INTELLIGENT COMPLEMENTARY RIDE SHARING SYSTEM

V.A.Wickramasinghe

IT16030190

BSc (Hons) in Information Technology

Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

August 2019

INTELLIGENT COMPLEMENTARY RIDE SHARING SYSTEM

V.A.Wickramasinghe

IT16030190

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

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

August 2019

DECLARATION

I declare that this is my work and this dissertation doesn't incorporate without acknowledgment any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except the acknowledgment is made in the text.

Also, I 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. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: _________________ Date: ___________________

(V.A.Wickramasinghe)

The above candidate has carried out research for the BSc Dissertation under my supervision.

Signature: _________________ Date: ___________________

(Dr. Janaka Wijekoon)

ABSTRACT

With the high number of vehicles that enter the city limits, traffic congestion has become a significant concern in society. Most of the vehicles are underutilized since only one or two people travel at once. To reduce this underutilization and traffic congestion, we proposed a new platform using the ridesharing concept: +Go. The Intelligent Complementary Ride-Sharing System. As this solution only targets the professionals, it is indeed necessary to suggest drivers who are in the same professional category as the passenger. For that, we introduced a new user profiling mechanism with the ride-matching algorithm to suggest the most suitable drivers based on the passenger's professional level and preferences. In here, we would only suggest drivers whose professional level is equal to or higher than the level of the passenger. We consider preferences like gender preference, smoking condition, vehicle comfortability, language preference, and motion sickness preferences of the passenger in selecting the most suitable driver list. The ultimate purpose of this document is to provide a detailed description of the User Profile Management in the +Go application and how it affects the success of it.

Keywords— Ride-sharing, Machine Learning, Ride matching, User Profiling, K-Means

Clustering

ACKNOWLEDGEMENT

I would like to extend my sincere gratitude to Dr.Janaka Wijekoon for the constant guidance throughout the project as the supervisor, Dr.Darshana 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 ABBREVIATIONS

UPM	**User Profile Management**
GDP	**Gross Domestic Product**
OTP	**One-Time Password**
API	**Application Programming Interface**
GPS	**Global Positioning System**

LIST OF APPENDICES

1. INTRODUCTION

1.1 Introduction

According to the statistics, over 400000 private vehicles enter the Colombo area[1]. One possible way to reduce this amount of vehicles is to travel by public transportation. Public transportation can carry many people at once so that the number of private vehicles on the road get reduces. Usually, the bus can take 33.6 persons on average while underutilized private vehicles carry 1.87 persons at once[1]. Hence the bus carries 17 times more than the individual vehicles which can have a significant impact in reducing the traffic congestion. But in Sri Lanka, public transportation is not up to the expectations of the people. Therefore, people tend to travel to their workplaces using their private vehicle despite the traffic congestion giving priority to their comfortability, expectations, and professional level. Due to this, the amount of money and time wasted on the road is high. Quantitatively 500 million is lost daily according to the transport authorities[2].

Hence, environmental pollution increases, causing an adverse effect on living beings. For instance, the number of private cars increased from 499,714 in 2012 to 756,856 in 2017, which is 51% rose[3]. While the bus population increased from 91,623 to 107,435 in corresponding years, which is comparatively lower than the private cars. In the next six years, it is likely to increase the ratio between person to the vehicle into a 1:2 ratio according to the statistics[3]. With the rising population and their needs cannot be fulfill with the current inadequate infrastructure facilities and public transportation modes. Hence, people use their private vehicles to travel to their working places increasing the density of vehicles on the road.

As a result of an in-depth study about the context; the new ridesharing application is introduce with ride-sharing concept to reduce this traffic congestion. From this application, it reduces the vehicles that are underutilized and increases the ratio of people who travel in a single car. Thereby, the number of private vehicles get reduces. The main advantage of this concept is that, people don't need to get fed up of traveling.

They can travel in their own comfortability by joining with a set people who travel to a similar destination. From our proposed ride-sharing application, we combine the people who go to similar destinations at a particular time interval into a single location mainly using User Profile Management.

1.2 Background Literature

From the survey we conducted with around 150 professionals, we identified most of the professional think ridesharing can reduce traffic congestion(72.5%) and save their fuel cost(68.3%). It is clear that ridesharing can be used as a solution to minimize traffic congestion in Sri Lanka, where they suggest more reasons like reduce environment pollution, save car ownership costs and to meet interesting people, they opt to use ridesharing.

