manually, and actual and expected results were compared. Application is alpha tested and planned to release it for beta testing. After the beta testing, the application will be officially released to the users. Technologies used for the development were Android for the frontend and python and node.js for the backend. The main reason to select python as the language to implement algorithm was the resources and libraries it provides in support of the complex logic development. With the high support provided from the python, it was easy to develop the algorithms. For the real-time data synchronization, firebase was used over the MySQL as firebase stores the data in large JSON documents rather than storing in tables like MySQL. Thereby overhead in updating the same record every time is pretty much lower in firebase compared to the Relational databases. For the normal data storage, MySQL was used as it provides integrity constraints which are useful in this kind of application.

4. **Summary of Each Student's Contribution**

Name	**Contribution**
V.A.Wickramasinghe	User Profile Management
A.E.Edirisinghe	Document Validation and Profile Rating Maintenance
G.L.S.R. Gunawardena	Fare Calculation
R.M.A.N.Gunathilake	Optimum Path Recognition

5. CONCLUSION

Since traffic congestion has aggravated by the society as a major concern, we introduced a new ride-sharing application: +Go; Intelligent Complementary Ride-Sharing System which can cater to this need. As there are no other application in the world possesses features of our applications mentioned previously. This application will make a new revolution in the IT industry. As mentioned, we focused only on professionals and their professional level. From our ride-matching algorithm, passengers will get drivers whom they are desired, and no dissatisfaction is granted.

With the technologies used, we were able to minimize the battery consumption of the phone, which is an added advantage for any user. It is important to highlight that; we planned to take the feedback from the professionals about the profession of the co-users to verify their profession. As there is no other mechanism integrated with our system for the verification, this will be implemented in the next version of our application. One possible limitation was the identification of the validity of the person. During the identification process, any user can submit fake documents to our application. Therefore, we planned to take the help of the government information providers to verify the identity of the particular users who are registering to the application. Finally, it is important to emphasise the fact that "Intelligent Complementary Ride Sharing System" which is also known with the application name "+Go" can become a unique and effective solution to reduce traffic congestion in Sri Lanka.

6. REFERENCES

[1] [online] Available at:
http://www.worldometers.info/world-population/sri-lanka-population/
Department of Economic and Social Affairs, Population Division. World Population
Prospects: The 2017 Revision. (Medium-fertility variant). [Accessed 10 Jan. 2019].

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

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

[4] Vehicleinfo.000webhostapp.com. (2019). Vehicle Details. [online] Available at:
https://vehicleinfo.000webhostapp.com/

[5] Dewan, K.K. & Ahmad, Israr. (2007). Carpooling: A Step to Reduce Congestion
(A Case Study of Delhi). Engineering Letters [Accessed 5 Jan. 2019].

[6] Ding, C.; Wang, D.; Liu, C.; Zhang, Y.; Yang, J. Exploring the influence of built
environment on travel mode choice considering the mediating effects of car ownership and travel distance. Transp. Res. Part A Policy Pract. 2017, 100, 65–80.
[CrossRef]

[7] Efthymiou, D.; Antoniou, C.; Waddell, P. Factors affecting the adoption of vehicle sharing systems by young drivers. Transp. Policy 2013, 29, 64–73

[8] Rayle, L.; Dai, D.; Chan, N.; Cervero, R.; Shaheen, S. Just a better taxi? A survey-based comparison of taxis, transit, and ridesourcing services in San Francisco. Transp. Policy 2016, 45, 168–178

[9] R. Manzini and A. Pareschi, “A Decision-Support System for the Car Pooling Problem,” journel on transportation technologies, Vol.2 No.2 ,2012,pp. 85-101.

[10] Swati. R. Tare, Neha B. Khalate and Ajita A. Mahapadi, "International Journal of Advanced Research in Computer Science and Software Engineering 3(4)", ISSN: 2277 128X April - 2013, pp. 54-57

[11] K. A. Abdul Nazeer, M. P. Sebastian, "Improving the Accuracy and Efficiency of the k-means Clustering Algorithm", Proceedings of the World Congress on Engineering 2009 Vol I, 2009 [Accessed 31 Dec. 2018].

