ENTREGA

BY

MUBASHIR HUSSAIN

NOMAN ASIF

WAJEEH ZIA UDDIN

MUHAMMAD USAMA

Submitted to the Faculty of Computer Software Engineering Department
National University of Sciences and Technology, Islamabadin partial fulfillment
for the requirements of a B.E Degree in Computer Software Engineering

June 2019

ABSTRACT

ENTREGA

Mobile application for Entrega is proposed as a platform to enable the citizens to easily use the loading vehicles under a systematic approach. Entrega which is an online delivery service solve the problems of the people to a huge extend. Although careem bike service, careem pickup services and tcs hazir service are performing the task of loading/delivery services. But they are limited to specified amount and size of package. They can carry out small packages with limited dimension and are not fully reserved to the delivery service. Due to time taking and high cost of loading vehicles people lose motivation. Entrega is the solution to all the problems which provides quick, low cost , and different sized transfer of packages/items.

Entrega is based on the concept of careem and uber but it is fully reserved to loading vehicles. Entrega is a android based application which is point to point delivery system that aims at developing a application for loading vehicles.User will place an order then a driver will be assigned which is nearest to it including the vehicle respective to the size of the order.User will be able to track the order in Real time .User will place order including the dimensions of the product with the location. Application will select a vehicle respective to product description and size. After the order for desired vehicle is placed the server will provide multiple routes for the transportation of package by selecting the best possible route. is done notifications will be generated for the overall summary of Process. The scope of the work includes two andriod application and a Back-End server.

The idea is to develop the above application keeping in view the requirement of the general public and implement the system for the betterment and prosperity of the general public by solving their problem ,saving their time budget and struggle and providing a pickup/loading service at their doorstep in just few clicks.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

Certified that work contained in the thesis – ENTREGA – carried out by MubashirHussain, NomanAsif, Wajeeh Zia Uddin and Muhammad Usamain supervision of Asst. Prof Bilal Rauf for partial fulfillment of Degree of Bachelor of Software Engineering is correct and approved.

Approved by

Asst. Prof Bilal Rauf

Department of CSE, MCS

DATED:

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent To our parents, without whose unflinching support and cooperation, a work of this magnitude would not have been possible.

ACKNOWLEDGEMENTS

We would like to thank Allah Almighty for His incessant blessings which have been bestowed upon us. Whatever we have achieved, we owe it to Him, in totality. We are also thankful to our families for their continuous moral support which makes us what we are.

We are extremely grateful to our project supervisor Asst. Prof Bilal Rauf from MCS who in addition to providing valuable technical help and guidance also provided us moral support and encouraged us throughout the development of the project.

We are highly thankful to all of our teachers and staff of MCS who supported and guided us throughout our course work.

Their knowledge, guidance and training enabled us to carry out this whole work. Finally we are grateful to the faculty of Computer Software Department of the Military College of Signals, NUST.

In the end we would like to acknowledge the support provided by all our friends, colleagues and a long list of well-wishers whose prayers and faith in us propelled us towards our goal.

Table of Contents

List of Figures

Chapter 4

List of Tables

Chapter 1

Chapter 3

Chapter 4

Chapter 6

Chapter 1

1. Introduction

1.1 Overview

With the advancement of technology and smart phone being common for all ages, some of the industries have put their foot in Pakistan to get going with their product like Careem, Uber etc. We felt like something is missing in the cab industry of Pakistan. Suppose you are in some mega mall and you have bought a package that is not going to fix in your vehicle. Now what should be done. Most of the people look for a loading vehicle which they can use to transport their parcel which can be of any dimension. This makes them suffer with time, money and finding the loader vehicle is not easy. To overcome this problem here comes “Entrega”.

Entrega is a android based application which is point-to-point delivery system that aims at developing a application for loading vehicles.User will place an order then a driver will be assigned which is nearest to it including the vehicle respective to the size of the order.User will be able to track the order/ride in real time .User will place order including the dimensions of the product with the location. Application will select a vehicle respective to product description and size. After the order for desired vehicle is placed the server will provide multiple routes for the transportation of package by selecting the best possible route.After this is done notifications will be generated for the overall summary of process.

The scope of the work includes two andriod application and a back-end server. Although application like careem and tcs hazir service are doing their part in this field. But no one is specifically targeting the field of loading vehicles. Careem is restricted to only small parcels and items. Entrega which is totally dedicated to loading vehicles provides a variety of vehicles according to the size and dimension of the product. We have made two application sides which will be andriod based. One side serves for the customer. Customer will go through a easy verification process so that we have the information of the customer for use in future. Similarly we have a application for driver as well. These both application are intregated with a back-end

server which will be based on ROR framework and is supported by SQLserver for the database. Both android application and web portal communicate with the online database to get, add and modify data.

1.2 Problem Statement

The purpose of the application is to enable the proper loading vehicle services so that the regular users of the loading vehicle can easily use the service and enhance their performance. One the other hand it helps the drivers of the loading vehicle to get more rides in a systematic manner and earn more. This will help both the user and the driver to save time, budget, other difficulties and improve the journey experiences.

1.3 Scope

Entrega is a android based application which is point-to-point delivery system that aims at developing a application for loading vehicles.User will place an order then a driver will be assigned which is nearest to it including the vehicle respective to the size of the order.User will be able to track the order/ride in real time .User will place order including the dimensions of the product with the location. Application will select a vehicle respective to product description and size. After the order for desired vehicle is placed the server will provide multiple routes for the transportation of package by selecting the best possible route.After this is done notifications will be generated for the overall summary of process.

1.4 Aims and Objectives

The objectives of project include:

1. Using software engineering techniques for gathering requirements during the development process, designing the software, implementing and testing requirements gathered.
2. To understand the concept of ROR ,Rest API, HTML, CSS ,Jquery, JS and react native that will be used in the project to design interface for display on screen and back-end server.

3. To learn application's hardware and software architecture
4. To learn web development
5. To learn database design and development.
6. To learn data transfer via internet.

1.5 Intended Audience and Reading Suggestions

The thesis report of Entrega is meant for all the stake holders.

1. **Project supervisor**: This document will allow the supervisor to supervise the project and guidethe team in an effective way. It will be used by him to check whether all the requirements have been understood and if all the requirements have been properly implemented or not.
2. **Developers:** Project developers will be able to understand the methodology adopted and personalize the product.
3. **Testers:** The testers of the system can check user requirements from this document and develop the test document accordingly. It will help them to test in a systematic and organized manner.
4. **Documentation writers:** The document can serve as a future reference for otherversions of the document and other documents like design documents, test documents and maintenance.
5. **UG Project Evaluation team**: It will help the evaluation team to evaluate the progress of FYP project. The document will provide the evaluators with the scope, requirements and details of the project to be made. It will also be used as basis for the evaluation of the implementation and final project.
6. **Students:** Any student who requires any reference or help can read this report.

1.6 Organization

The first part of thesis is the abstract which describes the main details of Entrega, followed by the introduction section which specifies the problem statement, approach, scope and objectives. The literature review section state the various resources read online before the commencement of

the project. They include learning about all the required skills, software's and other queries. The design and development part illustrate the diagrams which describe the detailed design of the Entrega, its components, interfaces and data necessary for the implementation phase. The analysis and evaluation part give details of the black box testing, unit testing and system integration testing; actual results against expected results. The future work gives states the enhancements that can be applied to the application.

1.7 Deliverables

Deliverable Name	Deliverable Summary Description
Software Requirements Specification(SRS) Document	Complete Description of what the system will do, who will use it. Detailed description of functional and non-functional requirements and the system features.
Design Document	Complete description of how the system will be implemented i.e. the detailed design.
Code	Complete code with the API.
Testing Document	The whole system is tested according to the specification described in the SRS document. Black box, unit and System integration testing is done.
Complete System	Complete working system.

Table 1.1 : Deliverables

Chapter 2

2. Literature Review

2.1 Introduction

The overall goal of this chapter is firstly to establish the significance of the general field of study, then identifying a place where a new contribution could be made. Conventional procedure of getting a loading vehicle is way old and time taking. It also needs more budget. Careem launched the bike service for transfer of parcel but it was a for a limited amount and size of parcels. Similarly tcs hazir service also has limited amount and size of products/parcels that it takes from the customer. Both of these did not provide the complete solution. Entrega supports a variety of dimensions and size of the product/parcels. It is more efficient and less time taking. Removed all the drawback in careem and hazir services respectively.

The scope of the work includes two andriod application and a back-end server. Although application like careem and tcs hazir service are doing there part in this field. But no one is specifically targeting the field of loading vehicles. Careem is restricted to only small parcels and items. Entrega which is totally dedicated to loading vehicles provides a variety of vehicles according to the size and dimension of the product. We have made two application sides which will be andriod based. One side serves for the customer. Customer will go through a easy verification process so that we have the information of the customer for use in future. Similarly we have a application for driver as well. These both application are intregated with a back-end server which will be based on ROR framework and is supported by SQLserver for the database. Both android application and web portal communicate with the online database to get, add and modify data.

The main purpose of the project is to facilitate the loading vehicle owners with more rides and earning. And the customers will also have easy transfer of their product at affordable price and in less time. It will improve the standards of the loading vehicle system in Pakistan.

2.2 Related Work

Careem launched the bike service for transfer of parcel but it was a for a limited amount and size of parcels. Similarly tcs hazir service also has limited amount and size of products/parcels that it takes from the customer. Both of these did not provide the complete solution. Entrega supports a variety of dimensions and size of the product/parcels. It is more efficient and less time taking. Removed the entire drawback in careem and hazir services respectively. Similarly careem launched pickup service which is also dedicated to the loading category. But the pick-up service is limited to only small packages and do not have scope of vehicles available.

Uber freight is also a similar type of the setup. It is in UAE, Qatar and other gulf countries. Uber freight in limited to long routes and is not available in Pakistan. So, Entrega has an edge as it is new in Pakistan and has a small town business and work and is easy for the customers as compared to Uber freight.

Similarly Us shipping is another type of the similar project which is only used for shipping of stuff to long distance and under certain conditions and do not involve the live tracking and online booking.

