INTELLIGENT COMPLEMENTARY RIDE SHARING SYSTEM

Ashane Eranda Edirisinghe

(IT16025936)

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

Ashane Eranda Edirisinghe

(IT16025936)

Dissertation submitted in partial fulfillment of the requirements for the 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 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, 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 medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: _________________ Date:___________________

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

Signature: _________________ Date:___________________

(Dr. Janaka Wijekoon)

ABSTRACT

Traffic congestion is a significant concern in the country as the number of vehicles daily entering the urban areas a high; most of them are underutilized as they carry only an individual passenger. To approach as a feasible solution to this issue, an Intelligent Complementary Ride-Sharing System named "+Go", which fills the gaps of existing applications was developed. As it is essential to have proper user authentication to get the attention and trust of the users, the platform uses computer vision methodologies to avoid fake registrations. The platform validates primary documents such as national identity card and license card, which are the documents common to every legitimate citizen in the country. The experience of the user is designed to be improved in the +Go application by maintaining a customized rating arrangement. This feature allows the driver to rate the passenger, and the passenger to rate the driver, vehicle and the co-passengers. Also, a feature of analyzing the reviews given by the users is associated with the platform and has been achieved using sentiment analysis with the help of Naive Bayes classification. The fundamental intent of this report is to demonstrate how the feature of document analysis and profile rating maintenance is developed and achieved in the ride-sharing platform stated above, and how it ultimately reduces traffic congestion in the country by providing a convenient, reliable and secure platform for the users.

Keywords— Ride Sharing, Machine Learning, Sentiment Analysis, Image Processing

ACKNOWLEDGEMENT

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

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

NIC	**National Identity Card**
ICRSS	**Intelligent Complementary Ride Sharing System**
DVPRM	**Document Validation and Profile Rating Maintenance**
API	**Application Programming Interface**
REST	**Representational State Transfer**
XML	**Extensible Markup Language**

1. INTRODUCTION

1.1 Introduction

The statistics collected in 2015 by the department of motor vehicle shows that over 500000 vehicles arrived in Colombo and more than 1.8 million people have arrived in rush hours; majority of them are privately owned vehicles[1]. It has been identified that the main reason for such an amount of vehicles is the office crowd,who do not tend to use the public transportation as it is not in the expected level of comfortability and standard. If single occupants who travel in their private vehicles can share their trip among other occupants who travel to the same destination, it will reduce the numbers of vehicles on the road; results in reducing the traffic congestion and minimize the emission of carbon dioxide (CO_2) to the environment[2]. Also it has been identified that Sri Lanka experienced 471 billion economic loss in the last several years because of the traffic congestion[3].

With the expectation of becoming a solution for this traffic issue, we came up with a concept of ridesharing platform. We have done an initial survey to confirm our hypothesis. From the survey results, we identified that over 82% people up-voted for ridesharing as a good option for Sri Lanka.

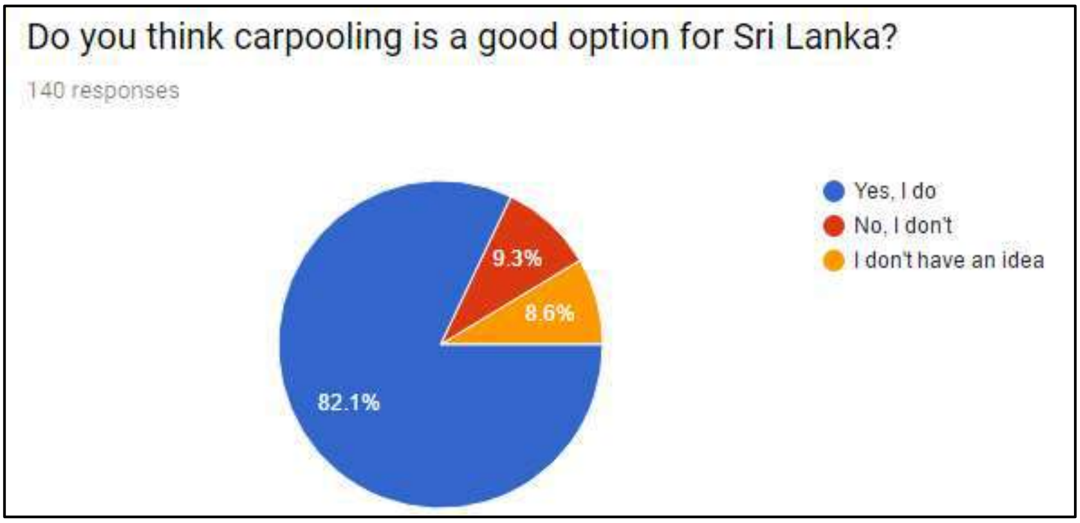


Figure 1.1.1: Suitability of ride-sharing in Sri Lanka

Also we identified that over 72% of the people believe traffic will reduce from this proposed solution and over 61% of people stated that they like to collaborate in the car-pooling platform.

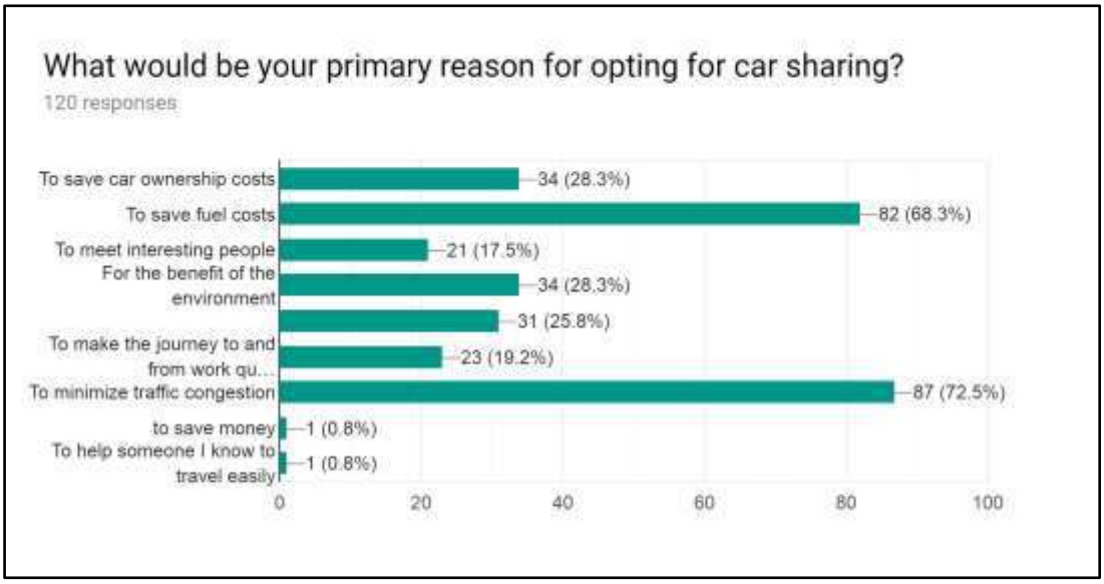


Figure 1.1.2 Reasons for opting Ridesharing

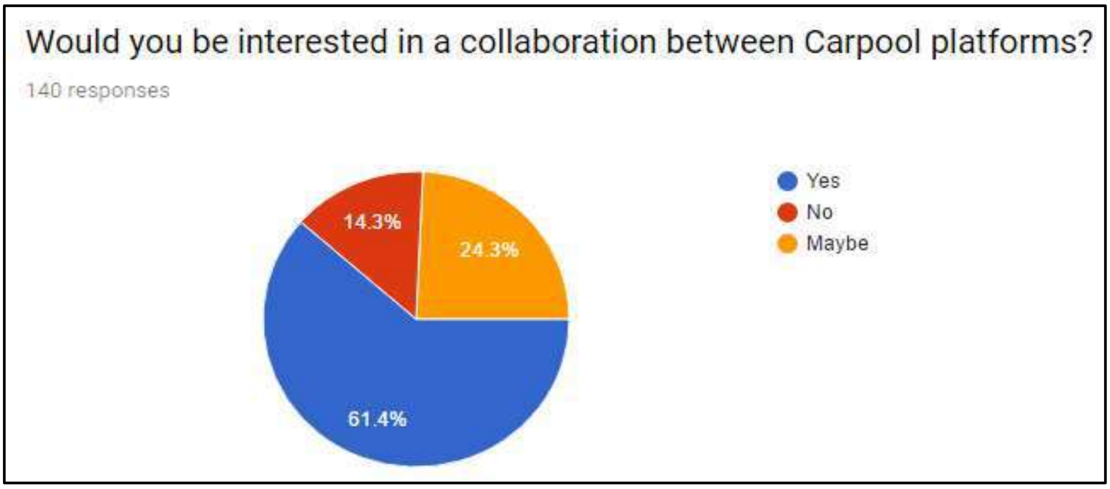


Figure 1.1.3: Interest in Ridesharing

Hence we did research on the concept of ridesharing into the next level as an “Intelligent Complementary Ride-Sharing System (ICRSS)”. ICRSS has been developed as a mechanism to reduce the traffic congestion in urban areas of the country by providing a mobile platform for the users as an effective medium of ride sharing. When it comes to the convenience, security and reliability of the software application, “Document validation and Profile Rating maintenance (DVPRM)” plays a vital role.