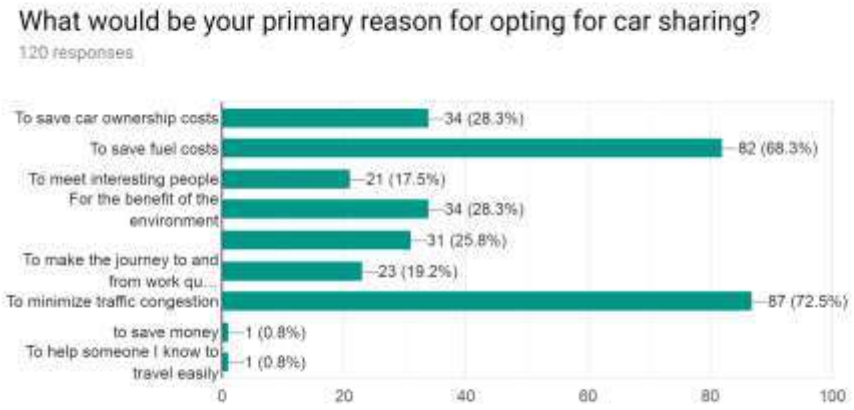

Figure 1.2.1 Reasons for opting Ridesharing

Furthermore, more than 61% of the professionals like to collaborate with the ridesharing platform, which is a significantly higher amount. Therefore, we decided to develop a new app using ridesharing concept. To do that, we associated many pieces of literature related to the field to identify the possible technologies and algorithms that can be used for the development process.

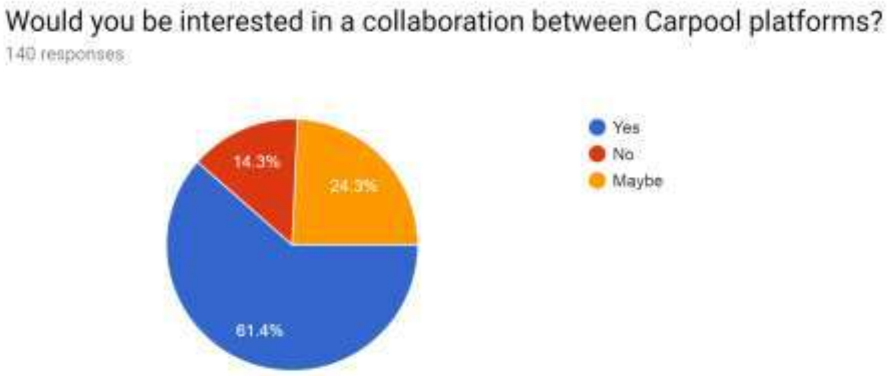

Figure 1.2.2 Percentage of Collaborations in Ridesharing Platforms

Hence, we considered the gender preference of the professionals while using ride-sharing platform. Most of the professionals were having no issue with gender, and they like to travel with either male or female (63.3%). As well, some people only want to go with a particular gender. Precisely, 11.7% only with the female, and 9.2% only with the male, etc. Therefore, by considering this fact, we decided to provide gender preference to our application users during the registration, where they can select whatever the gender preference they like.

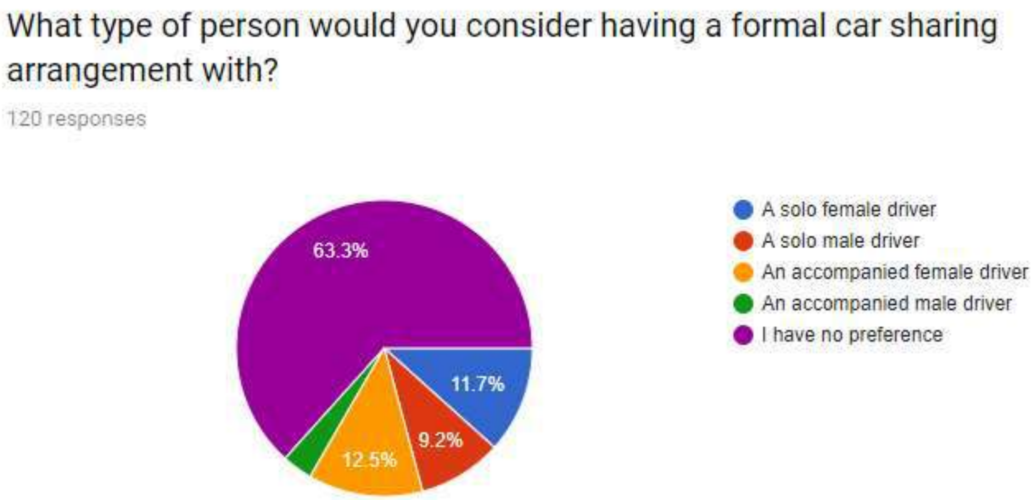

Figure 1.2.3 Percentage of Gender Preference

With our proposed system, we optimize the solutions provided by the previous studies by proposing only the drivers, who match the preferences and professional level of the passenger. The initial concept was studied from the research done by Swati, Neha, and Ajita[4] where they suggested drivers based on the ratings of drivers and provided more flexible user experience to the users. In our application: **+Go**, we used that basic concept and enhanced it to a more flexible solution which can provide the users with a better experience in using ride-sharing applications. Similar to the study of Fagin and Williams(1983), we considered gender preference when suggesting the drivers to the users. This was introduced as an initiative to increase the security of the users of the application. This feature is beneficial, especially for women who like to travel only with women and vice versa.