While working in Flip kart's last-mile logistics division in 2013, AravindSanka noticed a demand-supply mismatch for intra-city trucks. Strangely, many companies like Flip kart struggled to find trucks to ferry goods even as several vehicles idled barely a few hundred meters away. Sanka, along with friends PavanGuntupalli and RishikeshRamanath, spoke to over 400 drivers and 100 small businesses to discover there were discrepancies not only in vehicle availability but also rates. Drivers charged whatever they liked and there was no means to track the loaded vehicles. The trio then hit upon the idea of using technology to make intra-city logistics more transparent and less painful through their Bangalore-based venture TheKarrier.

In Mumbai, IIT graduates PranavGoel, UttamDigga and VikasChoudhary sensed a similar need and floated The Porter — an online marketplace to book mini-trucks and light commercial vehicles for intra-city transportation of goods.

The boom in e-commerce, coupled with the needs of frequently relocating students and professionals has led to a spurt in the need for intra-city transportation — currently an estimated $10-billion industry in India, which a plethora of start-ups are eager to partake of. “We are aggregators. We don’t own assets (trucks, mini-trucks et al) and just connect consumers with owners,” says Goal, co-founder of The Porter.

Bangalore-based Blow horn is another of these start-ups whose model is somewhat akin to online taxi services Ola and Uber. A customer looking to hire a truck should book either through an app or the company’s call centre. Based on the requirement, the nearest available truck is sent to the customer. The trucks typically have carrying capacity between 650kg and one tone.

2.3 Proposed System

Every day, thousands of citizens have to face multitudes of problems related to the transportation of items and parcels/products. People have to travel a long distance to find out suitable vehicle for their item which is time taking and expensive. To remove this burden we have proposed an online platform Entrega which is a android based application which is point-to-point delivery system that aims at developing a application for loading vehicles.User will place an order then a driver will be assigned which is nearest to it including the vehicle respective to the size of the order.User will be able to track the order/ride in real time .User will place order including the dimensions of the product with the location. Application will select a vehicle respective to product description and size. After the order for desired vehicle is placed the server will provide multiple routes for the transportation of package by selecting the best possible route.After this is done notifications will be generated for the overall summary of process.

We have made two application sides which will be andriod based. One side serves for the customer. Customer will go through a easy verification process so that we have the information of the customer for use in future. Similarly we have a application for driver as well. These both application are intregated with a back-end server which will be based on ROR framework and is supported by SQLserver for the database. Both android application and web portal communicate with the online database to get, add and modify data. This provides with a short overview of the proposed system.

Chapter 3

3. Overall Description

This part of the document contains information about the product, its features, perspective, users' characteristics and constraints.

3.1 Introduction

3.1.1 Purpose

The purpose of this chapter is to give the user a clear and precise description of the functionality of the Entrega , a software system for loading vehicles.This chapter is aimed to eliminate ambiguities and misunderstandings that may exist.

This chapter covers all basic features, objectives and attributes of the proposed system. It explains the system's interface, and the constraints under which it must function and how the system will respond to external stimuli. This chapter will give the user a clear and precise description of the functionality of the Entrega software.

For the user, this chapter will explain all functions that the software should perform. For the developer, it will be a reference point during software design, implementation and maintenance. This chapter encompasses the requirements for version-1 of Entrega.

3.1.2 Intended Audience

Intended audience Includes:

1. **Project supervisor**: This document will allow the supervisor to supervise the project and guide the team in an effective way. It will be used by him to check whether all the requirements have been understood and if all the requirements have been properly implemented or not.

2. **Developers:** Project developers will be able to understand the methodology adopted and personalize the product.

3. **Testers:** The testers of the system can check user requirements from this SRSand develop the test document accordingly. It will help them to test in a systematic andorganized manner.
4. **Documentation writers:** The document can serve as a future reference for other versions of the SRS and other documents like design documents, test documents and maintenance.
5. **UG Project Evaluation team**: It will help the evaluation team to evaluate the progress of FYP project. The document will provide the evaluators with the scope, requirements and details of the project to be made. It will also be used as basis for the evaluation of the implementation and final project.

3.1.3 Reading Suggestions

All level 1 and level 2 headings are given in the table of contents, but the lower sub headings are not included. Each main heading is succeeded by a number of sub headings, which are all in bold format. The product overview is given at the start, succeeded by the complete detailed features, including both functional and non-functional requirements. The entire interfaces are also described. The chapter ends with appendices, including a glossary.

3.1.4 Project Scope

For	People transporting goods from one place to the other, daily transportation of goods, shifting etc. Drivers of the loading vehicles to get more chances to earn.
What	A system consisting of a back-end server and to android interfaces one for customer and other for driver.
The	Entrega(A Spanish word which means delivery) .
Is	A web interface and android application
That	Provides cost effective and convenient to customers to perform transportation and drivers to earn more effective income.

Table 3.1: Project Scope

3.2 Overall Description

3.2.1 Product Perspective

With the advancement of technology and smart phone being common for all ages, some of the industries have put their foot in Pakistan to get going with their product like Careem, Uber etc. We felt like something is missing in the cab industry of Pakistan. Suppose you are in some mega mall and you have bought a package that is not going to fix in your vehicle. Now what should be done. Most of the people look for a loading vehicle which they can use to transport their parcel which can be of any dimension. This makes them suffer with time, money and finding the loader vehicle is not easy. To overcome this problem here comes “Entrega”.

Entrega is a android based application which is point-to-point delivery system that aims at developing a application for loading vehicles.User will place an order then a driver will be assigned which is nearest to it including the vehicle respective to the size of the order.User will be able to track the order/ride in real time .User will place order including the dimensions of the product with the location. Application will select a vehicle respective to product description and size. After the order for desired vehicle is placed the server will provide multiple routes for the transportation of package by selecting the best possible route.After this is done notifications will be generated for the overall summary of process.

The scope of the work includes two andriod application and a back-end server. Although application like careem and tcs hazir service are doing their part in this field. But no one is specifically targeting the field of loading vehicles. Careem is restricted to only small parcels and items. Entrega which is totally dedicated to loading vehicles provides a variety of vehicles according to the size and dimension of the product. We have made two application sides which will be andriod based. One side serves for the customer. Customer will go through a easy verification process so that we have the information of the customer for use in future. Similarly we have a application for driver as well. These both application are intregated with a back-end server which will be based on ROR framework and is supported by SQLserver for the database. Both android application and web portal communicate with the online database to get, add and modify data.

3.2.2 Product Functions

The main features of Entrega back-end server are highlighted below:

1. Log in mechanism.
2. Account creation, deletion, edit and update.
3. To fetchdriver the information of his rides and his total income.
4. To enable administrator to track, guide and keep a list of rides, expenses and other information.
5. To display progress of every driver.
6. To fetch driver the list of the rides assigned to him.
7. To let Admin to log out.

The main features of Entrega customer application interface are stated below:

1. Log in mechanism.
2. To let customer input the pick-up and drop off location.
3. To let customer, calculate estimated fare and driver's arrival time.
4. To let customer, know the vehicle name, color, number and driver information.
5. To enable customer to cancel ride within 2 minutes of the booking without fine.
6. Mechanism to calculate record and display rides of the customer and total fare.
7. Allow customer to track the ride.
8. To allow customer to log out.

The main features of Entrega drivers application interface are stated below:

1. Log in mechanism
2. To let Driver to see the pick-up and drop off location.
3. To let driver, check the total number of rides and his income.
4. To enable Driver to cancel ride within 2 minutes of the booking without fine.
5. To allow driver to have the customers contact number and customer's name.
6. To allow driver to log out.

3.2.3 User classes and characteristics

The different types of users are:

3.2.3.1 Administration users (Frequent users):

The administrator will be able to maintain all the accounts. Add new accounts, delete accounts no more required and edit information of all accounts.

3.2.3.2 Driver (Frequent users):

Driver will be able to view all his rides, his expected income, his acceptance rate and his rating by the customer. Driver will also be able to see the pickup location and other credentials of the customer.

3.2.3.3 Customer (Frequent users):

Customer will be able to interact with the system by selecting pickup location and drop off location. Customer will be able to track the driver's location and will be able to get all the required credentials of the drivers. Customer will be able to rate the driver after the ride has ended.

3.2.3.4 Developers (occasional user):

The developers will use this system at the developing time and at the time any defect occurs in the product during maintenance.

3.2.3.5 Tester (occasional user):

The testers will use the product at the time oftesting to make scenarios and check the functionality of the product.

3.2.4 Operating Environment

3.2.4.1 Software Interfaces

1. Entrega back-end server should be able to run on any version of the following web browsers: Microsoft Internet Explorer, Mozilla Firefox, Netscape, Opera, Safari and Google Chrome.

2. Primary Operating System supported by Entrega application Interface will be android.
3. Entrega should be able to run on Apache Web server configured in a stable Linux/Unix/MAC/Windows machine.
4. Entrega should work with MySQL database management system.
5. Entrega application should be able to run on all android devices with basic hardware requirements fulfilled that run Android OS 4.2 or above.
6. The app will require access to the device GPS, and request permission for location tracking via the Android OS.

3.2.4.2 Hardware Interfaces

3.2.4.2.1 Computer System

1. System shall have keyboard input.
2. System shall have mouse input.
3. System shall have a monitor.
4. System shall have a working internet connection and the hardware requirements that come with it (Network card, Ethernet Port, Modem etc.)

3.2.4.2.2 Mobile Device

1. Android Device (Cell phone or Tablet) running Android 4.2 or later, color display.
2. Touch Screen with hap tic feedback on key presses (Android Keyboard).
3. Global Positioning System.

3.2.4.2.3 Web and Database Server

1. To process requests and retrieve/store data.

3.2.4.2.4 Communications Interfaces

1. System shall be connected to the web services that we will create.
2. To access the data and request, PUSH, PULL and GET protocols can be used.
3. Communication between the Web Interface and the server will be through HTTP over a web browser.

4. Communication between the Android application and the server will be through API.

3.2.4.2.5 Programming Interface

Programming interfaces for project are:

1. Ruby on Rails.
2. Rails framework(REST API).
3. MYSQL .
4. Ubuntu
5. Postman
6. Android Studio/React Native.
7. Google Maps for GEO-Fencing
8. The system will interact with Database on loading and saving of the updates of the customers and the drivers.

3.2.5 Design and Implementation Constraints:

1. An Internet connection is also a constraint for the application. Since the application fetches data from the database over the internet and stores data in the database, it is crucial that there is an internet connection for the application to function.
2. Both the back-end server and the android applications will be constrained by the capacity of the database. Since the database is shared between both application it may be forced to queue incoming requests and therefore increase the time it takes to fetch data.