This report presents a detailed description about DVPRM with the purpose of describing how the important documents of users (National Identity Card and Driving License) gets validated in the proposed platform and how the profiles of the users maintain their ratings, to affect the behaviour of overall proposed system. Further this report highlights the main features, the feasibility, the importance of document validation and rating in the system and how the requirements are achieved in a proper methodology and acquire results to provide a user friendly product to the stakeholders. The main components under DVPRM in the proposed system consists of two sub components as,

- Validation of user documents at the stage of user registration

  The necessity of document validation is to minimize to fake registrations to the system. Here we mainly focus on two documents,

  - National Identity Card of the user (Both electronic and non-electronic)
  - License card of the user.

- Rating of the user done at the end of each trip

  The purpose of rating maintenance is to make the system more reliable to the users. At the end of each trip, the driver will get the chance to rate or review on the passengers, and vice versa.

Our product is fully software-based application, which is more specifically a mobile application and will be used as a platform for ride sharing by different professionals on a daily basis.

1.2 Background Literature

Identification documents are one of the main sources for verifying the identity of citizens. To proceed with the verification of National Identity Card (NIC) and license card, we used a smartphone to capture the picture, which is sent to a cloud server. Then the image is processed by server and results are sent back to the smartphone. As mentioned by Valiente, Sadaike, Gutiérrez, Soriano, Bressan, and Ruggiero (2016), by using this methodology, a cloud could be used for intensive processing by the devices, which has low computational power [4].

The process could be done in the following steps

I. Load Image

As the way, Parwar, Goverdhan, Gajbhiye, Deshbhratar, Zamare, and Lohe propose, the user is allowed to capture an image from the camera. If the Image is blurred, the user can take a new image again [5].

II. Crop Image

Eliminate the unnecessary parts in the image

III. Process Image

Image processing can be used to observe objects, which are not visible by the naked eye, sharpening or restoring images into better quality, measuring patterns in various objects, or distinguishing objects in an image[5]. Ohlsson (2016) states that the common way to start the preprocessing of an image is by converting it as a gray-scale image before continued preprocessing [6]. In the way, Chakraborty and Mallik (2013) explain, the color image has to be converted to grayscale for more accurate recognition.Next, the grayscale image is converted to binary using an Adaptive threshold. The adaptive threshold is necessary to convert the grayscale image to a binary image because it is difficult to convert some images to binary by applying a constant threshold and in order to simplify the extraction process[6],[7]. As mentioned in literature, this process plays a significant role in extracting text from the image, as RGB images contain noise at most times, and are not perfect in identifying text and non-text objects of the images[8]. According to Mordvintsev and Abid (2017)[9], image processing could be easily performed with the help of OpenCV. Further, this provides the capability for face detection using Haar Cascades. With reference to Clark (2018)[10], 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.

IV. Extract Image

The process by which image text converted into plain text is Text Extraction. Text Extraction is quite helpful in information retrieving, editing, documenting, searching, etc. Yet the need for a tool for text extraction has always been there [6],[7]. OCR is a technology for converting text on images into data strings. The strings can be used for many things but some examples are to digitize old documents, translate into other languages or to test and verify text positions[5]. This has also been widely used in various fields such as cheque processing, digital libraries, recognizing text from natural landscape, understanding handwritten text, etc.[10]. According to Chakraborty and Mallik (2013), the processed image can be sent for recognition using Tesseract recognition engine which was developed at HP between 1984-1994, then released for open source in late 2005, and currently owned by Google and is considered to be the best highly portable open source OCR engine currently available[4, 7, 11, 12]. It has been indicated by Ohlsson (2016) that Tesseract supports UTF 8, and has the ability to recognize 100+ languages, and the support for more languages is continuously increasing. The engine is trainable, meaning that a new language or font, which is not normally supported, can be trained and recognized. Tesseract is identified as the most accurate open source engine found on the market[5]. According to text extraction done from images of vehicle number plates by C. Patel, A. Patel and D. Patel (2012)[13], Tesseract has proven the accuracy of 61% and 70% with the color and grayscale images respectively. It proves the fact that Tesseract performs better in grayscale images as compared to color images. Further, it has mentioned that in some color images with text extraction accuracy of 100% or near to 100%, and when converted to grayscale, remained with the same extraction accuracy. In addition, it was observed that the processing time of character extraction from grayscale images is reduced by 10% to 50%. So it implies the fact that Tesseract works fast with better text extraction accuracy when it comes to grayscale images.

Therefore, by considering all the facts regarding text extraction from images, we proceeded with Tesseract for text extraction and OpenCV for image processing functions.

To analyze the reviews given by the users, the sentiment analysis approach was used. Sentiment analysis aims to determine the polarity of emotions like happiness, sorrow, grief, hatred, anger and affection and opinions from the text, reviews, posts which are available online on these platforms[14],[15]. 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[15],[16]. Llombart and Romero (2017) states that in a real scenario, this independence can hardly be found. As a precaution, they have used Multinomial Naive Bayes, which was provided by SciKit-Learn. This leads to model the same probability but with multinomial distribution[16]. 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[14].To avoid the problem of memorizing data and performing poor with new data; which is mostly happened with machine learning algorithms, Llombart and Romero (2017) propose to work with the test-driven methodology as follows.

- Train (60%): Used in the learning process of the machine learning algorithm
- Test (20%): To verify the algorithm is overfitting or not.
- Validation (20%): To evaluate the accuracy of the machine-learning algorithm[15]

Ramya and Rao (2018) suggest using 80% for training and 20% for testing. Further, this paper comes to the conclusion that the Naïve Bayes algorithm performs better in text analysis when it is compared with algorithms such as Support Vector Machine (SVM) and Multinomial Logistic Regression[17]. The paper of Dey, Chakraborty, Biswas, Bose, and Tiwari (2016) elaborately compares overall accuracy, precision as well as recall values of K-Nearest Neighbour(K-NN) and Naïve Bayes' and it was obvious that when it comes to movie reviews Naïve Bayes' gave far better results than K-NN, but with hotel reviews, both produced lesser nearly same accuracies [14]. According to the comparison between Naive Bayes ,K-Nearest Neighbour and Random Forest Algorithms done by Baid, Gupta, and Chaplot (2017), they suggest 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[15]. With respect to the opinions of Vidushi and Sodhi (2017) and other related literature, it concludes that the results are found to be satisfactory and when the comparative analysis is done, Naïve Bayes algorithm outperforms KNN algorithm[18],[19]. By considering all the facts in literature, we used Naive Bayes in analyzing the reviews given by the users in our ride-sharing platform.

1.3 Research Gap

Ride sharing applications are common in the market. But with 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[20],[21,[22],[23].