In recent days most of the ridesharing applications are either classic or dynamic model which works on smartphones with integrated GPS, which makes it possible to assume that the online users are always trackable and have permanent connectivity and reachability according to the study done by Furuhata et al. (2013) and Stach(2011). This increases the capability to generate driver lists which are accurate to pinpoint. Hence, we used a similar technique to identify the current location of the driver real-time by utilizing the polling technique. It synchronizes with the database and updates the current position of the online users in the given time interval[10]. However, Winter and Nittel(2010) proposed an ad hoc methodology to get the location using the Mobile Geosensor Networks, and it is being implemented in some ridesharing applications as well.

As a grouping of drivers was needed in the ride-matching algorithm, we decided to use K-Means clustering which was used by Abdul and Sebastian[4] in their study and proved the accuracy of 78% with their dataset. They also enhanced some features in the K-Means and confirmed that the new Enhanced version of the K-Means yield accuracy over 89%. From the study done by Chih-Ping, Yen-Hsien and Che-Ming 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[5]. From the literature, it is found that the number of clusters in a dataset can be easily calculated using the Elbow method. From the study of Ketchen and Shook[6], they have analyzed Elbow method returns accurate value depending on the dataset used. In our study, we used the Elbow method to calculate the number of clusters and K-Means clustering to cluster the drivers and passengers into separate groups.

1.3 Research Gap

According to the research we have done, we found out that many ride-sharing applications are prevailing in Sri Lanka. Most of them are failures due to the problems associated with technology used, concept, and problems associated with quality and targeted audience, etc. Therefore, we considered all these factors before developing our ride-sharing application. In this section, a full description of features in our application is given with the comparison with the existing applications.

Table 1.3.1 - Comparison of Existing Products

Features	Uber	UDIO	Carpooling.lk	RideShare.lk	+Go
Focused only on professionals	✗	✗	✗	✗	✓
Matching the passengers' profile with drivers	✗	✗	✗	✗	✓
Allow the spouse/guardian to check the passenger’s trip details	✗	✗	✗	✗	✓
The suggestion of drivers per passenger profession and preferences	✗	✗	✗	✗	✓
Consider gender preference when registering to provide high security	✗	✓	✗	✗	✓

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.

1.4 Research Problem

It is a common fact that Sri Lanka heavily suffers from traffic congestion, economically and socially. From the side of a countries economy, it is found out that Rs.60 billion is lost due to this traffic congestion and exactly Rs.500 million is lost daily according to transport authorities including 1.5 GDP loss[2]. Not only that, traffic congestion has a bad influence on the environment as well. When the number of vehicles on the road increases significantly, the amount of CO_2 emits from the vehicles to the environment also increases. Quantitatively 0.88555 metric tons per capita[7]. This has an adverse effect on living beings in the long term process. Hence, when it takes a long time to travel to the working places, the stress of the people increases, and their productivity level get decreases. Meanwhile, the amount of fuel burnt onto the road increases having a lousy effect economically and socially.

The main reason behind this traffic congestion is the number of vehicles that enter the city limits. The best possible solution to reduce the number of vehicles on the road is to travel by public transportation. Since, public transport in Sri Lanka at its worst condition, people go into their car considering their comfortability and professional level despite the traffic congestion at rush hours. But, what most of the people don't know is the fact that how much of money they wasted onto the road due to this traffic congestion. With the sizes of Sri Lankan roads and their conditions, we need to reduce the number of vehicles traveling in a single way to a lower number of vehicles to reduce this traffic congestion. Notably, during the rush hours like morning and evening, massive traffic queues are seen, especially in Colombo.

Therefore, we thought to introduce a new **solution which can minimize traffic congestion** providing the same comfort level and expectations of the people. Hence, we identified that ridesharing is the best possible solution to traffic congestion in Sri Lanka.

1.5 Research Objectives

1.5.1 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 traveling 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.

1.5.2 Specific Objectives

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.