3.2.6 Assumptions and Dependencies

3.2.6.1 Administration: This back-end server will have an admin account with a username and password. The admin will be able to create and manage other accounts.

3.2.6.2 Memory and CPU

1. 2 GB RAM or higher.
2. 10 GB hard disk memory or higher

3. 1.6 GHz processor or higher
4. Core i5 or higher
5. 3.0 USB port

3.2.7 User Documentation

A user manual will be provided to the users in which separate instructions will be given according to the particular user i.e., customer, driver, developers and testers. It will include the details of the software's working. Help documents will also be a part of the system. The project report will also be available for the users which will highlight the software's features, working and procedures.

3.3 External Interface Requirement

This section contains the requirements specification for interfaces among different modules of the software and their external capabilities.

3.3.1 User Interfaces:

The requirements for user interfaces would be;

1. The interface shall be user friendly and very simple to use.
2. The buttons shall be big to make their selection easy.
3. The interface of application shall be in some light color to make it more attractive.
4. Each screen will be explanatory regarding the options and functionality provided by the system.

3.3.2 Hardware Interfaces:

Following hardware is required for the project.

1. Android Device with Android Version 4.2 or above.
2. Personal Computer

3.3.3 Software Interfaces

1. Ruby on Rails.
2. Rails framework(REST API).
3. MYSQL .

4. Ubuntu/Ubuntu terminal
5. Postman
6. Android Studio/React Native.
7. Google Maps for GEO-Fencing
8. The system will interact with Database on loading and saving of the updates of the customers and the drivers.

3.4System Features

Backend Server

3.4.1 Login

This function allows the users (admin) to log into the web portal by providing the required information for login. This function is of high priority as all the other functions are performed after the successful completion of this feature.

3.4.1.1 Stimulus/Response Sequences:

1. Enter the user name in the given field.
2. Enter the password in the given field.
3. Press login to enter the portal.
4. Next screen of the portal will appear after successful login.
5. If incorrect username or password is entered, login will not be successful.

3.4.1.2 Functional Requirements:

REQ-1: The database shall contain definitions for user names, passwords and access roles, e.g. administrator role, drivers role and customer role.

REQ-2: The password as entered shall display the character '*' in place of each password character entered.

REQ-3: Portal shall be able to notify invalid username and password if it is not found in database.

REQ-4: A login failure shall redisplay the login method with all fields blank.

REQ-5: If both username and password are valid, portal shall load next screen.

3.4.2 Add User

This function allows the admin to register user in the database. The admin can add another user account by providing the required information for the addition. The required information may include name, contact number, email, gender etc.This function is of medium priority.

3.4.3 Delete Driver and Customer:

This function allows the admin to delete the user (driver and Customer) and his information saved in the database.

This function is of low priority because it will be used very rarely and other functions do not depend on this function.

3.4.3.1Stimulus/Response Sequences:

1- Admin can delete by clicking on the 'Delete User' button.

2- On clicking the delete option dialog box will appear to confirm deletion.

3- Click on 'Ok'.

4- User will be deleted from database.

3.4.3.2 Functional Requirements:

REQ-6: The system shall allow admin to delete any user.

REQ-7: The system shall display a dialog box to reconfirm deletion of user when admin requests to delete a user.

REQ-8: The system shall delete information of the user from the database.

3.4.4 Select Driver

3.4.4.1 Description and Priority:

This feature allows the admin to select the driver. It will enable the admin to view information, progress etc. selecting the name of the driver.

This function is of high priority because it will be used frequently and other functions depend upon it.

3.4.4.2 Stimulus/Response Sequences:

1- Admin will click on the name shown on his home page in the list

2- Driver's screen will open

3.4.4.3 Functional Requirements:

REQ-9: The system shall allow Admin to select User (Driver, Customer) from the list available on his home screen.

REQ-10: The system shall display information screen of the selected drivers.

3.4.5 View Driver's Progress

3.4.5.1 Description and Priority:

This feature allows the admin and user to view a summary of driver's progress and performed rides. This will help the admin to analyze the performance of driver. This function is of high priority.

3.4.5.2 Stimulus/Response Sequences:

1- Click on 'View Progress'

2- Progress of the selected driver will be displayed.

3.4.5.3 Functional Requirements:

REQ-11: The system shall display progress table on request of user.

3.4.6 Logout

3.4.6.1 Description and Priority:

This function will enable the user (Admin) to logout of the backend server. The option of logout will be available on every screen.

This function is of medium priority because user to shall use this feature to end the session.

3.4.6.2 Stimulus/Response Sequences

1. Select the logout option available at different screens.
2. The applications will successfully logout.

ANDROID APPLICATIONS

3.4.7 Login to Application

3.4.7.1 Description and Priority:

This function allows the users (drivers and customer) to log into the application by providing the required information for login.

This function is of high priority as all the other functions are performed after the successful completion of this feature.

3.4.7.2 Stimulus/Response Sequences:

1. Enter the user name in the given field.
2. Enter the password in the given field.
3. Press login to enter the application.
4. Next screen of the application will appear after successful login.
5. If incorrect username or password is entered, login will not be successful.

3.4.7.3 Functional Requirements:

REQ-1: Application shall notify invalid username and password if it is not found in database.

REQ-2: If the invalid username or invalid password is entered, application shall not load next screen and generate an error message.

REQ-3: The application shall load the next screen when valid and authenticates username and passwords are entered.

3.4.8 Select Option:

3.4.8.1 Description and Priority:

This function allows the user (Customer) to select vehicle and pickup, drop of location. There will be 3 different vehicles available. This function is of high priority.

3.4.8.2 Stimulus/Response Sequences:

1. Log into the application.
2. The Vehicle, pickup and drop off selector with map screen will open up.
3. Select any vehicle, pick up and drop of location on the selector screen and book the ride.

3.4.8.3 Functional Requirements:

REQ-4: The application shall allow user to select vehicle, pickup and drop off location.

REQ-5: The application shall open the corresponding vehicle allotted screens selected from the first menu.

REQ-6: The application shall enable the customer to see credentials of the driver who is allotted the ride.

3.4.9 Calculate fare:

3.4.9.1 Description and Priority:

The feature will allow the user to system to calculate fare or estimated fare based on pre-set calculating method or formulae.

3.4.9.2 Stimulus/Response Sequences:

3.4.9.3 Functional Requirements:

REQ-7: The application shall be able to calculate the fare or estimated fare of the ride.

3.4.10 Logout:

3.4.10.1 Description and Priority:

This function will enable the users (customer and driver) to logout of the application. The option of logout will be available on every screen.

This function is of medium priority because user to shut down the application at any stage of using the application.

3.4.10.2 Stimulus/Response Sequences

1. Select the logout option available at different screens.
2. The applications will successfully logout.

3.4.10.3 Functional Requirements:

REQ-11: The system shall allow user to log out from the application.

REQ-12: All subsequent displayed pages shall contain a logout control.

REQ-13: The system shall display the main log in page after user logs out.

3.5 Other Non-Functional Requirements

3.5.1 Performance Requirements:

PR-1 The user should be connected to internet and should have android device with android version 4.2 or above.

PR-2 The system should run in real-time.

PR-3 The system should be able to track all the users (customer and driver).

PR-4 On average a database query shall take less than 0.5 seconds.

PR-5 Database connection active.

PR-6 The database shall be available 99% of the time.

PR-7 On average no page shall take more than 2 seconds to access.

PR-8 The portal shall be accessible via an internet connection 99.9% of the time during business hours.

3.5.2 Safety Requirements:

SR-1 The area where users performing exercise is to have no obstacles or obstructions in the way of the user.

SR-2 The system shall not accidently lose/delete the files associated with it, such as information related to drivers and the customers.

SR-3 If the applications crash during addition, deletion or editing there will be no change in the database.

3.5.3 Security Requirements:

SR-1 Only authorized admin, drivers and customers shall be permitted to access information and progress.

3.5.4 Software Quality Attributes:

3.5.4.1 Availability

The system should be available 24/7 as long as the computer system and network connection works properly.

3.5.4.2 Correctness

If the users give right/defined commands for a particular action the options should be selected correctly.

3.5.4.3 Extensibility and Maintainability

1. Changes required by law will be applied in at least 3 months.
2. The system can be extended for high school/university level studies.
3. The system can be improved to track more than one person at a time.

3.5.4.4 Portability

The system can be installed on any compatible hardware meeting the requirements stated in section 3.2.

3.5.4.5 Reliability

The system will be available to users 98% of normal working hours.

3.5.4.6 Usability

Someone with little to none technical experience in the operations of electronics should be able setup and use this system by following a simple set of instructions.

3.5.4.7 Business Rules

N/A

Chapter 4

4. Design and Development

4.1 Introduction

4.1.1 Purpose

This chapter describes the architecture and system design of Entrega. It mostly contains different design diagrams and their explanation. The document is intended to inform stakeholders, developers and support team at organization of the details of the design and the design process. This document will help the developer(s) in implementation and maintenance of the Software.

The purpose of the application is to enable the proper loading vehicle services based on which the users can avail the services based on which the completion of the requirements of the system can be implemented as well as the structure of the application will be based on parallel users based on which several users will be able to use the services with complete domain modeling as well as the system analytics for the services can be more appropriate.

4.1.2 Scope

The purpose of this Software Requirements Specification document is to elucidate the software requirements for final year degree project titled “ENTREGA”. This document covers all basic features, objectives and attributes of the proposed system. It explains the system’s interface, and the constraints under which it must function and how the system will respond to external stimuli. This document will give the user a clear and precise description of the functionality of the Entrega software. For the user, the chapter 4 will explain all functions that the software should perform. For the developer, it will be a reference point during software design, implementation, testing and maintenance. The scope of the project is listed below with complete details.

1. System will be working with internet based on which the connectivity for each user will be appropriate.
2. Every user should have to provide the complete login information based on which the execution modules of the system can be deployed in the complex environment.

3. User will place the request , based on which the nearest cars will be accessing the users based on which the completion of the workflow can be managed.
4. The fares will be calculated by the users which will be shown to the end users to have the complete and the core functional system based on which the completion of the projects can be determined.
5. Each ride record will be saved into the database based on which the rating of the customer as well as well as the driver will be implemented based on which the completion of the services can be deployed in an appropriate way.