[12] Chih-Ping Wei, Yen-Hsien Lee and Che-Ming Hsu, Empirical Comparison of Fast Clustering. Algorithms for Large Data Sets,2000, Proceedings of the 33rd Hawaii International Conference on System Sciences – 2000.

[13] Ketchen Jr DJ, Shook CL. The Application of Cluster Analysis in Strategic Management Research: An Analysis and Critique. Strategic Management Journal. 1996; p. 441-458

[14] R. Valiente, M. T. Sadaike, J. C. Gutiérrez, D. F. Soriano, G. Bressan and W. V Ruggiero, “A Process for Text Recognition of Generic Identification Documents Over Cloud Computing,” 2016.

[15] R. Valiente, M. T. Sadaike, J. C. Gutiérrez, D. F. Soriano, G. Bressan and W. V. Ruggiero, “A Process for Text Recognition of Generic Identification Documents Over Cloud Computing,” 2016.

[16] A. Parwar, A. Goverdhan, A. Gajbhiye, P. Deshbhratar ,R. Zamare and P. Lohe, “Implementation to Extract Text from Different Images by Using Tesseract Algorithm,” International Journal Of Engineering And Computer Science ISSN: 2319-7242, vol. 6 Issue 2, Page No. 20298-20300, Feb. 2017.

[17] P. Chakraborty and A. Mallik, “An Open Source Tesseract based Tool for Extracting Text from Images with Application in Braille Translation for the Visually

Impaired," International Journal of Computer Applications (0975 – 8887), vol. 68,
no.16, April 2013.

[18] K. N. Natei, J. Viradiya and S. Sasikumar, "Extracting Text from Image Document and Displaying Its Related Information", K.N. Natei Journal of Engineering Research and Application, vol. 8, pp. 27-33, May 2018.

[19] A. Mordvintsev and K.Abid, OpenCV-Python Tutorials Documentation, Nov 05
2017.

[20] A. Clark, Pillow (PIL Fork) Documentation, Release 5.3.0, Oct 24 2018.

[21] S. Ramiah, T. Y. Liong and M. Jayabalan, "Detecting text based image with optical character recognition for English translation and speech using Android," 2015
IEEE Student Conference on Research and Development (SCOReD), Kuala Lumpur,
2015, pp. 272-277.

[22] 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.

[23] P. Baid, A. Gupta and N.Chaplot, "Sentiment Analysis of Movie Reviews using
Machine Learning Techniques", 2017.

[24] Llombart, Òscar Romero, "Using Machine Learning Techniques for Sentiment
Analysis," 2017.

[25] L. Dey, S. Chakraborty, A. Biswas, B. Bose and S. Tiwari,"Sentiment Analysis
of Review Datasets Using Naïve Bayes‘ and K-NN Classifier,"International Journal of Information Engineering and Electronic Business, vol. 6, Issue 4, pp. 54-62, 2016.

[26] 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.

[27] P. Baid, A. Gupta and N.Chaplot, “Sentiment Analysis of Movie Reviews using
Machine Learning Techniques”, 2017.

[28] Vidushi, G. S. Sodhi,”Sentiment Mining of Online Reviews Using Machine Learning Algorithms,” International Journal of Engineering Development and Research (IJEDR), ISSN:2321-9939, vol.5, Issue 2, pp.1321-1334, May 2017.

[29] M. H. Hassan, S. P. Shakthi and R. Sasikala,”Sentimental analysis of Amazon
reviews using naïve bayes on laptop products with MongoDB and R,” IOP Conference Series: Materials Science and Engineering, 2017, vol. 263, pp. 42090.