1.5.2.1 Operations

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

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

2. RESEARCH METHODOLOGY

2.1 Methodology

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

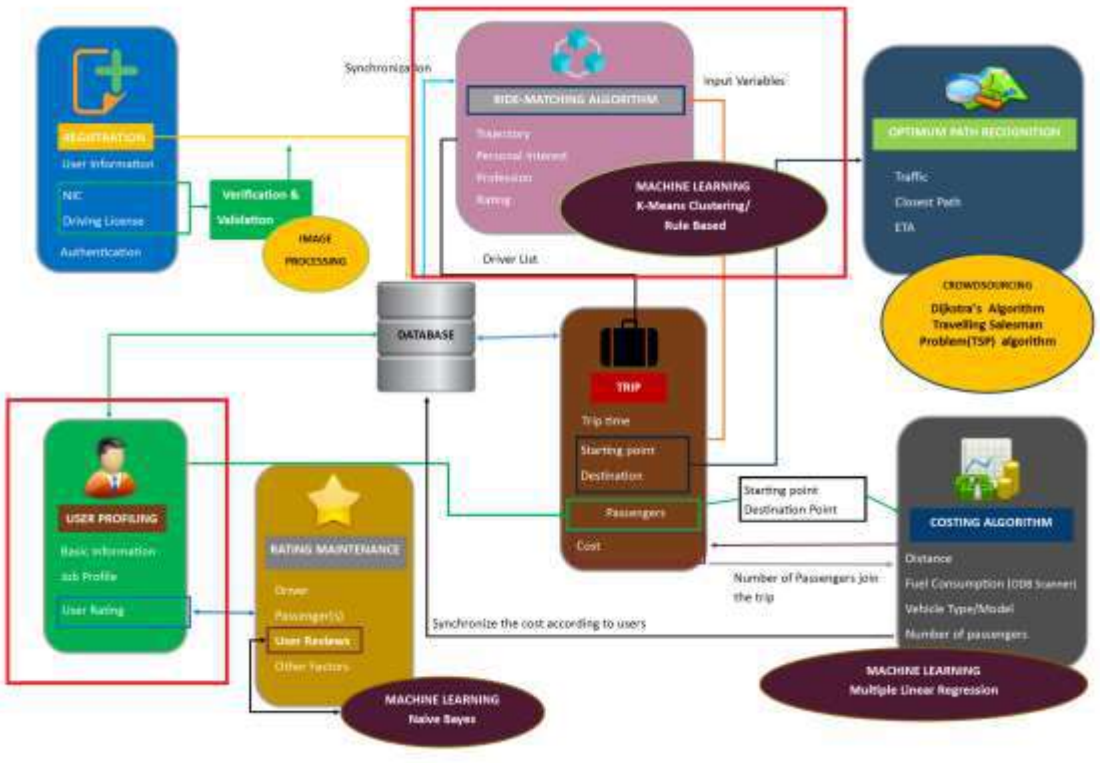

Figure 2.1.1 System Diagram of +Go

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 the 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.

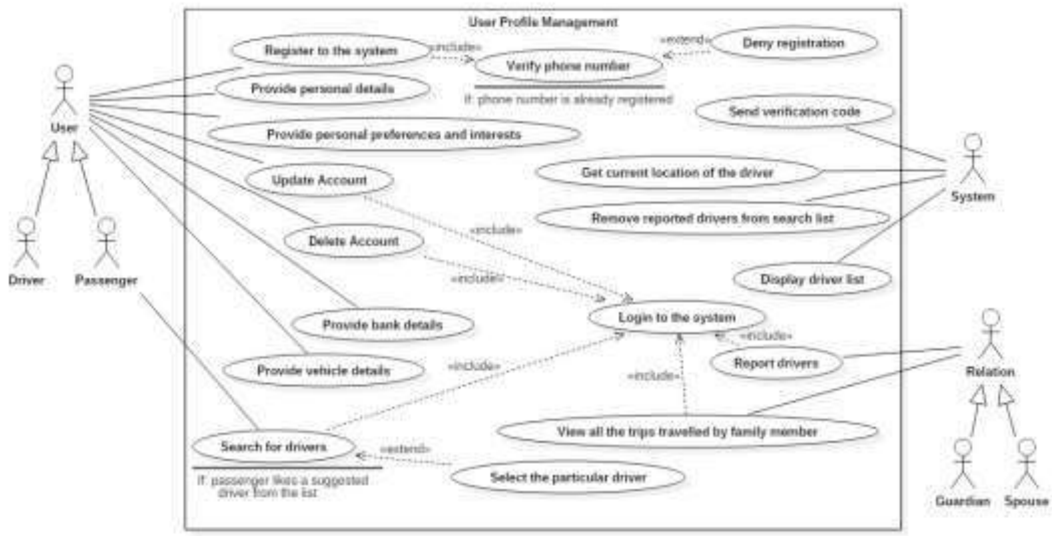

Figure 2.1.2 Use case diagram of UPM

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.

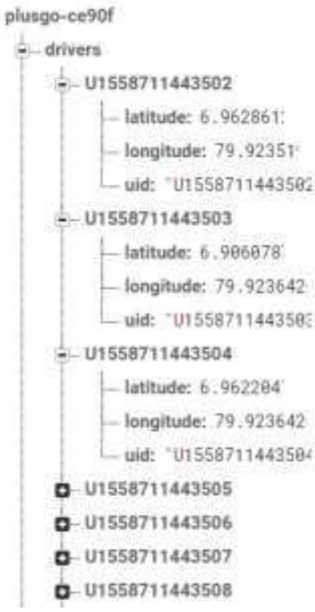

Figure 2.1.2 Polling the location

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.