4.1.3 Document Overview

This document shows the design and working of Entrega. It starts from higher level details for a non-technical reader to understand just by seeing the diagrams to the lower level details that aid the developer to code and understand other technical details of the application.

In Section 2, the **System Architecture Description** gives a detailed overview of the application. Section 2.1 Overview of Modules/components shows the main component of the application and their inter-relationships. Section 2.2 Structure and Relationships shows the higher level details system working by the means of system block, activity, state transition, and use case diagrams. Lower level details are described using the class, sequence diagrams and structure chart. Section 2.3 describes how the application is designed to curb the tendency of user interface Issues and problems during User Interaction.

In Section 3, **Detailed Description of Component** is given to show the working of modules with low level details. It shows the purpose, function, subordinates, dependencies, interfaces, resources, processing and data of the components and their relationships with each other.

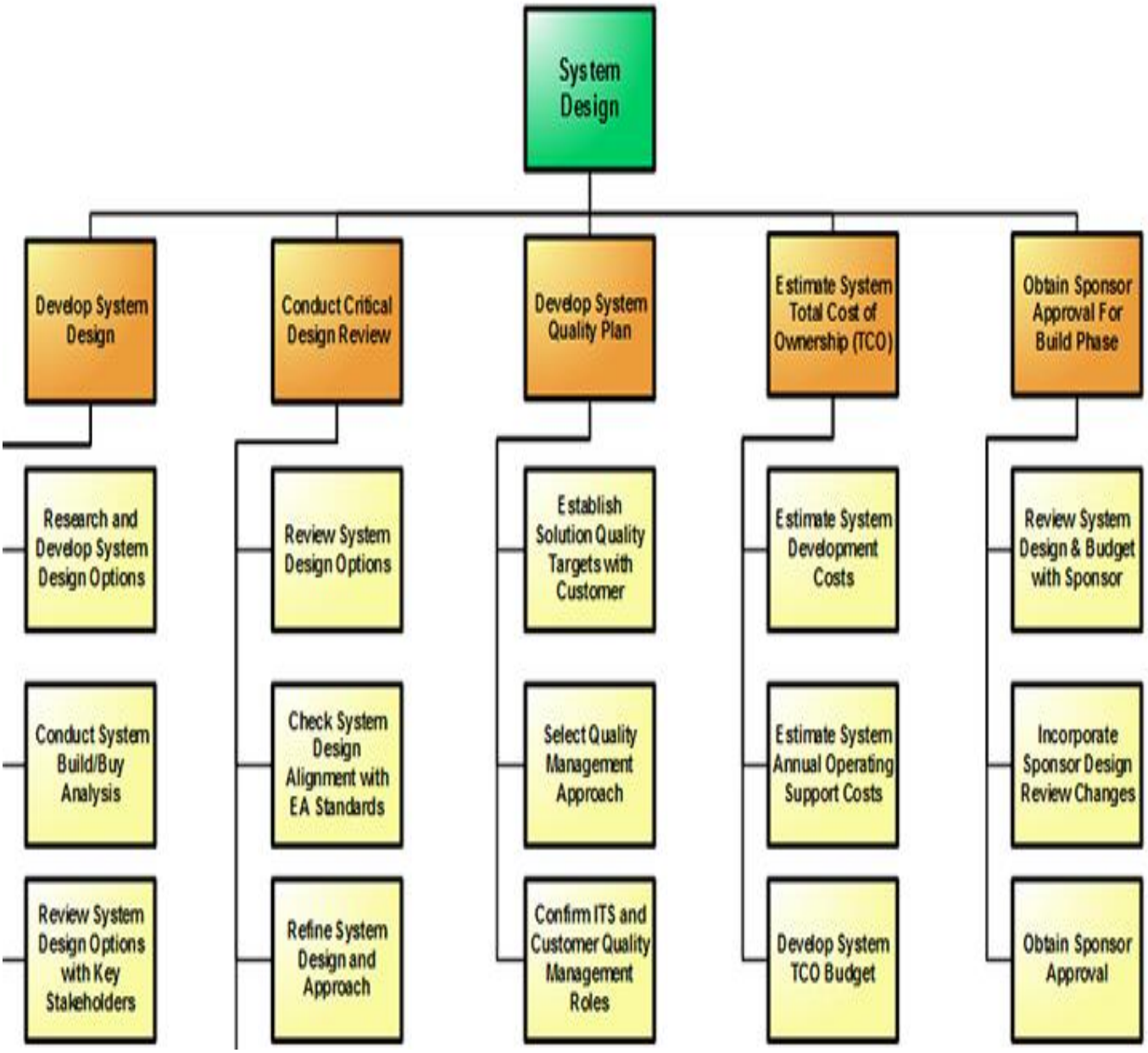
Section 4 shows the **Reuse and Relationship to other Products** i.e.; information about work done in the same project before and any reuse of the same work. The section also provides a key to reuse this system for further upgrades.

Section 5 **'Design decisions and tradeoffs'** shows the architecture style and design pattern of the application.

Section 6 **'shows the pseudo code of components**' which will help the developer to code the components.

4.2 Work Breakdown Structure

Figure 4.1 : Work Breakdown Structure



4.3 System Architecture Description

This section provides detailed system architecture of Entrega and Assessment. Overview of system modules, their structure and relationships are described in this section. User interfaces and related issues are also discussed.

4.4 Overview of Modules

The system is based on the complete Navigation of the mobile application based on which the complete scenario of the implementation can be described as well as the information modeling and the functional requirements of the application are based on the complete structural and the functional navigation which the users will search for the nearest cab service as well as the cab service will be able to put the core functional information based on which the analytical methods can be described with respect to the modeling and the analysis based on which the executive methods can be measured. This System Application has following required modules. Here we give a brief overview of all these modules. Detailed descriptions of these modules are presented in section 3.

1. Customer Module: User module is the ways to interact with application. It packages all those screens, dialogs and forms that are visible to user. It provides user access to drivers available and to order products and give them feedback.

2. Feedback System Module: It is the module where the user is facilitated with the ability to both the rider and the customer based on his experience. The customer feedback will have a direct impact on the overall rating of the driver. The weight age of the rating will depend on the previous order history of the customer.

3. Driver Module: This module has a task to receive the notification once the Admin assign driver the task to deliver a specific product. The module will let rider to track customer location to deliver the product to customer.

5. Complaint Module: The complaint module will entertain all kinds of complains related to the whole system of delivery.

6. Admin Panel: This is a hidden module in application and is accessible to user on providing login ID and password. This module has access to application settings and log data.

4.5 Structure and Relationships

This section covers the technical description of bitwise. It shows relationships between different components and how system modules are connected. This section also covers working with respect to

different point-of-views. This also covers its higher and lower levels details, user interfaces, and system architecture and design pattern.

4.5.1 System Block Diagram

This diagram shows the higher level description of the application. It shows all the modules of the system and their associations and flow of data between modules.

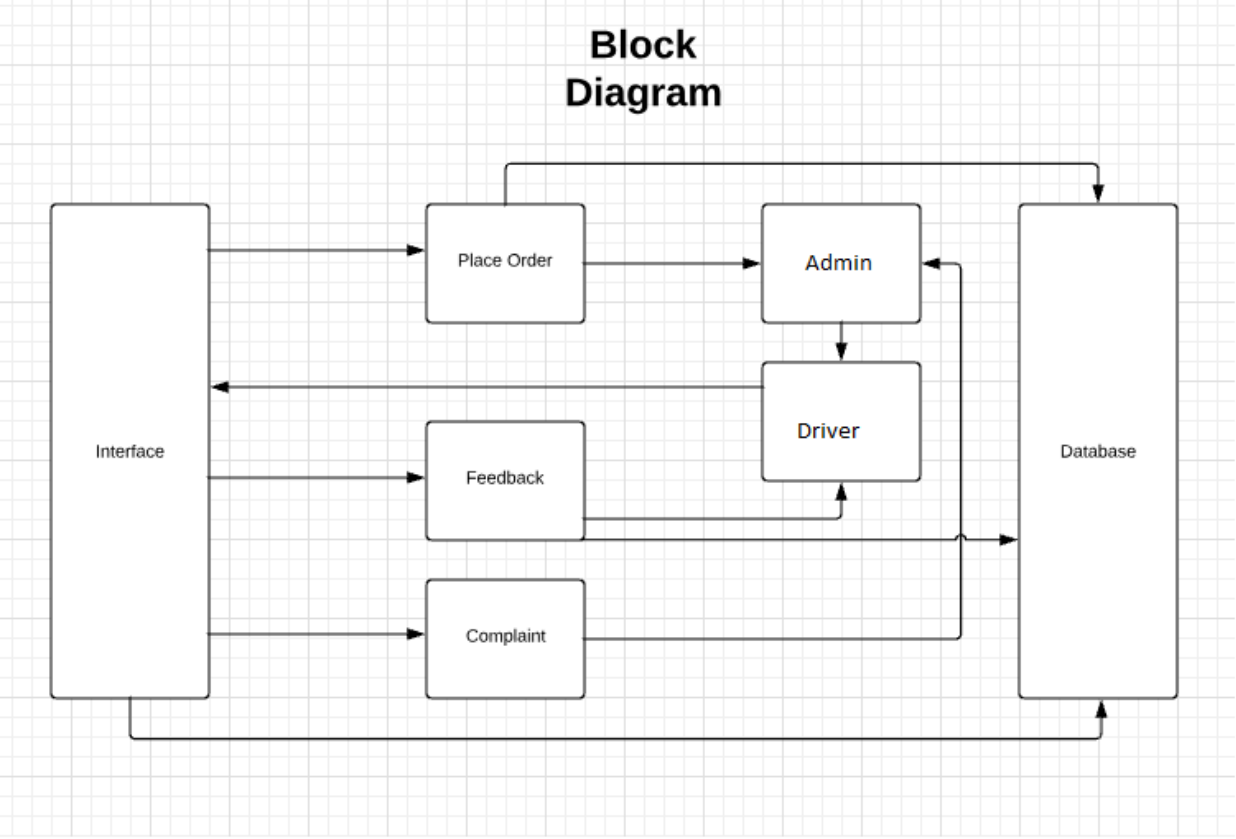


Figure 4.2: Block Diagram

4.5.2 Database Design:

Entrega requires user's username and password, user's personal information and his respective progress along with customer's remarks to be saved in database. There are eleven different tables in database structure which will manage all the requirements mentioned above.