[30] 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

[31] 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

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

[33]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

[34] Adewole Philip, Akinwale Adio Taofiki and Otunbanowo Kehinde, “A Genetic Algorithm for Solving Travelling Salesman Problem,” (IJACSA) International Journal of Advanced Computer Science and Applications,vol. 2, no.1, January 2011

[35] Dweepna Garg and Saurabh Shah, “Ant Colony Optimization For Solving Traveling Salesman Problem,” *International Journal of Computer Science and System Analysis,*vol. 5, no. 1, January-June 2011, pp. 23-29, ISSN 0973-7448

[36] M.I.C., Tunon and M.R. Lopez. ”Design and Use of the CPAN Branch and Bound for The Solution of the Travelling Salesman Problem (TSP),” International Conference on Electronics, Communication and Computers, Conielecomp, 1 August 2005.

[37] Pragya, M.Dutta, and Pratyush. “TSP Solution Using Dimensional Ant Colony Optimization,” International Conference on Advanced Computing & Communication Technologies, 6 April 2015.

[38] M.M. Mena, R.H. Ucan, V.C. Vetina, and F.M. Ramirez, “Web Services Composition using the Bidirectional Dijkstra Algorithm,” IEEE Latin America Transaction, vol. 14, Issue 5, May 2016

[39] A. Ratnasari, F. Ardiani, F. Nurvita. “Penentuan Jarak Terpendek dan Jarak Terpendek Alternatif menggunakan Algoritma Dijkstra serta Estimasi Waktu

[40] What is GIS? [Online].Available: http://www.esri.com/what-is-gis/overview.html# overview panel [Accessed: 9th September 2012]

[41] Zoepf, S., Chen, S., Adu, P. and Pozo, G. (2013). *The Economics of Ride-Hailing: Driver Revenue, Expenses and Taxes*. [ebook] MIT Center for Energy and Environmental Policy Research, p.1. Available at: https://orfe.princeton.edu/~alaink/SmartDrivingCars/PDFs/Zoepf_The%20Economics%20of%20RideHialing_OriginalPdfFeb2018.pdf

[42] O. Santos, D. and C. Xavier, E. (n.d.). *Dynamic Taxi and Ridesharing: A Framework and Heuristics for the Optimization Problem*. Eduardo C. Xavier, p.1.

[43] Riquelme, C., Banerjee, S. and Johari, R. (n.d.). *Pricing in Ride-share Platforms: A Queueing-Theoretic Approach*. [ebook] pp.6,7,15. Available at: http://www.columbia.edu/~ww2040/8100F16/Riquelme-Johari-Banerjee.pdf [Accessed 6 Jan. 2019].

[44] Aleksandar, P., Silvana, P. and Pancovska Valentina, Z. (2015). *Multiple Linear regression model for predicting bidding price*.

[45] URBAN TRANSPORT SYSTEM DEVELOPMENT PROJECT. (2014). FOR COLOMBO METROPOLITAN REGION AND SUBURBS. [online] Colombo , Sri Lanka: DEMOCRATIC SOCIALIST REPUBLIC OF SRI LANKA MINISTRY OF TRANSPORT, pp.121 ,150. Available at: http://www.transport.gov.lk/web/images/downloads/F-CoMTrans_Main_S.pdf[Accessed 11 Mar. 2019].

[46] Face detection using OpenCV, Software.intel.com, 2019. [Online]. Available: https://software.intel.com/en-us/node/754941. [Accessed: 09- Mar-2019].

[47] Army.lk, 2019. [Online]. Available: https://www.army.lk/html/images/image/News%20Highlight/idcard.jpg. [Accessed: 05- Mar- 2019].

[48] S3.ap-south-1.amazonaws.com, 2019. [Online]. Available: https://s3.ap-south-1.amazonaws.com/techleer/204.png. [Accessed: 09- Apr- 2019].

[49]TO THE NEW BLOG. (2019). *Push notifications using Firebase Cloud Messaging | TO THE NEW Blog*. [online] Available at: http://www.tothenew.com/blog/push-notifications-using-firebase-cloud-messaging/ [Accessed 12 Aug. 2019].

[50]Bandara, H., Azeez, A., Amarasinghe, M., Kottegoda, S., Liyana Arachchi, A. and Muramudalige, S. (n.d.). *Cloud-Based Driver Monitoring and Vehicle Diagnostic with OBD2 Telematics*. [online] Dr. Dilum Bandara, p.22. Available at: http://dilum.bandara.lk/wp-content/uploads/2017/04/Final-Report-OBD2.pdf [Accessed 6 Aug. 2019].