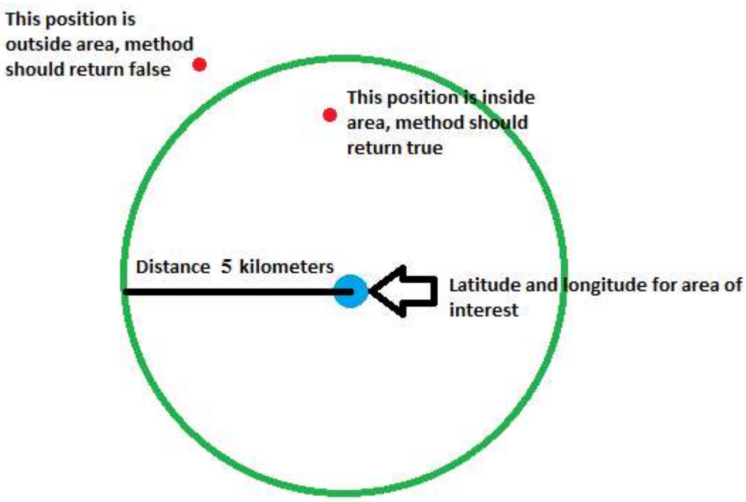

Figure 2.1.3 Distance Comparison

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.

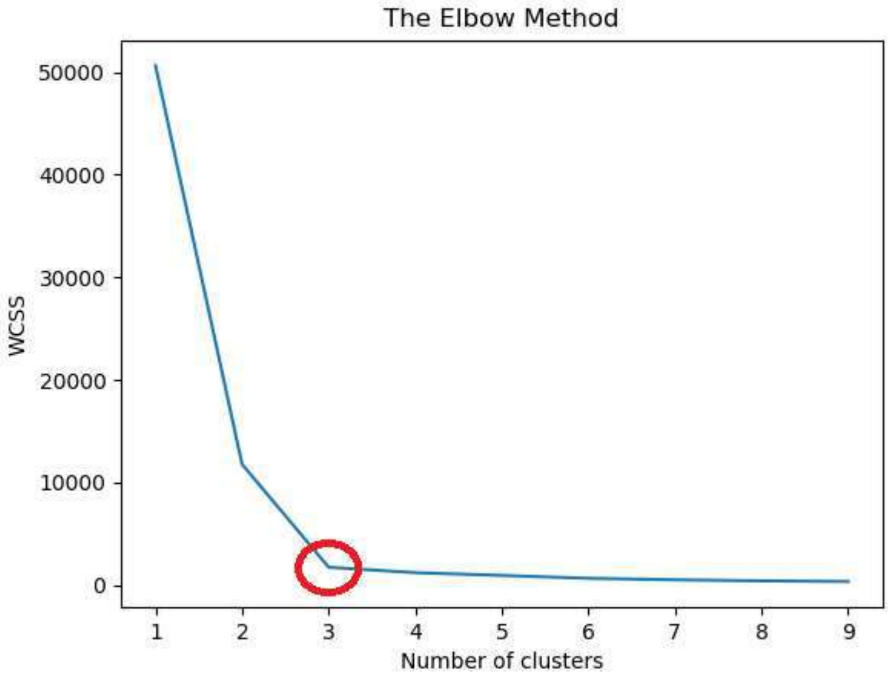

Figure 2.1.4 Identifying the number of clusters

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.

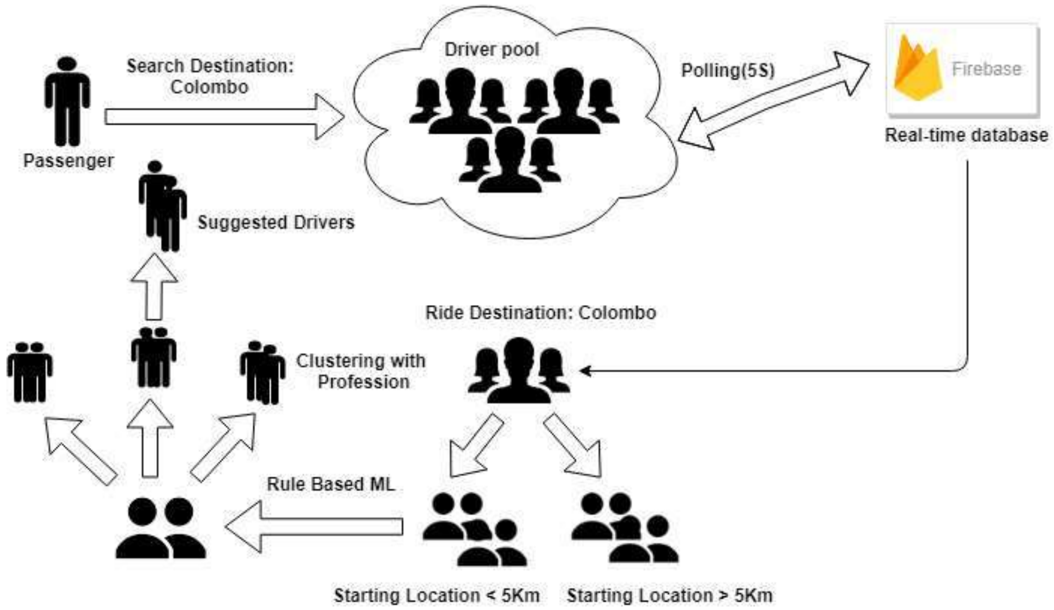

Figure 2.1.5 Flow of the Ride matching algorithm

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 specialty 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.