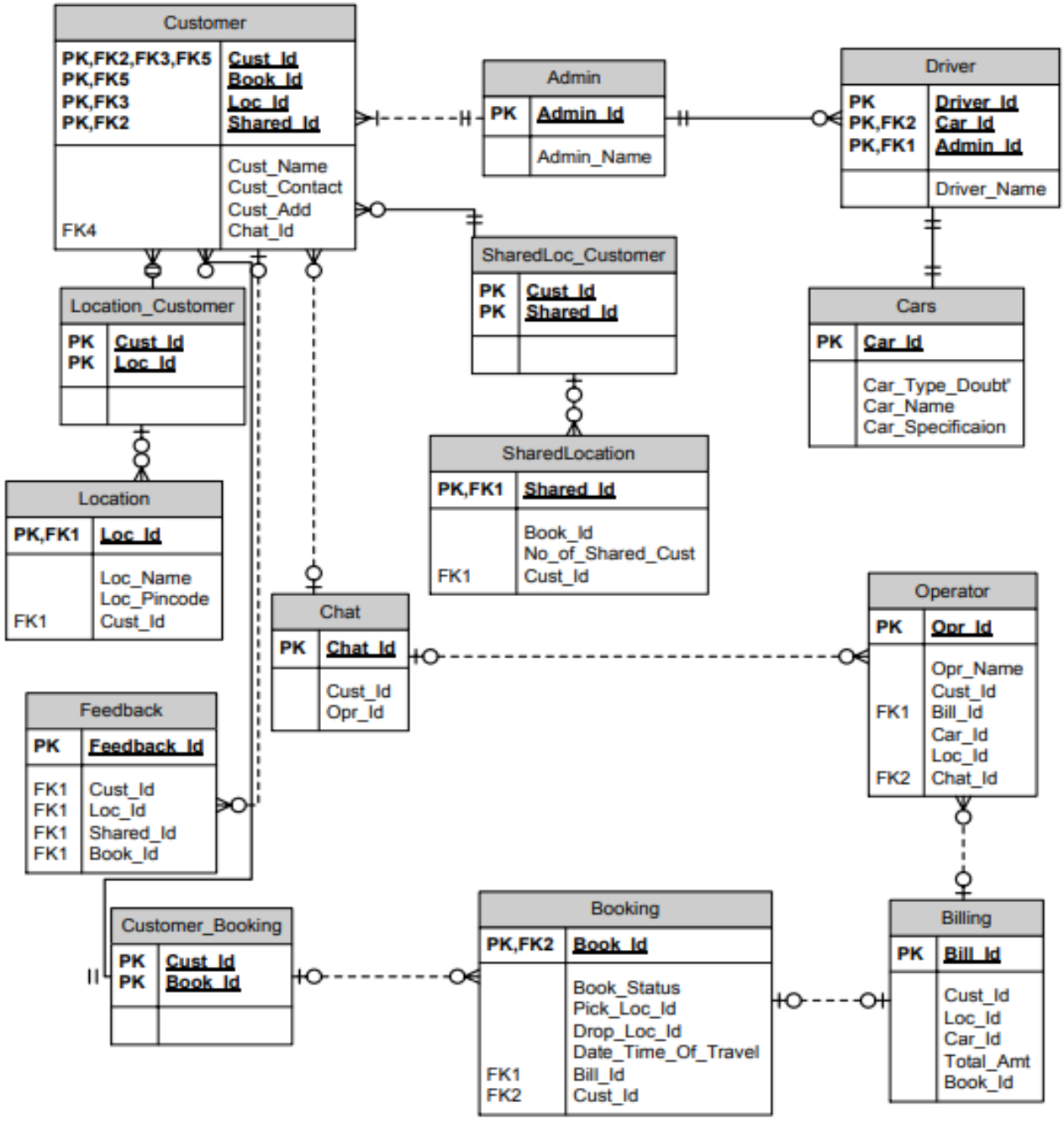


Figure 4.3: ERD Diagram

4.5.3 Use Case Diagram

Following diagram shows course of events that take place when an actor (user and other allowed interactions) interacts with system.

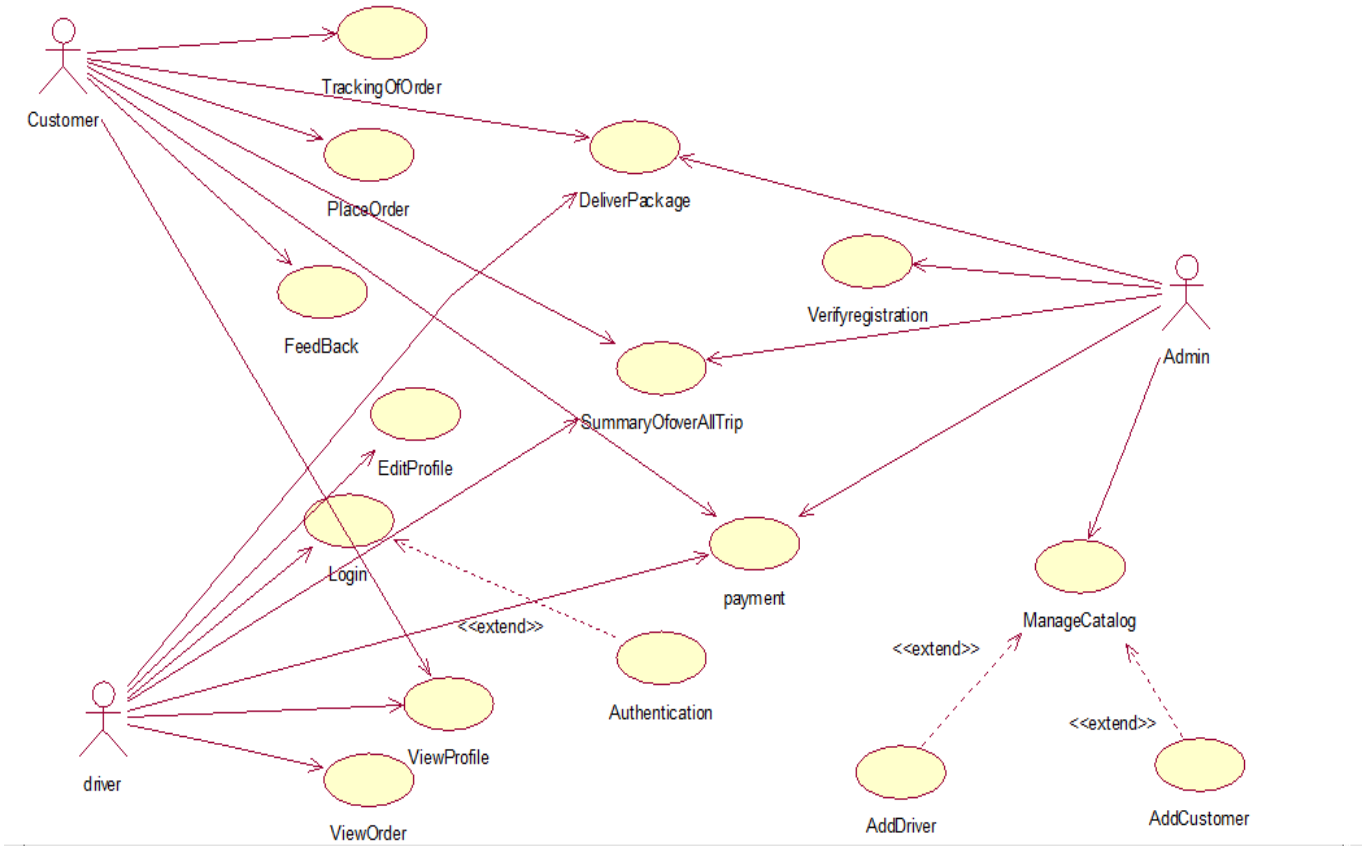


Figure 4.4: Use Case Diagram

4.5.3.1 Actors

1. Admin
2. Customer
3. Driver

Secondary Actor

1. Database

4.5.3.2 Use Cases

1. Login
2. Add user

3. Add driver
4. Delete driver
5. Check payments
6. Tracking of Order
7. Feedback
8. Place Order
9. Summary
10. Help/Feedback

4.5.3.3 Use Case Description

4.5.3.3.1 Login

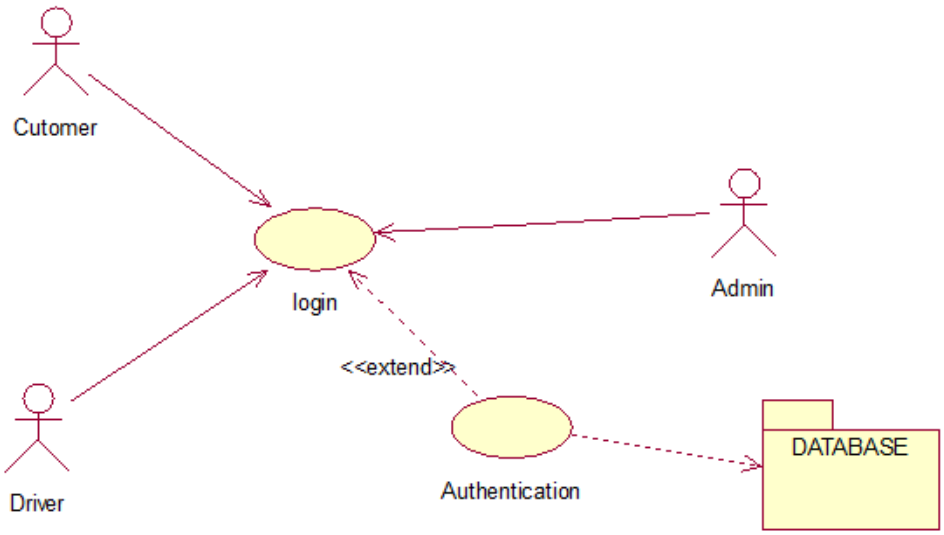
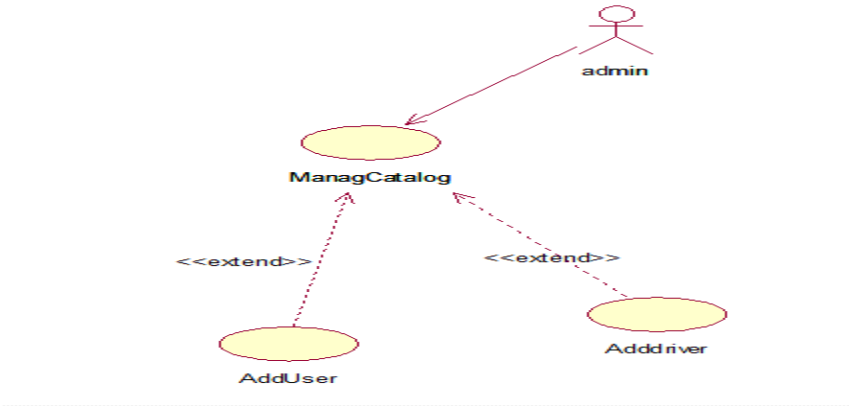