Table 2.1.1: Feature Comparison Table

Features	**Uber**	**UDIO**	**Carpooling.lk**	**RideShare.lk**	**+GO**
Validating the user by processing and comparing the images of both NIC and license in real time *This feature reduces the fake registrations, and eliminates the need of manual validations in*	✗	✗	✗	✗	✓

the system					
Analyze the reviews given by users based on their severity and categorizing them *The system 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*	✗	✗	✗	✗	✓
Allowing the passengers to rate the driver, vehicle and co-passengers separately at the end of trip. *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.*	✗	✗	✗	✗	✓

1.4 Research Problem

With reference to the articles, surveys, and police reports, there is a gradual increase of vehicles that enter to Colombo during rush hours[24]. Moreover, with the increase in traffic congestion, the number of accidents too increases. From the surveys conducted by the Central Bank, they identified that low occupancy vehicles like cars have been in high number and resulted in traffic congestion; wasting financial resources, moreover polluting the environment badly. It has also stated that there is Rs.500 million loss due to the daily traffic congestion[24]. Due to the heavy traffic during rush hours, there is a huge loss of time wasted on the road. So it becomes obvious that traffic is a major concern in urban areas, especially during office hours.

The answer is obvious if we can find a way of reducing the number of vehicles enter the city; we can reduce the traffic congestion up to a certain extent. So we thought of introducing a ride-sharing app, which could possibly become a **solution to the traffic congestion.** The basic idea behind that was to combine similar group of professionals who travel the same route to work by their private vehicles, and provide a common comfortable platform as an alternative with the help of ride sharing concept. Hence, when one vehicle carries several people together, it will reduce the number of vehicles entering the city.

1.5 Research Objectives

1.5.1 Main objective

The main objective of this research is to develop an effective solution to minimize the traffic congestion during office hours in urban areas. Because of that, we thought of introducing a ride-sharing app for the working people (office staff) which will help to minimize the traffic congestion. Ride sharing service has become a convenient and felicitous transportation system to everyone everywhere at any time. Apart from reducing the traffic congestion, there are some other tremendous positive impact on our proposed solution. Some of them are building the network among professionals with similar social status, which will help to reduce stress and improve the productivity while travelling as a passenger, minimize the environment pollution and fuel consumption cost and help to save the car ownership cost and ensure the security of passengers too.

1.5.2 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.

1.5.3 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.

2. RESEARCH METHODOLOGY

2.1 Methodology

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.

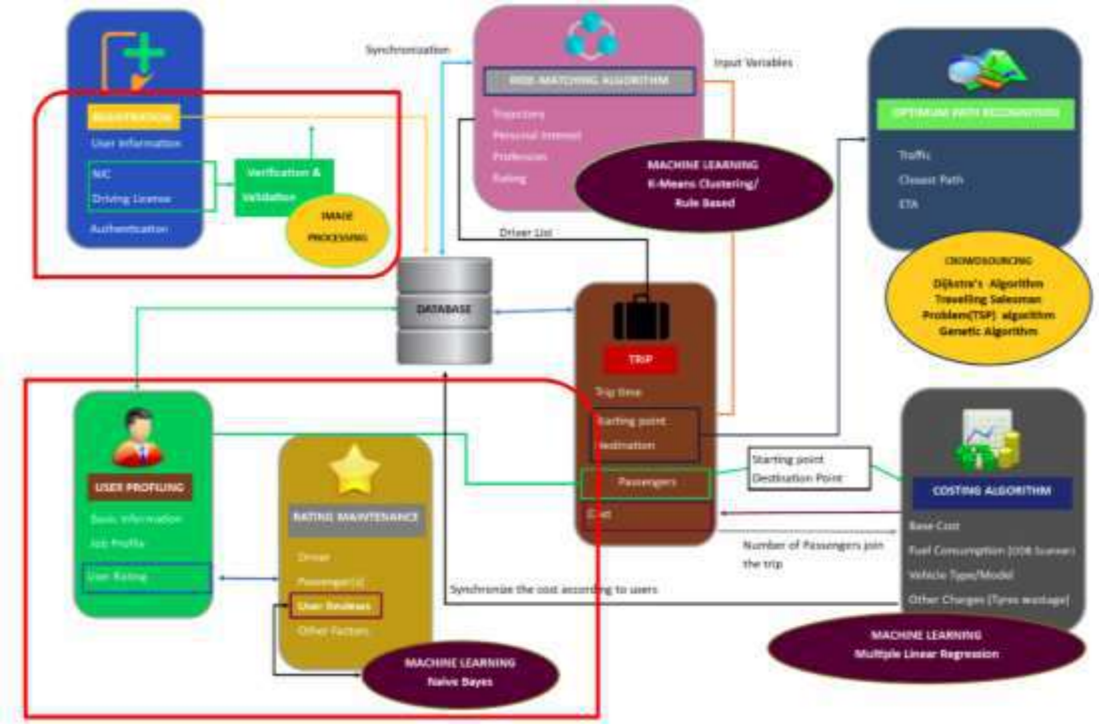


Figure 2.1.1: System Diagram of +Go

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 too 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.

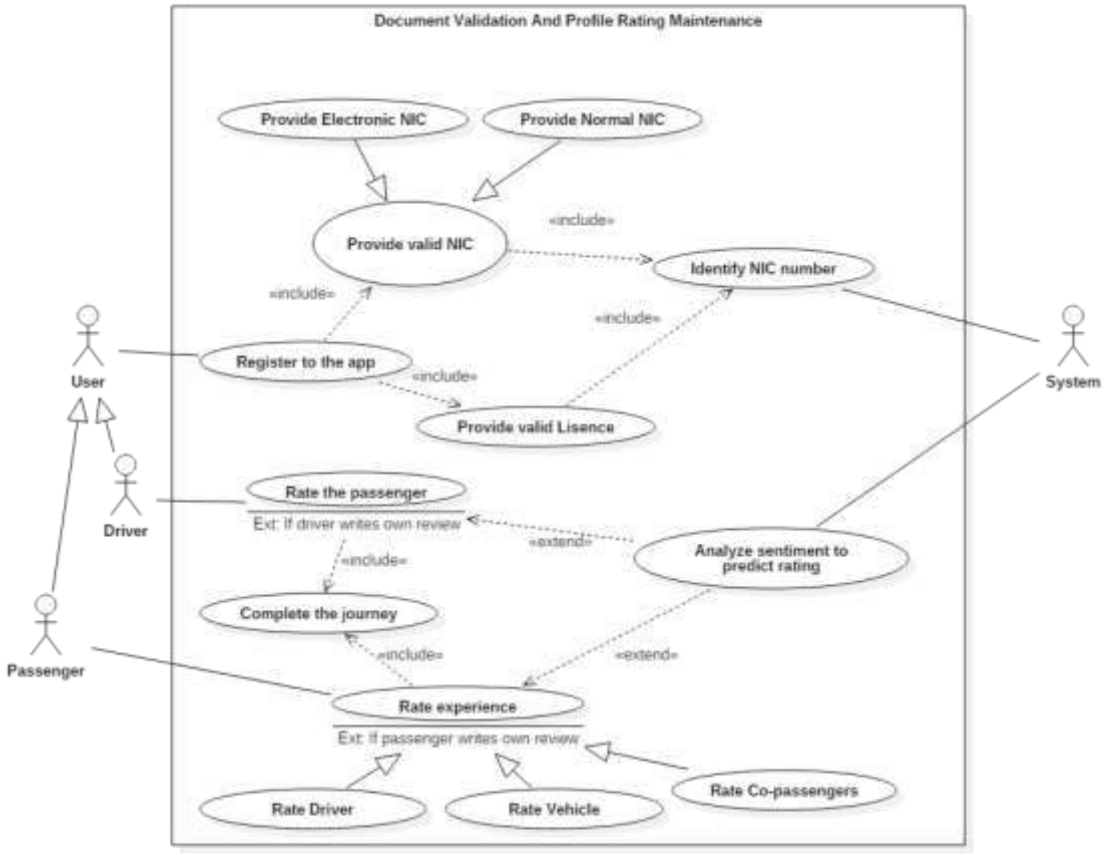


Figure 2.1.2: Use Case Diagram of DVPRM

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

2.1.1 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[25].

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.

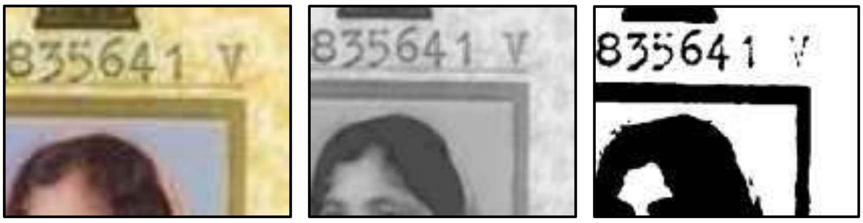


Figure 2.1.1.1: Raw, grayscale and binary image parts of non-electronic NIC respectively[26]

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.