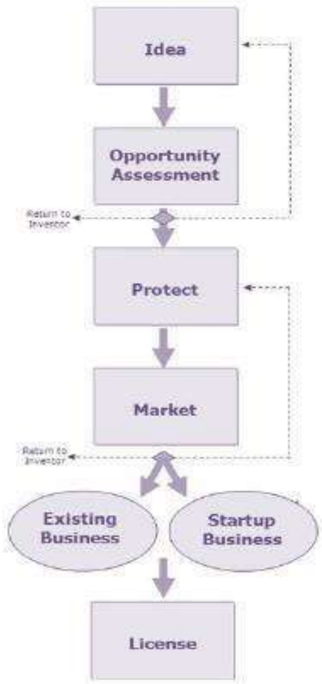

Figure 2.2.1 Commercialization Plan

2.3 Testing & Implementation

2.3.1 Implementation

2.3.1.1 Hardware Interfaces

As this is a mobile platform based application, less amount of hardware is needed. For the application to run, internet connectivity is a must. For the UPM, the camera component and GPS module of the phone is used as the hardware components.

2.3.1.2 Software Interfaces

The primary software interfaces used in UPM component 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.

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

```
F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[140317582640], [108440265434], [774600029093], [130662579047]]

F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[732384554714], [291355602859], [716682232721], [201089799703], [506120731958]]

F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[168378556424], [665344305022], [368037594021], [831723524731], [990759552672]]

F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[946992600171], [597671393064], [570262867319], [627932418484]]

F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[904496347395], [957823937946], [891461215565], [479617555363], [932129745351]]

F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[692418906910], [680553354927], [619218911720], [664640888367]]

F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[945857457263], [594225809519], [463703039941], [978537133753]]

F:\SLIIT\4th Year\Research\Python Work\CodeFlexers-Initial-Implementation\Viraj>python old.py
[[476485588839], [183407800430], [935678955456], [818421009548], [395017668326]]
```

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.