Figure 4.5: Login



Use Case: Log in
Actors: Customer, Driver, Admin and Database
Normal Flow: □ User enters his username in the required field to log in to the application. □ User enters his password. □ User clicks the login button to enter.
Alternate Flow: • If incorrect username or password is entered show an error message and login will not be successful.
Preconditions: Username and password of user must be already registered at the time of coding.
Post conditions: The user successfully logs in
Includes: Authorization by the database
Extends: N/A

4.5.3.3.2 Tracking of Order

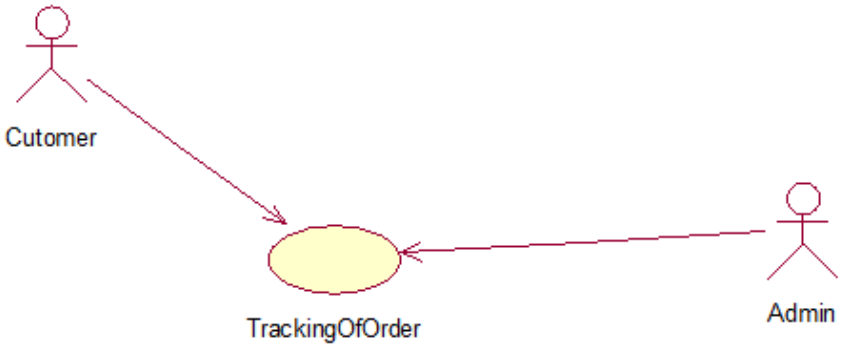
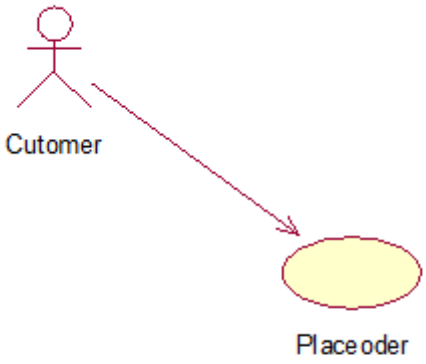


Figure 4.6: Tracking of Order



Use Case: Customer and Admin
Actors: Admin, Customer
Use Case Description: This use case provides the admin and customer to track the ride or the parcel. The required information may include drop of location, arrival time, estimate fare etc.
Normal Flow: • Track of ride. • Current location of the vehicle. • Arrival time of the vehicle.
Alternate Flow: nil

4.5.3.3.3 Feedback

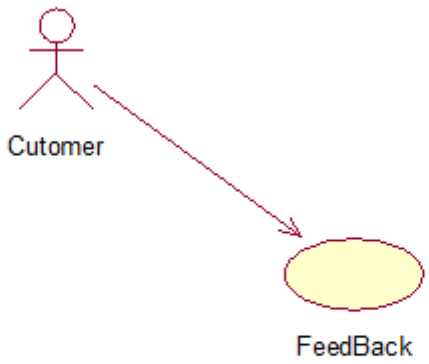


Figure 4.7: Feedback

Use Case: Feedback
Actors: Customer
Normal Flow: Customer adds feedback at the end of ride. Feedback is recorded in database.
Alternate Flow: •NIL
Preconditions:A ride must complete to add feedback for the driver.
Post conditions: The user successfully sends feedback
Includes: Authorization by the database
Extends: N/A

4.5.3.3.4 View Profile

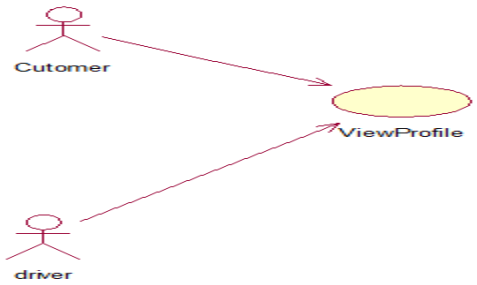
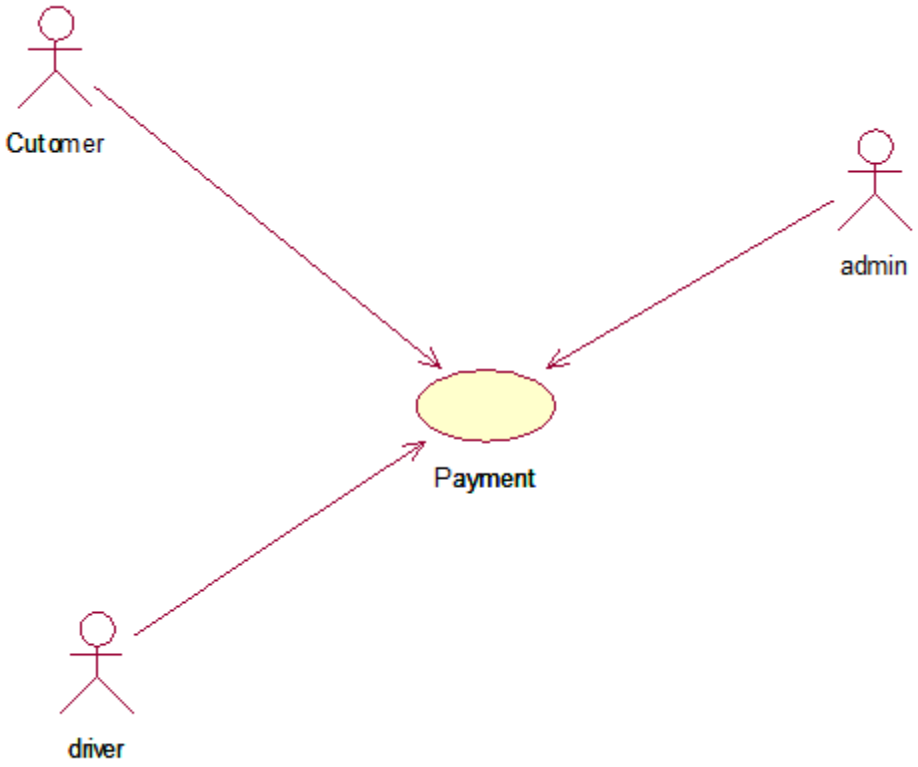


Figure 4.8: View Profile

Use Case: View
Actors: Customer,Database,Driver
Normal Flow: • Customer clicks on drivers profile to view it. • A new screen appears with the credentials of the driver • Driver can also check for customers information with the same procedure •
Alternate Flow: N/A
Preconditions: Click on back button to go on the main menu.
Post conditions: The users successfully view the profiles
Includes: Authorization by the database
Extends: N/A

4.5.3.3.5 PAYMENTS


Cutomer
admin
Payment
driver

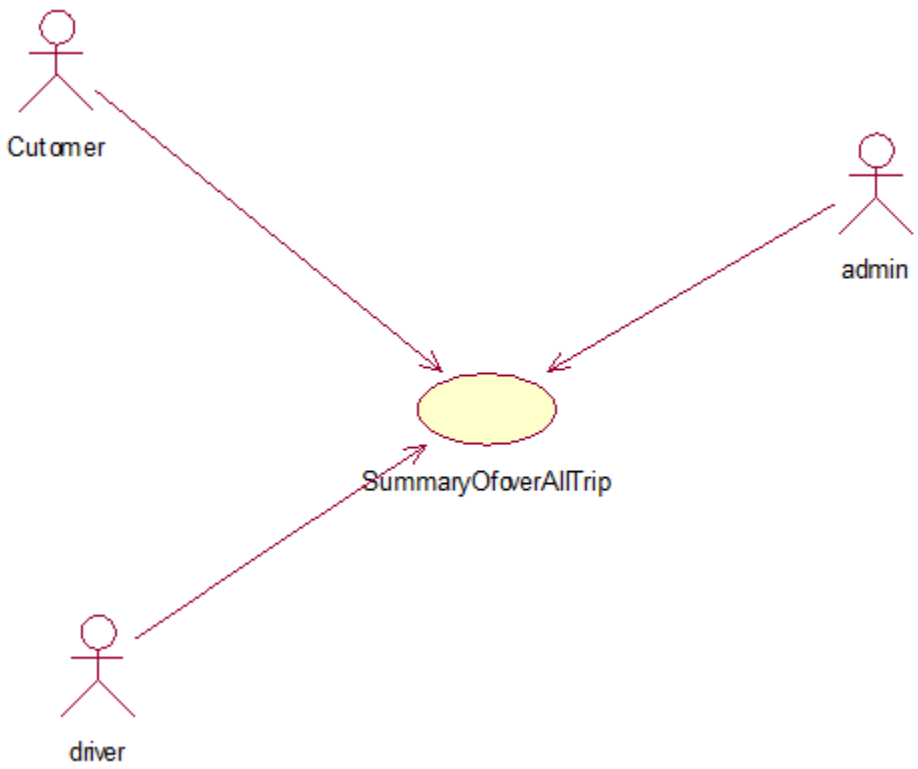


Figure 4.9: Payment

Use Case: Payment (Fare)
Actors: Customer,Database,Driver,Admin
Normal Flow: • Customer puts in the pick-up and drops of location. Then estimated fare is calculated using certain formulae • Similarly driver can also check the fare he gained from all the rides by clicking on payment • Admin can keep track of all the customers and drivers income and expenditure.
Alternate Flow: N/A
Preconditions: Click on back button to go on the main menu.

Post conditions: The users successfully view the payments.
Includes: Authorization by the database
Extends: N/A

4.5.4 Sequence Diagram

Following sequence diagrams show the sequence of activities performed in key use cases described in section 4.5.3.