2.1.2 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.

$$\underset{\text{Posterior Probability}}{P(c\mid x)}=\frac{\overset{\text{Likelihood}}{P(x\mid c)}\,\overset{\text{Class Prior Probability}}{P(c)}}{\underset{\text{Predictor Prior Probability}}{P(x)}}$$

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

Figure 2.1.2 : Concept of Naive Bayes algorithm used [27]

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.

2.2 Commercialization Aspects of the Product

Main aim is to highlight the business value of the ICRSS to the system users by focusing on reducing the number of vehicles entering urban areas of the country ; resulting the reduction traffic congestion and wastage of valuable time of people on the road. Furthermore, this saved time of people can be useful in the development of the country or else for the users to spend with their loved ones. Also there is an indirect impact heavily on the environment, as the amount of CO_2 being added to the environment is reduced.

In the business point of view, it is important to market the product among the people. For that we should have a comprehensive plan addressing every part of the commercialization. Our product behaves unique from other ride sharing applications as we have only focused on the professionals.

As the initial phase +Go application covers only the Colombo area and the next phases are planned to be implemented across the whole country. Our application 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.

Before releasing it to the market officially, we have 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 will be releasing to the market officially. Currently, our application can only be used by Android users and if the iOS market urges for product, we’ll be developing for iOS as well. Our product will be mainly promoted via the social media platforms as well as advertisements.

2.3 Testing and Implementation

2.3.1 Implementation

In the phase of implementation the requirements are identified as follows.

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 and also will need to allow the camera to capture images using the mobile phone

2) Software interfaces

The main software interfaces used in DVPRM component are,

Android Studio - For development of mobile application
Genymotion Emulator (Preferred) - Emulating purposes
MySQL - As the database in the server
Python - To implement the backend algorithms
Express js - For development of web api

3) Communication interfaces

Internet connection provided by modem will be used for the communication between the mobile application and the web server.

4) Memory constraints

The android mobile application is required,

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

2.3.2 Testing

For any kind of application, testing plays a major 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 important 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, 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.

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 2.3.2.1 : Test Case 01

Test Case ID	TC01
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	1. Choose to enter non electronic 2. Select Upload image button 3. Select image of NIC from gallery or capture it from the camera 4. Click button with upload icon 5. 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 2.3.2.2 : Test Case 02

Test Case ID	TC02
Test Case Description	Licence Card Verification
Pre-Condition	User has been successful in NIC verification
Test Procedure	1. Select the upload image button 2. Upload image from gallery or capture image 3. Click button with upload icon 4. 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 2.3.2.3 : Test Case 03

Test Case ID	TC03
Test Case Description	Rating given as a passenger
Pre-Condition	The user has completed the trip session as a passenger
Test Procedure	1. Provide a rating below 5 2. Select vehicle out of the three categories 3. Click other 4. Write a review in the text box 5. 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 2.3.2.4 : Test Case 04

Test Case ID	TC04
Test Case Description	Rating as a driver
Pre-Condition	The user has completed the trip session as a driver
Test Procedure	1. Provide a rating below 5 2. 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

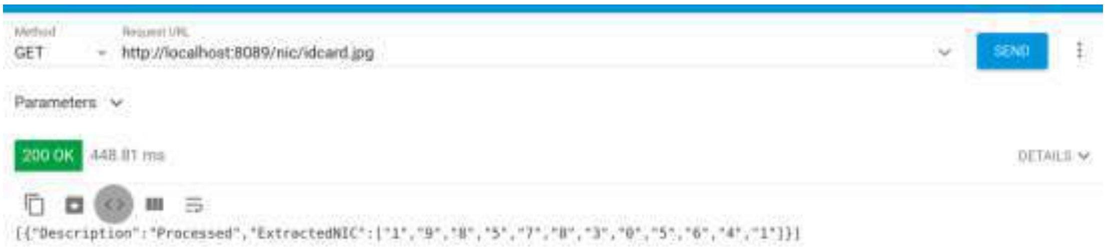
3. RESULTS AND DISCUSSION

3.1 Results

3.1.1 Test results

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 [26] can be identified as below.



Method: GET

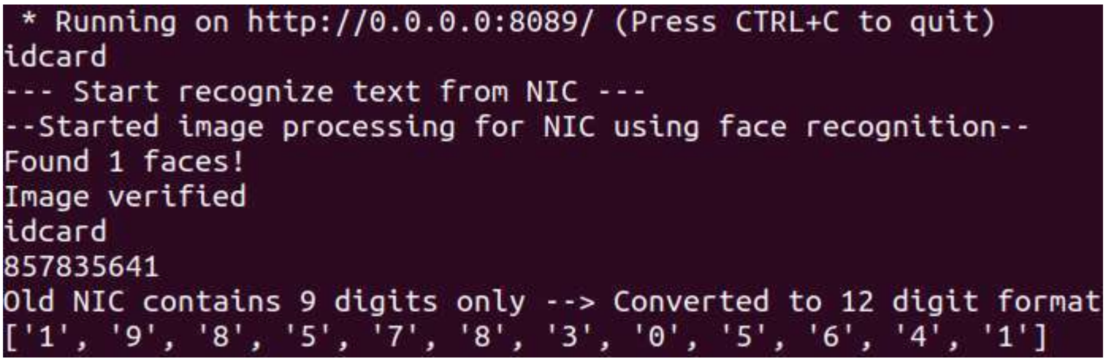
Request URL: http://localhost:8089/nic/idcard.jpg

200 OK 448.81 ms

```
[{"Description":"Processed","ExtractedNIC":["1","9","8","5","7","8","3","0","5","6","4","1"]}]
```

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.

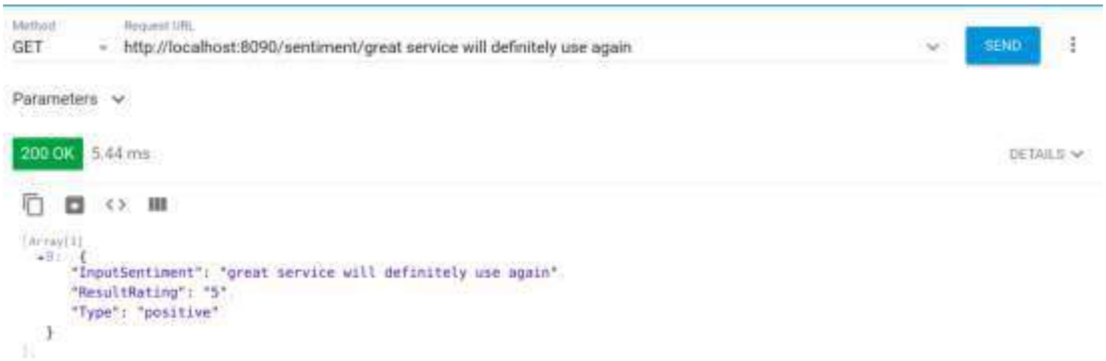


```
 * Running on http://0.0.0.0:8089/ (Press CTRL+C to quit)
idcard
--- Start recognize text from NIC ---
--Started image processing for NIC using face recognition--
Found 1 faces!
Image verified
idcard
857835641
Old NIC contains 9 digits only --> Converted to 12 digit format
['1', '9', '8', '5', '7',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

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



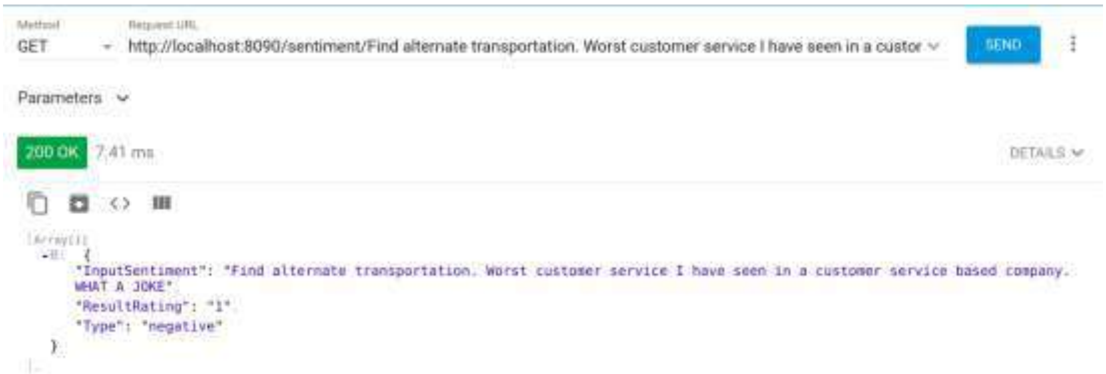
Method: GET

Request URL: http://localhost:8090/sentiment/great service will definitely use again

200 OK 5.44 ms