[51] H. Tsao, D.-J. Lin, Spatial and temporal factors in estimating the potential of ride-sharing for demand reduction, Tech. Rep., Institute of Transportation Studies, University of California, Berkeley (1999).

[52] "International Standard Classification of Occupations", En.wikipedia.org, 2019. [Online]. Available: https://en.wikipedia.org/wiki/International_Standard_Classification_of_Occupations. [Accessed: 14- Mar- 2019].

7. APPENDICES

Appendix A: Survey Questionnaire

This is a survey conducted to understand the transportation patterns of the office staff and to understand their interest on Carpooling.

Carpooling (car-sharing, ride-sharing and lift-sharing) is the sharing of car journeys so that more than one person travels in a car, and prevents the need for others to have to drive to a location themselves.[wikipedia]

Personal Information

1) How do you travel between your workplace and home in each day ?

- ❑ I drive
- ❑ I use public transport
- ❑ I walk
- ❑ I get a lift
- ❑ I don’t need to commute

** If you are walking ,skip to question 20 .*

2) What is your position in the job?

- ❑ Trainee
- ❑ Driver
- ❑ Engineer (Software /Civil /Mechanical)
- ❑ Clerk
- ❑ Accountant
- ❑ Manager

- ❑ CEO / CFO / CIO
- ❑ Other ___________________

3) Please specify your gender

- ❑ Male
- ❑ Female

4) Please specify your age group

- ❑ 17 - 21
- ❑ 22 - 30
- ❑ 31 - 40
- ❑ 41 - 50
- ❑ 51 - 60
- ❑ 60+

5) Which is the OS of the mobile phone you use?

- ❑ Android
- ❑ iOS
- ❑ Windows
- ❑ Other _____________________

6) Do you have a vehicle?

a)Yes.

b) No.

7) Vehicle type you own?

- ❑ Car
- ❑ Van
- ❑ Cab
- ❑ SUV
- ❑ No preference
- ❑ Other ___________________

8) Vehicle Brand

- ❑ Honda
- ❑ Toyota
- ❑ Nissan
- ❑ Suzuki
- ❑ Benz/BMW
- ❑ Other _______________

9)Fuel Type

- ❑ Petrol

- ❑ Diesel
- ❑ Hybrid
- ❑ Electric
- ❑ Other ___________

10)Fuel Consumption(km/l)

- ❑ less than 5
- ❑ 5-7
- ❑ 7-9
- ❑ 9-12
- ❑ 12-15
- ❑ greater than 15

11)What is the vehicle type you prefer to travel the most?

- ❑ Car
- ❑ Van
- ❑ Cab
- ❑ SUV
- ❑ No preference

12) Have you ever carpooled (either offered a ride in your vehicle or took a ride in others' vehicle)?

- ❑ Yes, on an informal basis
- ❑ Yes, on a formal paying basis
- ❑ No, I have never had a car sharing arrangement

13) Would you carpool in your car if you get paid by the people you are offering ride to?

- ❑ Yes
- ❑ I don't own a car but yes, I'd like to carpool
- ❑ I don't own a car & I wouldn't carpool
- ❑ Other ___________

14) Would you be interested in a collaboration between Carpool platforms

- ❑ Yes
- ❑ No
- ❑ Maybe

** If you are not interested in carpooling ,skip to question 20.*

Carpooling Information

15)What would be your primary reason for opting for car sharing?

- ❑ To save car ownership costs
- ❑ To save fuel costs
- ❑ To meet interesting people

- ❑ For the benefit of the environment
- ❑ To avoid the hassle of looking for a parking space
- ❑ To make the journey to and from work quicker
- ❑ To minimize traffic congestion
- ❑ Other __________

16)What are the important factors to be considered before choosing car sharing service in the security and comfort perspective of the car owner?