4.5.4.1 Sequence Diagram 1 (Customer Login/Register)

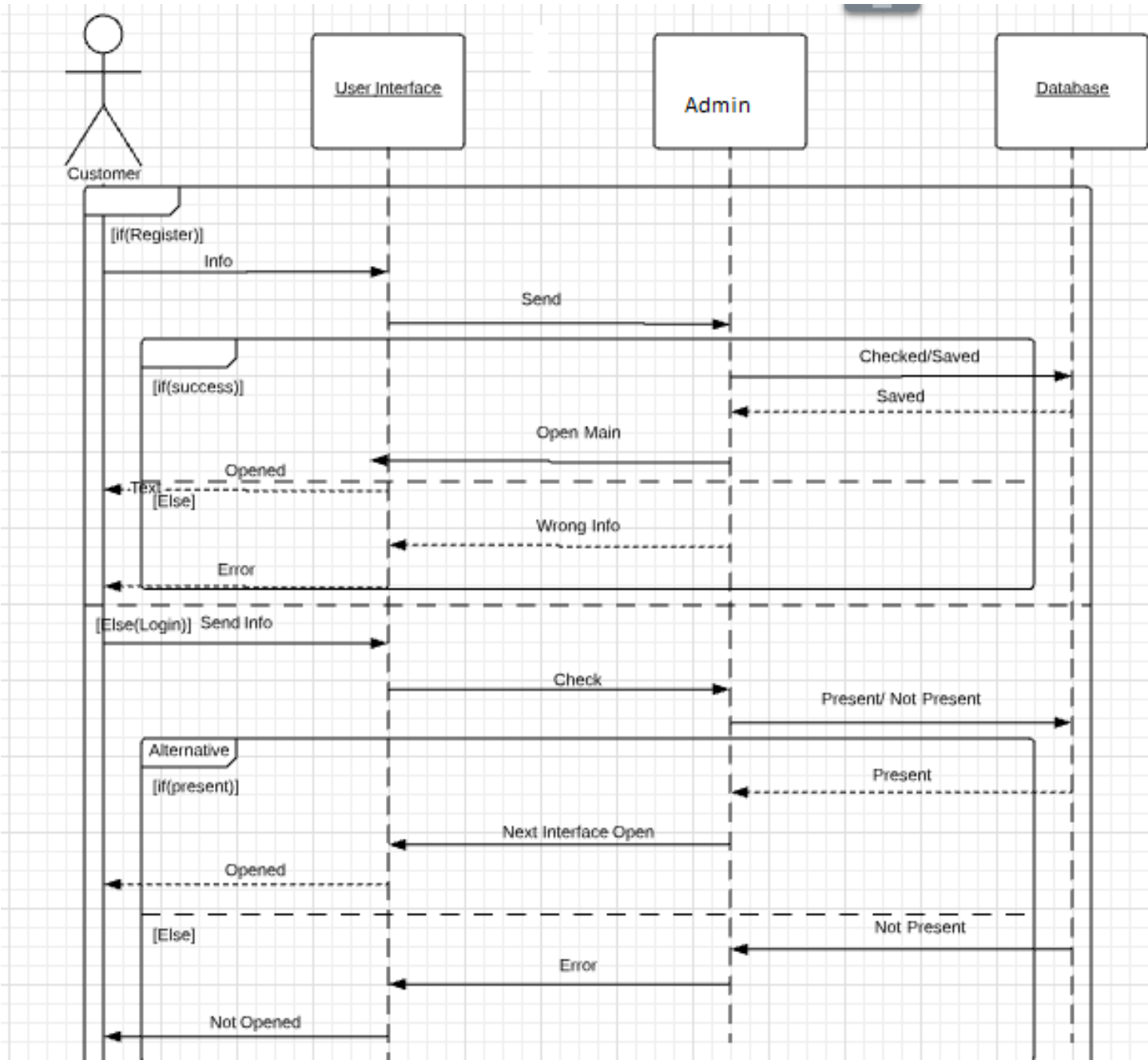


Figure 4.10: Sequence Diagram 1(Login and Registration)