```
[Array[1]
  -0: {
      "InputSentiment": "great service will definitely use again"
      "ResultRating": "5"
      "Type": "positive"
    }
]
```

Figure 3.1.1.3 : Testing a positive sentiment

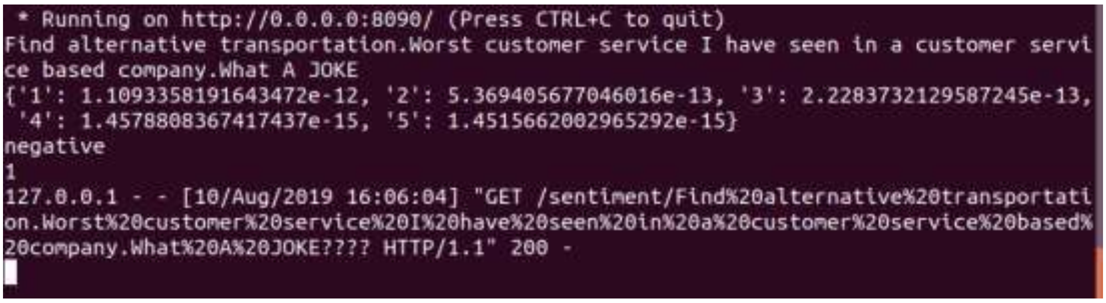


Method
GET
Request URL
http://localhost:8090/sentiment/Find alternate transportation. Worst customer service I have seen in a custor
SEND
Parameters
200 OK
7.41 ms
DETAILS

```
[Array[1]
-0: {
    "InputSentiment": "Find alternate transportation. Worst customer service I have seen in a customer service based company. WHAT A JOKE"
    "ResultRating": "1"
    "Type": "negative"
  }
]
```

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.

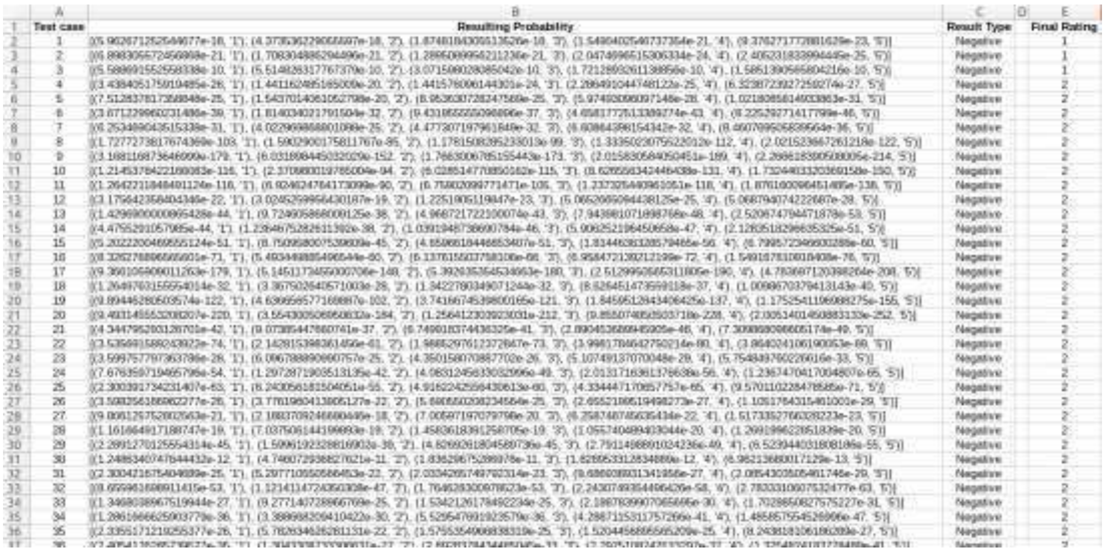


```
 * Running on http://0.0.0.0:8090/ (Press CTRL+C to quit)
Find alternative transportation.Worst customer service I have seen in a customer servi
ce based company.What A JOKE
{'1': 1.1093358191643472e-12, '2': 5.369405677046016e-13, '3': 2.2283732129587245e-13,
 '4': 1.4578808367417437e-15, '5': 1.4515662002965292e-15}
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 -
```

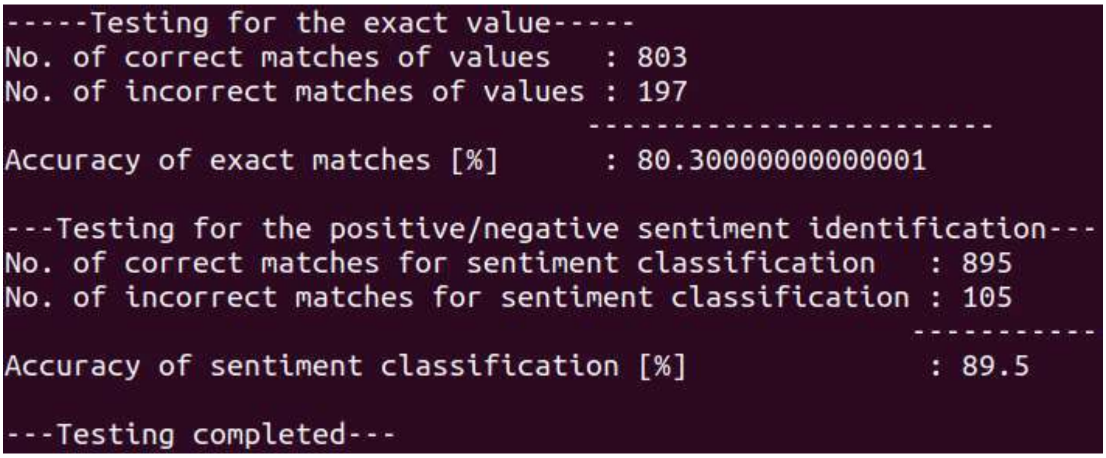
Figure 3.1.1.5 : Logs to identify the sentiment analysis function

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



Test case	Resulting Probability	Result Type	Final Rating
1	[illegible]	Negative	1
2	[illegible]	Negative	1
3	[illegible]	Negative	1
4	[illegible]	Negative	2
5	[illegible]	Negative	2
6	[illegible]	Negative	2
7	[illegible]	Negative	2
8	[illegible]	Negative	2
9	[illegible]	Negative	2
10	[illegible]	Negative	2
11	[illegible]	Negative	2
12	[illegible]	Negative	2
13	[illegible]	Negative	2
14	[illegible]	Negative	2
15	[illegible]	Negative	2
16	[illegible]	Negative	2
17	[illegible]	Negative	2
18	[illegible]	Negative	2
19	[illegible]	Negative	2
20	[illegible]	Negative	2
21	[illegible]	Negative	2
22	[illegible]	Negative	2
23	[illegible]	Negative	2
24	[illegible]	Negative	2
25	[illegible]	Negative	2
26	[illegible]	Negative	2
27	[illegible]	Negative	2
28	[illegible]	Negative	2
29	[illegible]	Negative	2
30	[illegible]	Negative	2
31	[illegible]	Negative	2
32	[illegible]	Negative	2
33	[illegible]	Negative	2
34	[illegible]	Negative	2
35	[illegible]	Negative	2
36	[illegible]	Negative	2