(please tick the relevant)

	Very Important	Important	Neither important or unimportant	Unimportant	Not at all Important	Don't know
Type of vehicle						
Number of people car sharing in a given period of time						
Previous user reviews						
Luggage(s) allowed(size,etc)						
Insurance details						

Brief profile (Having access to their social profile,like Facebook etc)							

17)Please rate the following incentives that could be used to encourage the car sharing service.

- On a scale of 1 to 10,how comfortable would you be giving a stranger a lift?

  (1 - not comfortable , 10 - very comfortable)

  1( ) 2( ) 3( ) 4( ) 5( ) 6( ) 7( ) 8( ) 9( ) 10( )

- On a scale of 1 to 10,how comfortable would you be receiving a lift off a stranger?

  (1 - not comfortable , 10 - very comfortable)

1( ) 2( ) 3( ) 4( ) 5( ) 6( ) 7( ) 8( ) 9( ) 10( )

18)What type of person would you consider having a formal car sharing arrangement with?

- ❑ A solo female driver
- ❑ A solo male driver
- ❑ An accompanied female driver
- ❑ An accompanied male driver

❏ I have no preference

19)What would you consider when calculating the pricing for a shared journey?

❏ Cost per Kilometer

❏ Depending on the distance and frequency ,agreeing for a fixed cost with the vehicle owner .

❏ Other(specify) ___________________________

20)Suppose you travelling companion is working in a company where he use to travel more than your work place.What would you prefer (The total travelling cost gets divided among the number of passengers) ?

❏ Drop the companion at his office by travelling the extra distance

❏ Drop the companion at your working place and asking him to find another ride.

❏ Other ____

21)What are the main reasons for not interested in sharing a ride with others?

❏ I like the independence of having my car.

❏ Carpooling takes too much time.

❏ I like the privacy of driving alone.

❏ I need my car for business reasons.

❏ I need my car for other personal reasons before or after work.

❏ Inconvenient to wait for others.

❏ I can’t have flexible work times.

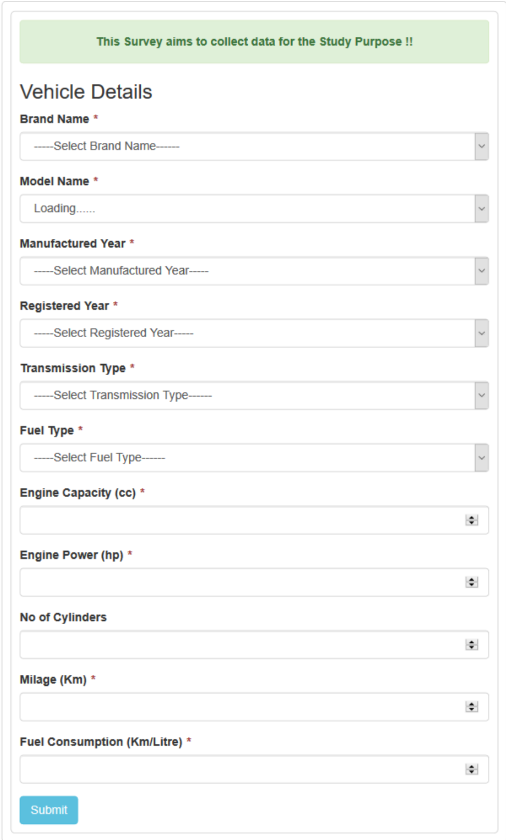
- ❑ I have an irregular work schedule.
- ❑ Other (specify) ______________________________

22) Do you think carpooling is a good option for Sri Lanka?

- ❑ Yes, I do
- ❑ No, I don't
- ❑ I don't have an idea

- Any Additional Comments

Appendix B: Survey to collect fuel consumption details



This Survey aims to collect data for the Study Purpose !!

Vehicle Details

Brand Name *

-----Select Brand Name------

Model Name *

Loading......

Manufactured Year *

-----Select Manufactured Year-----

Registered Year *

-----Select Registered Year-----

Transmission Type *

-----Select Transmission Type------

Fuel Type *

-----Select Fuel Type------

Engine Capacity (cc) *

Engine Power (hp) *

No of Cylinders

Milage (Km) *

Fuel Consumption (Km/Litre) *

Submit