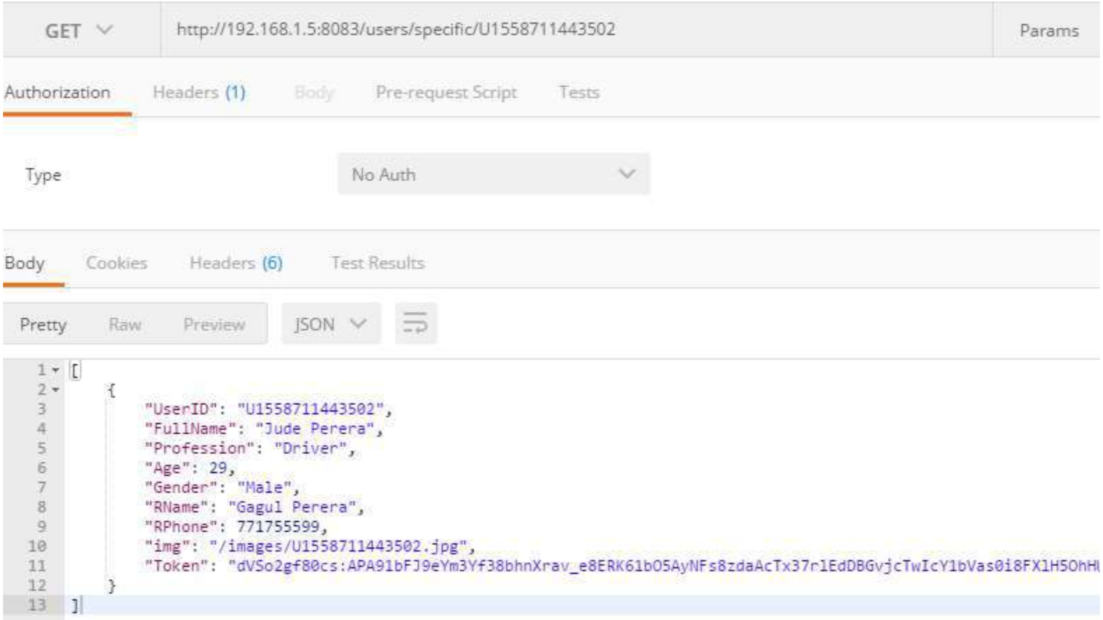

Figure 2.3.2.3 GET request testing

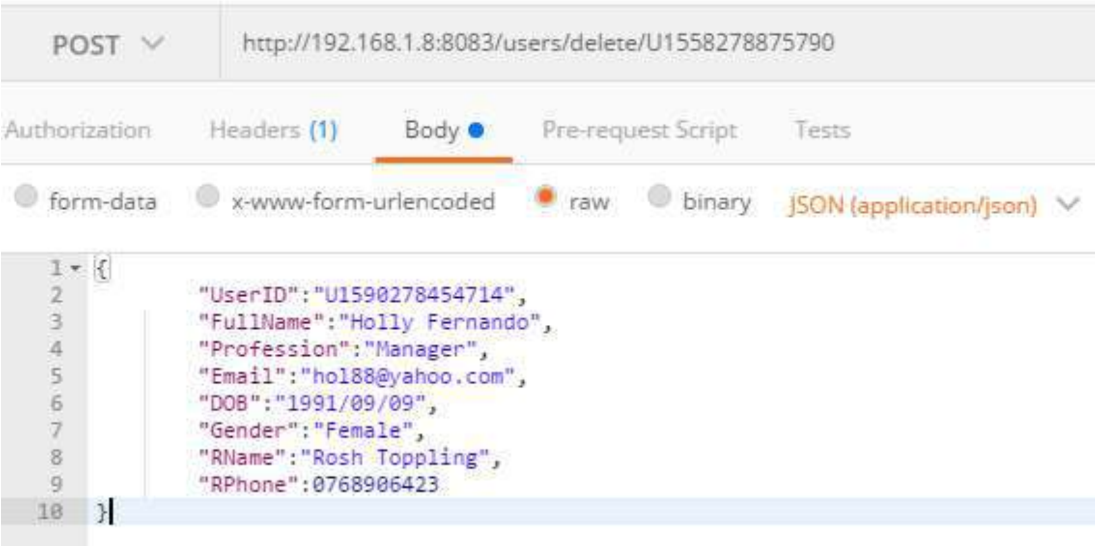

Figure 2.3.2.4 POST request testing

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 traveled 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 traveled to Colombo
Actual Output	List of drivers who traveled 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

3. RESULTS AND DISCUSSION

3.1 Results

3.1.1. 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[9]. 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	U9650469397532	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.

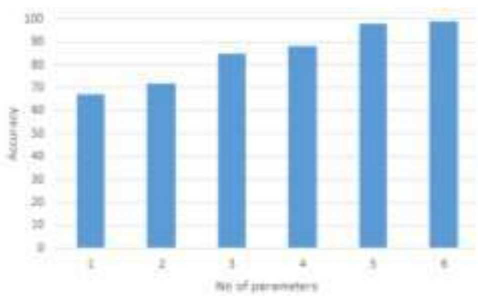

Figure 3.1.1.2 Accuracy vs. Parameter count

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 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.

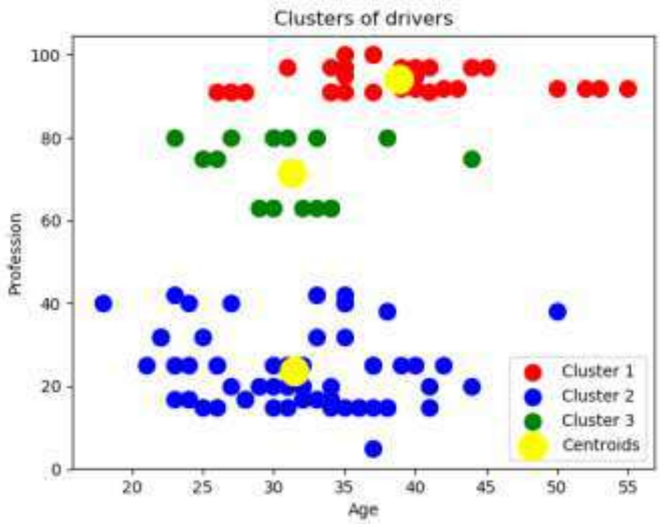

Figure 3.1.1.3 Clustering of Drivers

Aligning with fig. 3.1.1.3(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.

3.1.2 System Interfaces

Figure 3.1.1.1 Login Page

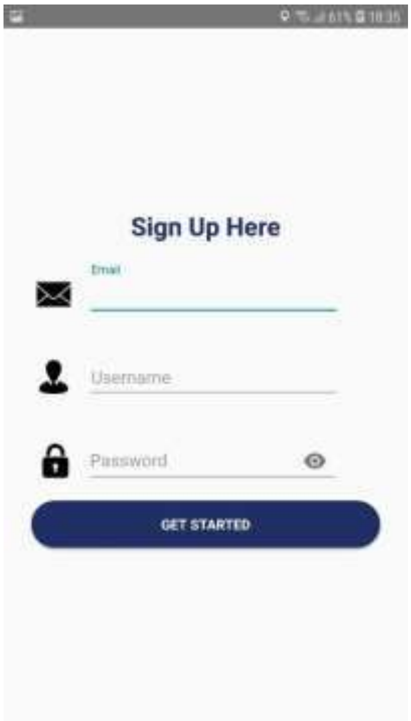

Figure 3.1.1.2 Signup Page

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.

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, traveling cost as well as environmental pollution catering the users' needs.

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 labor. 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.

In the application point of view, we found that the number of parameters included in the algorithm directly affects the accuracy of the results and the K-Means clustering algorithm recorded with 98% accuracy. 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.

3.3 Discussion

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 in 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 the 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[11], and we have done some minor changes to it to suit our purpose.