```
-----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---
```

Figure 3.1.1.7 : Test results of the sentiment analysis algorithm

3.1.2 User interfaces

In order to release the version 1 of the application, we have designed mobile interfaces to cater the functionalities. We have designed the mobile interfaces related to DVPRM component. User Interfaces relevant to document validation process are as follows.

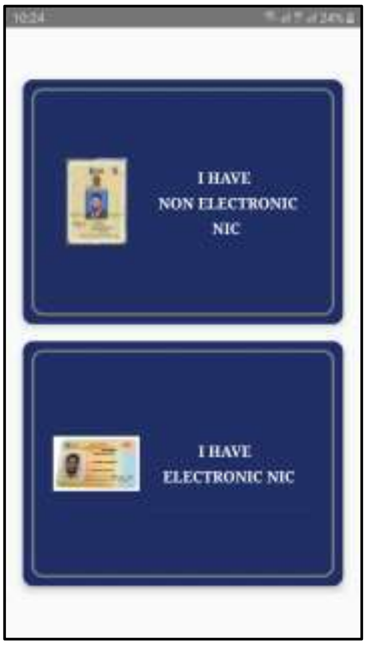


Figure 3.1.2.1: NIC Type selection

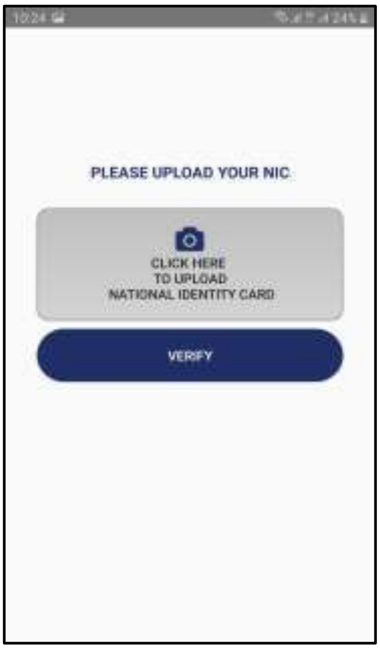


Figure 3.1.2.2: NIC upload

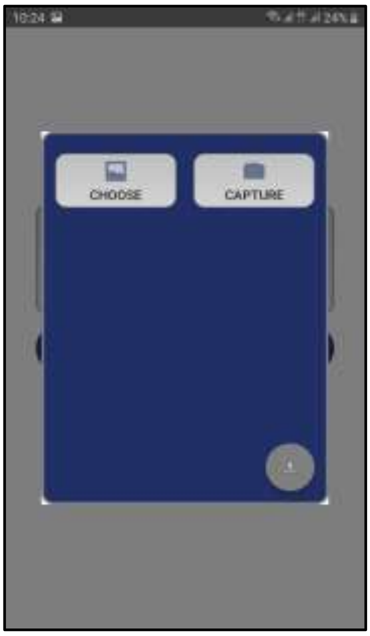
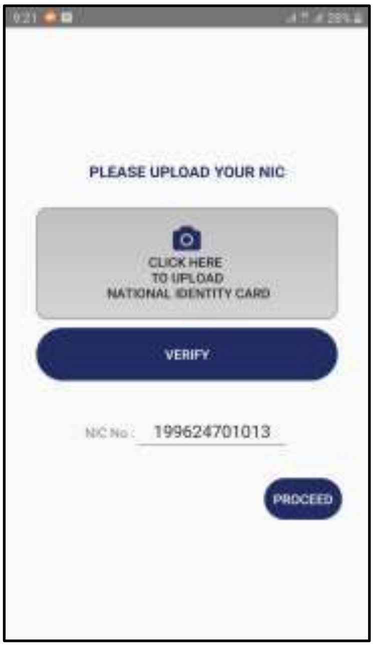


Figure 3.1.2.3: Capture or upload


PLEASE UPLOAD YOUR NIC

NIC No : 199624701013

Figure 3.1.2.4: NIC extraction

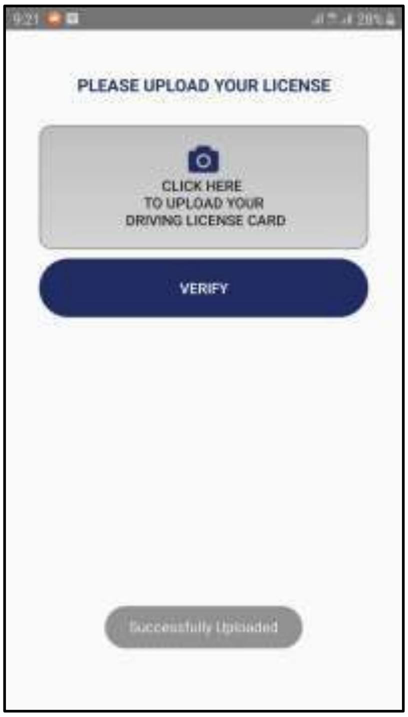
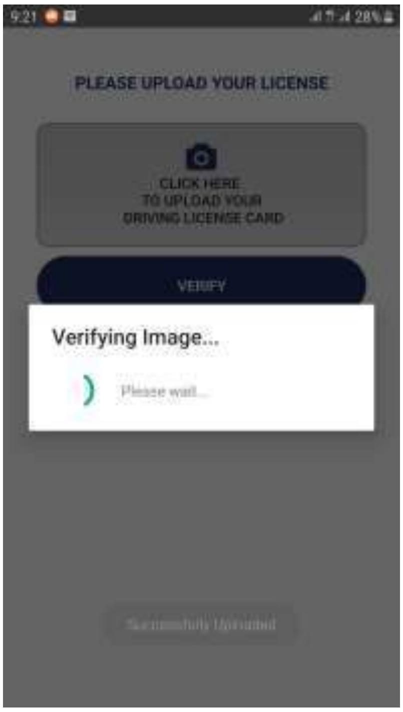


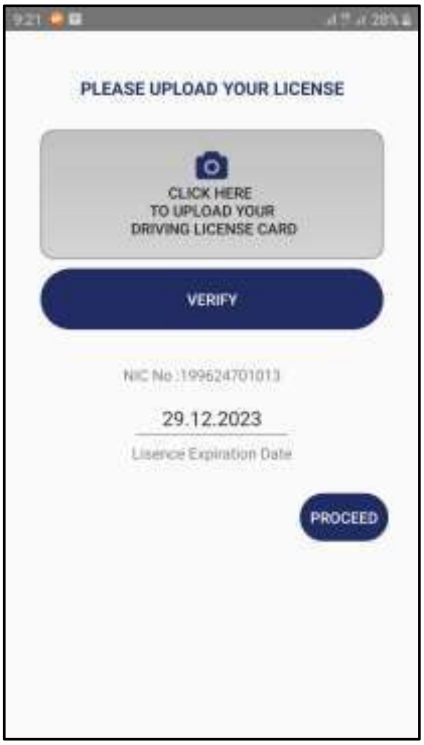
Figure 3.1.2.5 : Successful license upload



Verifying Image...

Please wait...

Figure 3.1.2.6 : License verification



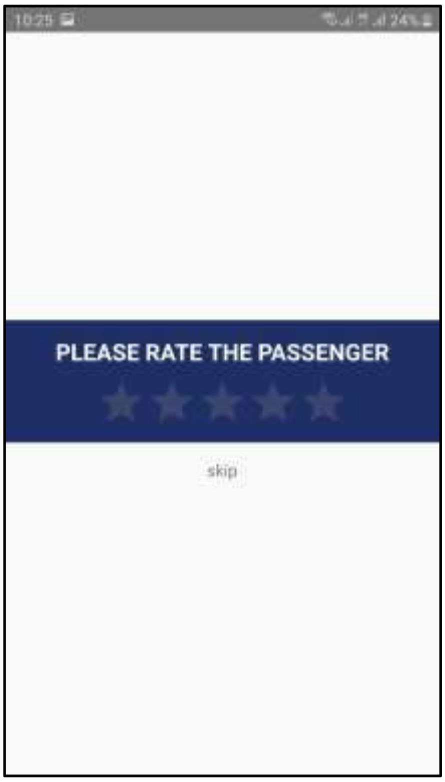
NIC No :199624701013

29.12.2023

Lisence Expiration Date

Figure 3.1.2.7 : License Extraction

User Interfaces relevant to the driver in profile rating maintenance.



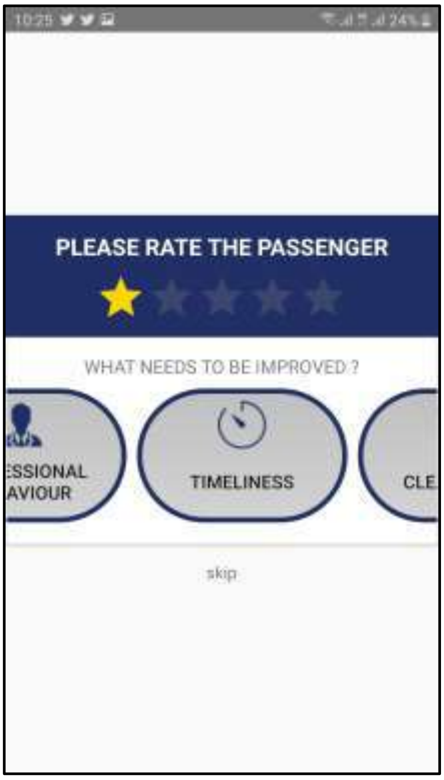


Figure 3.1.2.8 : Rate as a driver

Figure 3.1.2.9 : Keyword selection

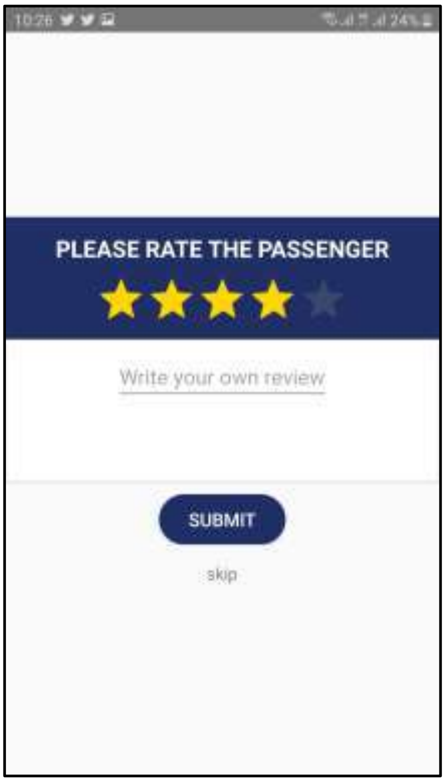


Figure 3.1.2.10 : Review writing

User Interfaces relevant to the passenger in profile rating maintenance.

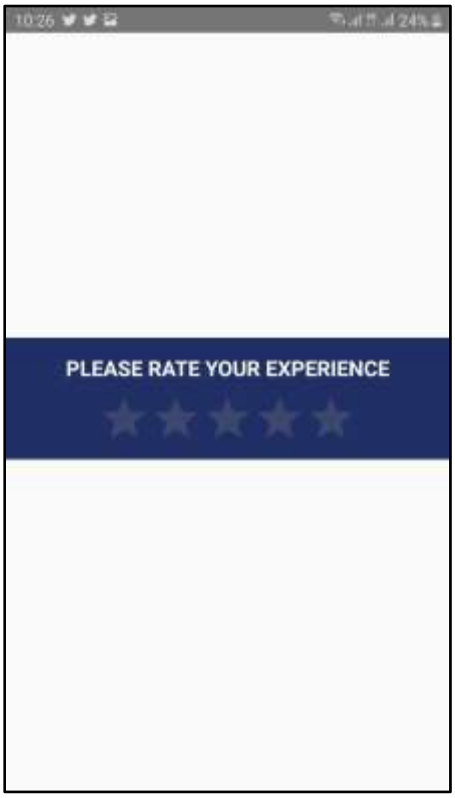
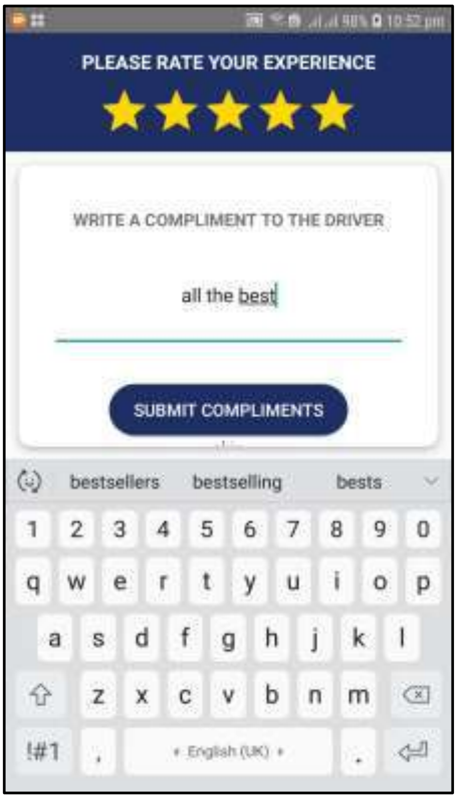


Figure 3.1.2.11 : Rate as a passenger



all the best

Figure 3.1.2.12 : All star rating

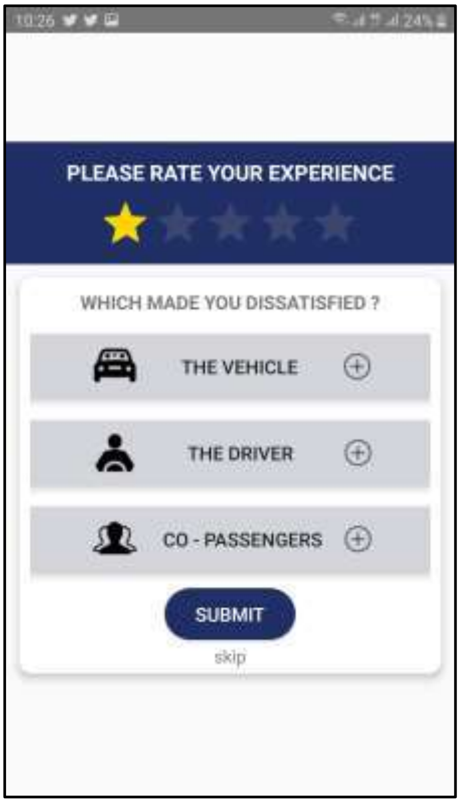


Figure 3.1.2.13 : Low rating

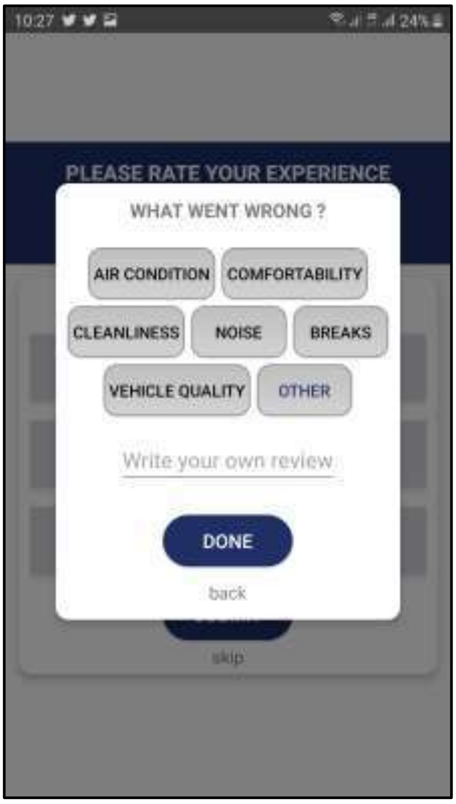
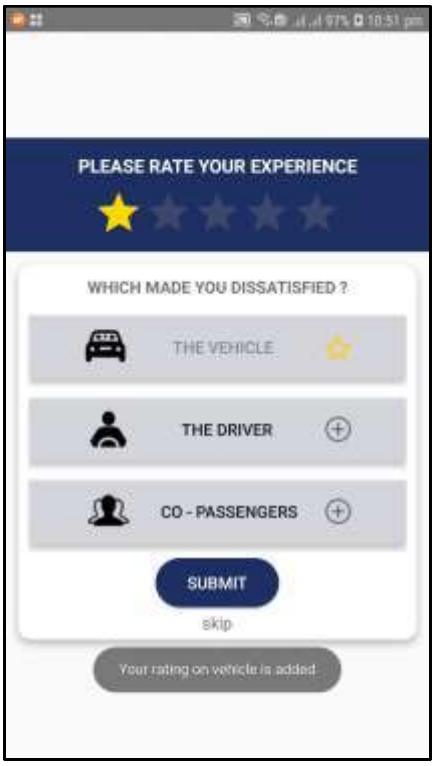


Figure 3.1.2.14 : Keywords to rate vehicle


PLEASE RATE YOUR EXPERIENCE

WHICH MADE YOU DISSATISFIED ?

Your rating on vehicle is added

Figure 3.1.2.15 : After rating vehicle

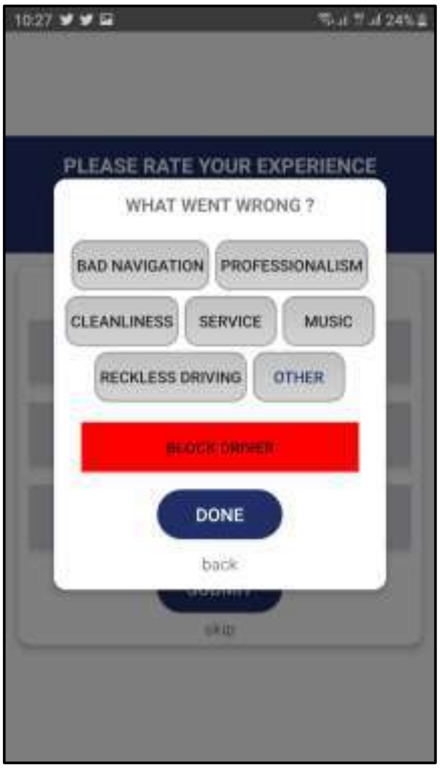

WHAT WENT WRONG ?

Figure 3.1.2.16 : Keywords to rate driver

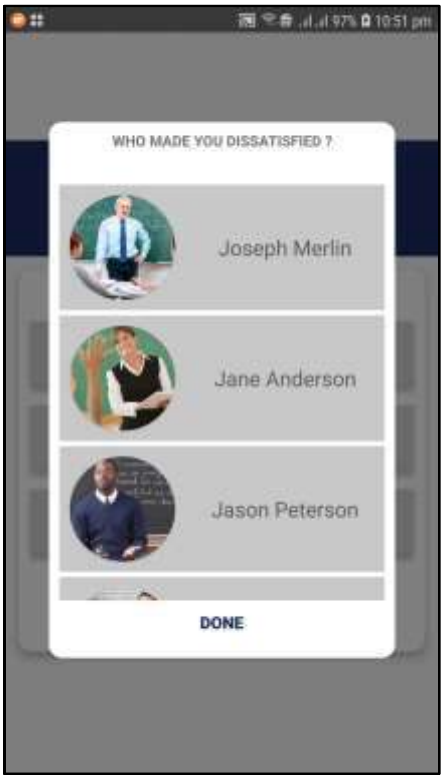

WHO MADE YOU DISSATISFIED ?

Figure 3.1.2.17 : Select co-passenger to be rated

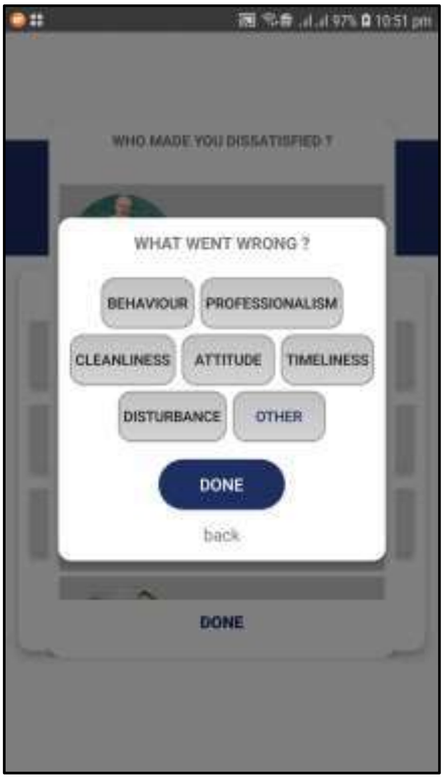

WHAT WENT WRONG ?

Figure 3.1.2.18 : Keywords to rate co-passenger

3.2 Research Findings

The proposed Intelligent Complementary Ride-Sharing System consists of many features and benefits to users. Throughout the research life cycle, several milestones have been passed. the most important sections related to DVPRM component has
been discussed throughout this document
With this section, the effort is to manifest the findings from the study we have
done. Initially we were able to identify that the high number of private vehicles on
the road is the major cause of traffic congestion. With the desire of finding a proper mechanism to this problem, we did a survey; confirmed the need of a reliable and a proper ride sharing platform. We also identified that many people are ready to collaborate with ridesharing.
Therefore we came up with a solution for this need of proper platform in ride-sharing; which would maintain proper comfortability among the professionals and ultimately result in reduction of traffic congestion in the country. We included several features into the application and we identified that those features give additional value to the proposed system. Out of those features , 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.
By considering the literature related to the field of study and also testing them by implementing, we were able to note that python does a wonderful job in providing support for the development of complex logics and algorithms. We also managed to develop the platform with the latest stable versions of latest technologies as it was identified that it causes the platform to extend easily in future.

3.3 Discussion

With the implementation of this proposed system, several new identifications were noted. Out of them, this section discusses the major facts which need to be discussed related to document validation and profile rating maintenance.

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.

Table 3.3.1 : Text extraction from different types of same image

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%)

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.

4. CONCLUSION

We identified and confirmed that traffic congestion is the root cause for several negative effects in the country as well as the environment. In order to become a proper solution to this issue , we extended the concept of ride sharing as an Intelligent Complementary Ride Sharing System;which was enriched with several machine learning and image processing techniques and has proven that the results are produced with more distinguished accuracy in the context of ride-sharing. As this is a complex application, it was researched and developed by categorizing into several major sub components.

Out of those components, Document validation and profile rating maintenance does a significant role in the process of developing a proper mechanism in authenticating the valid users for registration and also to enhance the experience of the existing users. Image processing techniques have been used to extract text from the images in the process of document validation, and naive bayes algorithm was used to fulfill the functionality of categorizing the reviews provided by the users in the profile rating maintenance.