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[12]. 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. 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 is no other application in the world possesses features of our applications mentioned previously. This application will make a new revolution in the IT industry. In this document, more focus is given to the core functionality of our application. From the UPM, it brings a new mechanism to the world in suggesting the drivers for a passenger. 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 evident that +Go application can bring a new solution to reduce traffic congestion in Sri Lanka.

5. REFERENCES

[1][Online].Available:https://indi.ca/2015/10/colombo-vehicle-statistics-2015/. [Accessed: 02- 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] [Online]. Available:http://www.sundaytimes.lk/190224/business-times/colombos-traffic-nightmare-337578.html. [Accessed: 09- Apr- 2019].

[4] 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

[5] R. Fagin and J. H. Williams, "A Fair Carpool Scheduling Algorithm," in IBM Journal of Research and Development, vol. 27, no. 2, pp. 133-139, March 1983.

[6] [Online]. Available: https://www.researchgate.net/publication/257186810_Real_Real-Time_Carpooling. [Accessed: 05- Jun- 2019].

[7] 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].

[8] 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.

[9] 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.

[10] [online] Available at: https://tradingeconomics.com/sri-lanka/co2-emissions-metric-tons-per-capita-wb-data.html [Accessed 9 Jan. 2019]

[11] 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).

[12] "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].

[13] "Firebase Realtime Database - Techotopia", Techotopia.com, 2019. [Online]. Available: https://www.techotopia.com/index.php/Firebase_Realtime_Database. [Accessed: 09- Feb- 2019].

6. APPENDICES

Appendix – A: Clusters created after the execution of the algorithm

	A	B	C	D	E	F	G	H	I	J	K	L
1	UID	Profession	Rating	Age	Profession_Category	Language_Spoken	Gender_Preference	Smoking	Music_Lover	Motion_Sickness	Like_Quietness	Cluster
2	U1558711443502	Driver	5	32	25	Sinhala	No	Yes	Yes	Yes	Yes	0
3	U1558711443506	Driver	3.6	23	25	Sinhala	No	Yes	Yes	Yes	Yes	0
4	U1558711443507	Clerk	2.6	31	33	Sinhala	No	Yes	Yes	Yes	Yes	0
5	U1558711443508	Security Officer	4.9	23	15	Sinhala	No	Yes	Yes	Yes	Yes	0
6	U1558711443511	Driver	5	29	15	Sinhala	No	Yes	Yes	Yes	Yes	0
7	U1558711443512	Driver	2.2	31	25	Sinhala	No	Yes	Yes	Yes	Yes	0
8	U1558711443502	Driver	5	32	25	Sinhala	No	Yes	Yes	Yes	Yes	0
9	U3059530743447	Network Engineer	2	26	55	Sinhala	No	Yes	Yes	Yes	Yes	1
10	U7055772550429	Project Manager	2	37	68	Sinhala	No	Yes	Yes	Yes	Yes	1
11	U5354268659531	Project Manager	4.6	36	68	Sinhala	No	Yes	Yes	Yes	Yes	1
12	U1192930487003	Project Manager	4.6	31	68	Sinhala	No	Yes	Yes	Yes	Yes	1
13	U3057021975852	Project Manager	4.5	34	68	Sinhala	No	Yes	Yes	Yes	Yes	1
14	U6310173164476	Project Manager	4.3	28	68	Sinhala	No	Yes	Yes	Yes	Yes	1
15	U5285388397741	Database Administrator	3.8	34	60	Sinhala	No	Yes	Yes	Yes	Yes	1
16	U1575461621344	Nurse	2.6	31	80	Sinhala	No	Yes	Yes	Yes	Yes	1
17	U2357757972667	Nurse	3.7	39	80	Sinhala	No	Yes	Yes	Yes	Yes	1
18	U2831639648422	Lecturer	4.5	34	80	Sinhala	No	Yes	Yes	Yes	Yes	1
19	U5999050894832	Lecturer	3.7	29	80	Sinhala	No	Yes	Yes	Yes	Yes	1
20	U4560957424883	Lecturer	3.4	32	80	English	No	Yes	Yes	Yes	Yes	1
21	U8913440965128	Senior Accountant	2.6	26	92	Sinhala	No	Yes	Yes	Yes	Yes	2
22	U528597768384	Senior Accountant	3.7	30	95	Sinhala	No	Yes	Yes	Yes	Yes	2
23	U6201056312849	IT Director	4.5	22	95	Sinhala	No	Yes	Yes	Yes	Yes	2
24	U8884889763492	IT Director	3.7	33	95	Sinhala	No	Yes	Yes	Yes	Yes	2
25	U3490245058599	IT Director	3.4	31	95	Sinhala	No	Yes	Yes	Yes	Yes	2
26	U7489627118484	Doctor	4.6	41	100	Sinhala	No	Yes	Yes	Yes	Yes	2
27	U6736154665359	Doctor	3.8	34	97	Sinhala	No	Yes	Yes	Yes	Yes	2