It was also proposed as feedback, that the validation process of new user can be optimized further by taking the help of government information providers to verify the identity. The feasibility of this suggestion will be tested and considered in the development of the later versions of the application. Also it was clear that our application does not consume resources unnecessarily from our application as majority of the critical functionalities are done in the server side.

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.

5. REFERENCES

[1] [online] Available at: https://indi.ca/2015/10/colombo-vehicle-statistics-2015/
[Accessed 2 Jan. 2019].

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

[3] Draft Urban Transport Master Plan - Ministry of Transport. (2013). [ebook] Colombo,Sri Lanka: Ministry of Transport, pp.66,74,75.
Available at: http://www.transport.gov.lk/web/images/stories/comtrans.pdf
[Accessed 21 Jan. 2019].

[4] 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.

[5] 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.

[6] V. Ohlsson, “Optical Character and Symbol Recognition using Tesseract”, Dissertation, 2016.

[7] 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.

[8] 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.

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

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

[11] 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.

[12] 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.

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

[14] 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.

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

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

[17] 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.

[18] 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.

[19] 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. 042090.

[20] "Help | Uber", *Uber*, 2019. [Online]. Available: https://help.uber.com/riders. [Accessed: 30- Jul- 2019].

[21] Ft.lk. (2017). Innovative car pooling app UDIO goes live. [online] Available at: http://www.ft.lk/motor/Innovative-car-pooling-app-UDIO-goes-live/55-645554 [Accessed 18 Jan. 2019].

[22] Carpooling.lk. (2019). Carpooling. [online] Available at: http://www.carpooling.lk/ [Accessed 23 Jan. 2019].

[23] RideShare.lk. (2019). Carpool for better commute and quality of life. [online] Available at: http://www.rideshare.lk/ [Accessed 01 Feb. 2019].

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

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

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

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