4.5.4.2 Sequence Diagram 2(Book Ride):

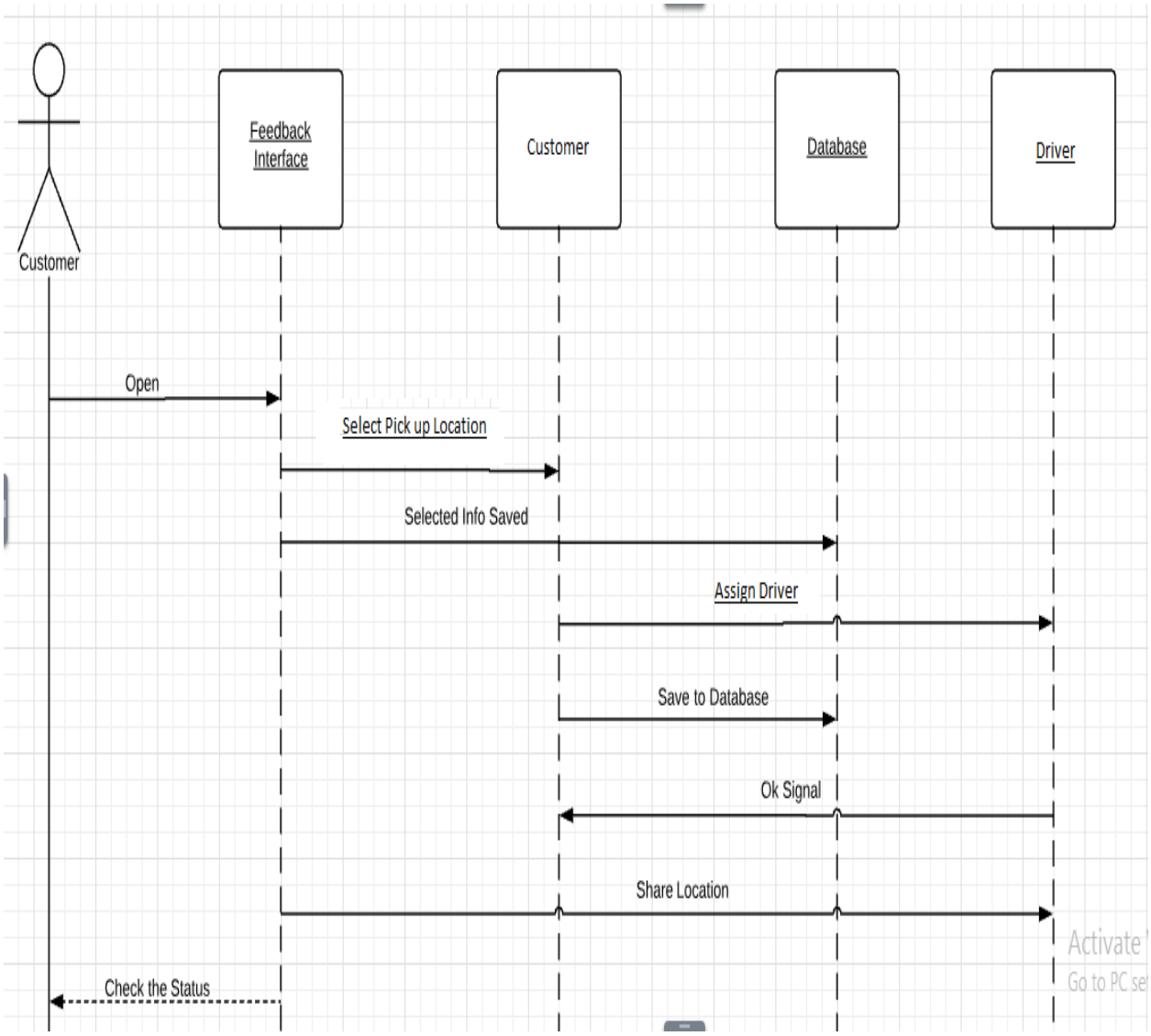


Figure 4.11: Place Order Sequence diagram

4.5.4.3 Sequence Diagram 3(Feedback)

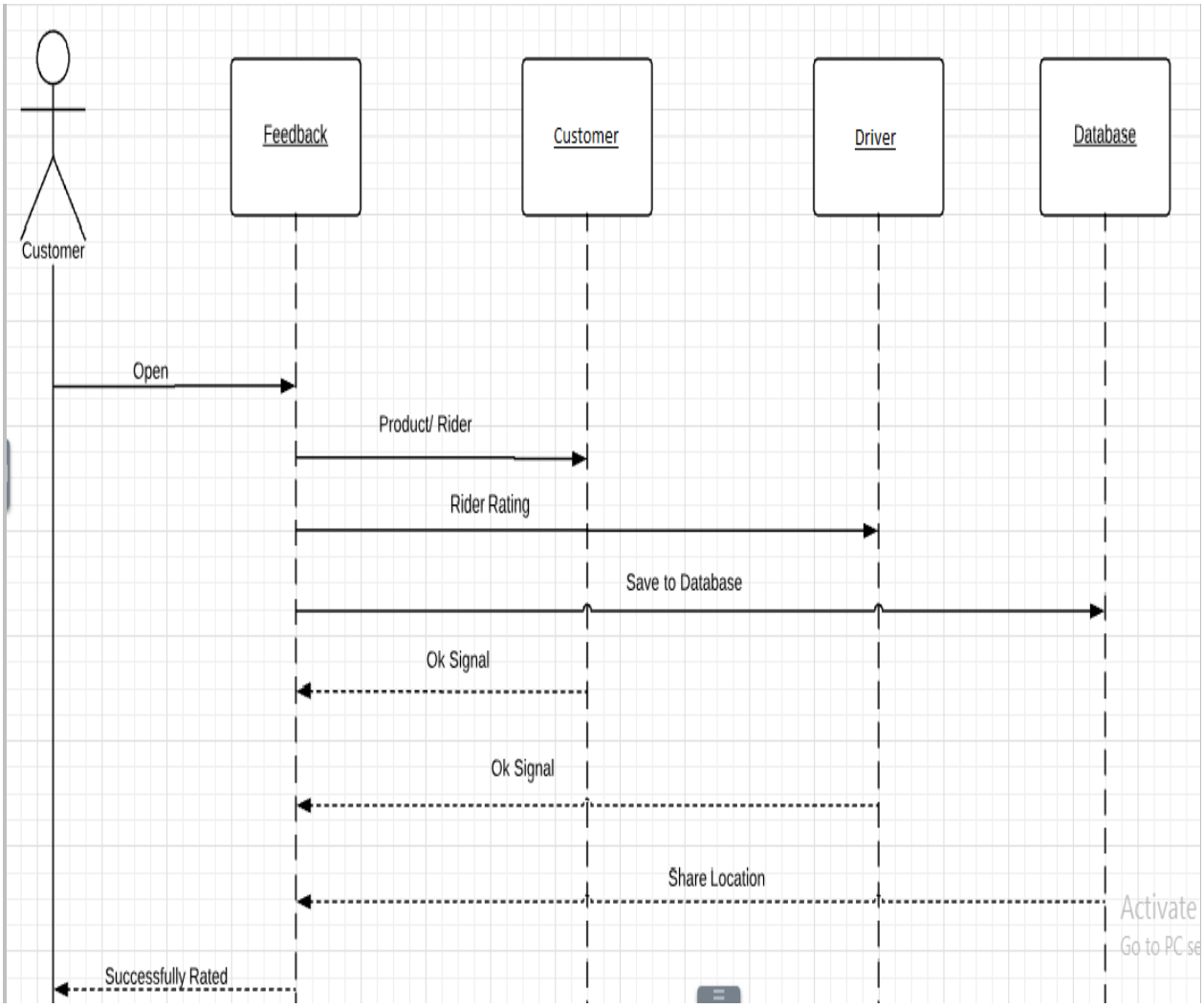


Figure 4.12: Feedback sequence diagram

4.5.4.4 Sequence Diagram 4(Complaint)

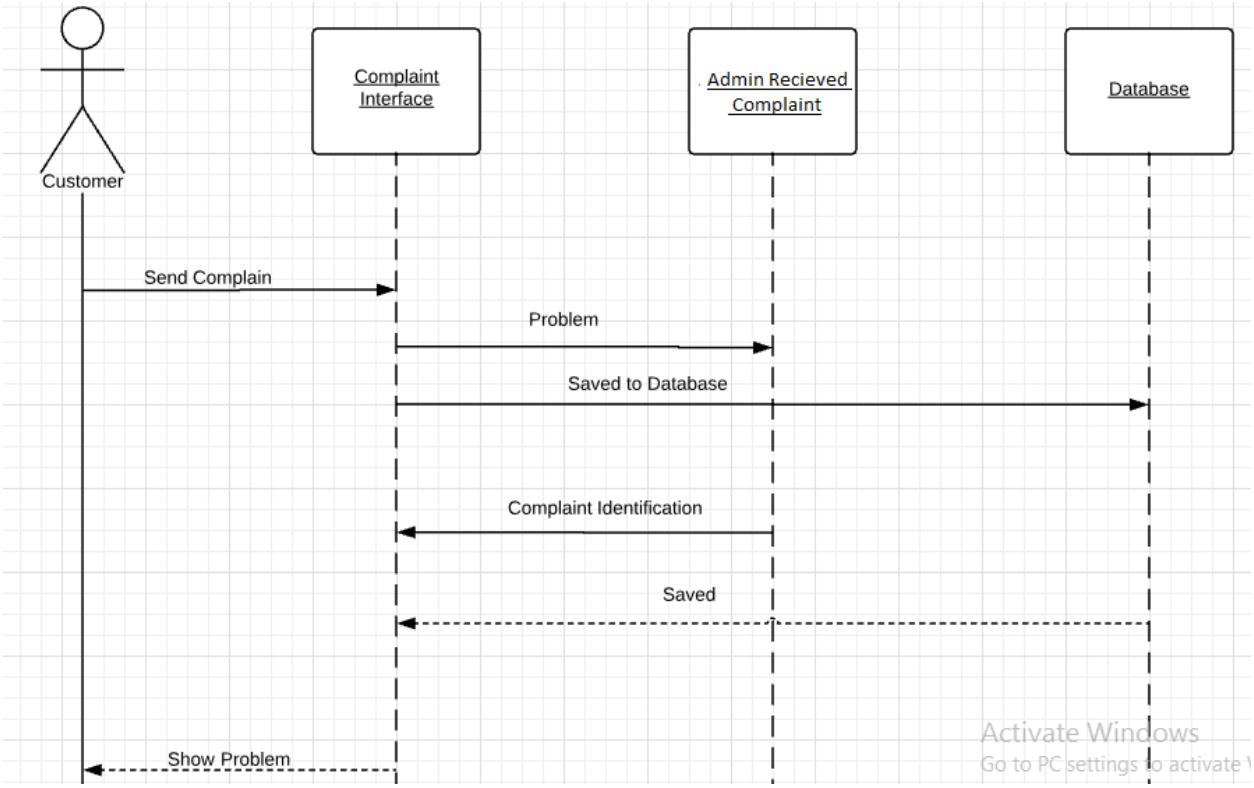


Figure 4.13 : Complaint Sequence Diagram

4.5.5 Implementation View (Class Diagram)

In activity diagram, the dynamic view of the system is shown. All the activities are shown concurrently with their respective start and end states.

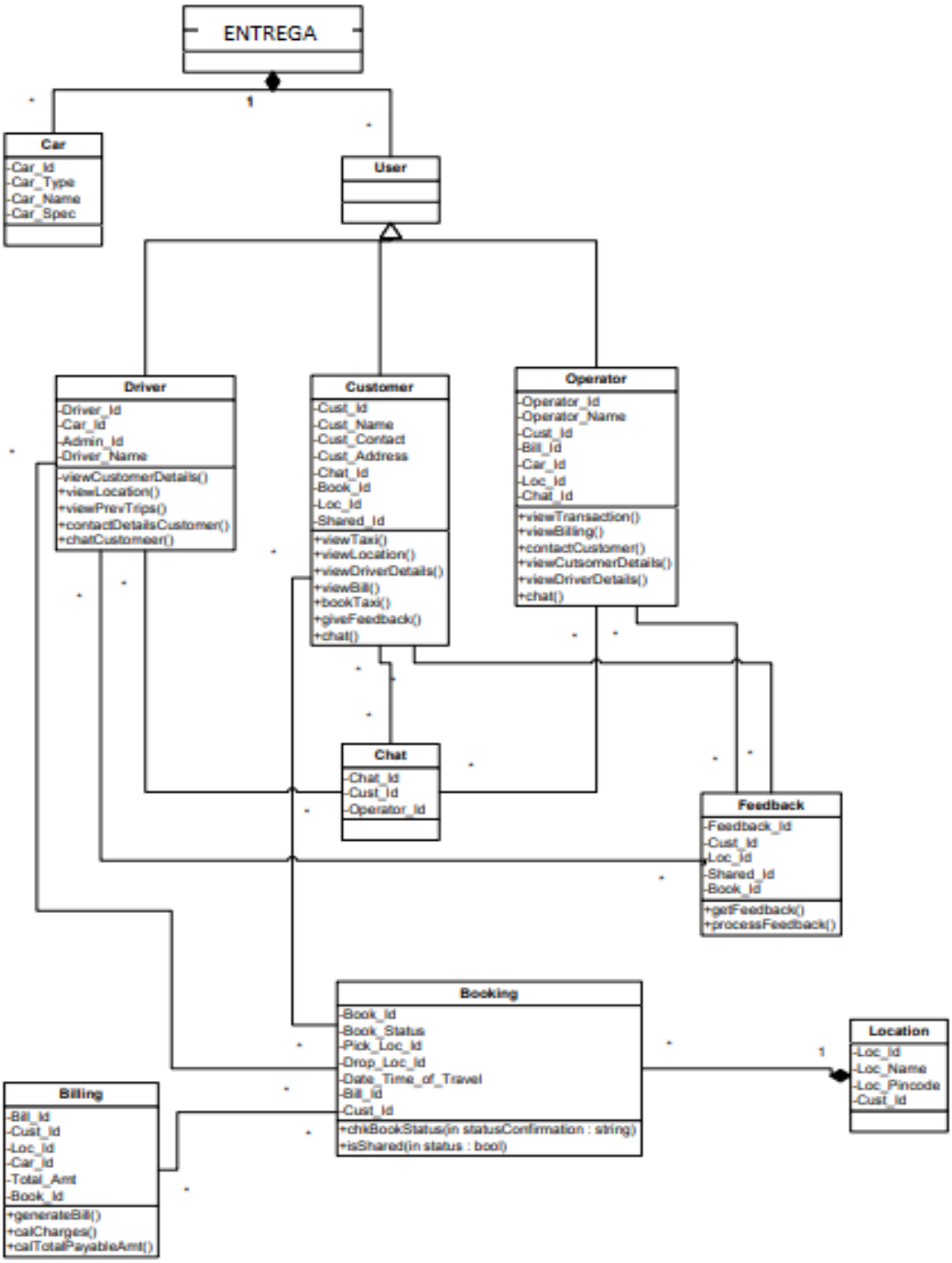


Figure 4.14 : Class Diagram

Classes	Description
Main	This is the main class of the application. It will be executed first in the program. It will first authenticate the user and then shift the control to the user (customer, operator and driver).
User	This class will be used for authentication of the user and it will be called from Main class and it will interact with Database Manger to check the authentic user.
Car	This class will be used for authentication of the driver cars and it will be called from Main class and it will interact with the customer and system and the main class.
Driver	It will interact with Main class whenever the user logs in to perform some function. It will further interact with Database Manager for saving the data into database.
Customer	It will interact with Main class whenever the user logs in to perform some function. It will further interact with Database Manager for saving the data into database.
Operator	It will interact with the customer and driver and finally with the main class. It will search for particular driver and a specific vehicle and

	will a lot it to the user along with the information of the driver and estimated fare along with estimated arrival time
Chat	This class interacts with user and driver and will allow them to interact while driver is on his way to arrive at the pick-up spot.
Feedback	This class interacts with chat, user and driver and allows customer and driver to give feedback in the form of ratings.
Booking	This class is used to book ride. This class is called upon when the user wants to book a ride or cancel a ride which is already booked.
Billing	This class is used for billing and fare purpose. This class will be called whenever the user puts drop of location for the estimate fare. And at the end of the ride to generate bill and send a report to the drivers and customer email and phone via Text message.
Location	This class is called upon when the user puts in the pick-up location and drop of location. It consists of API of Google map and is used to guide the customer and the driver respectively.

Table 4.1: Class Diagram Description

4.5.6 Collaboration Diagram

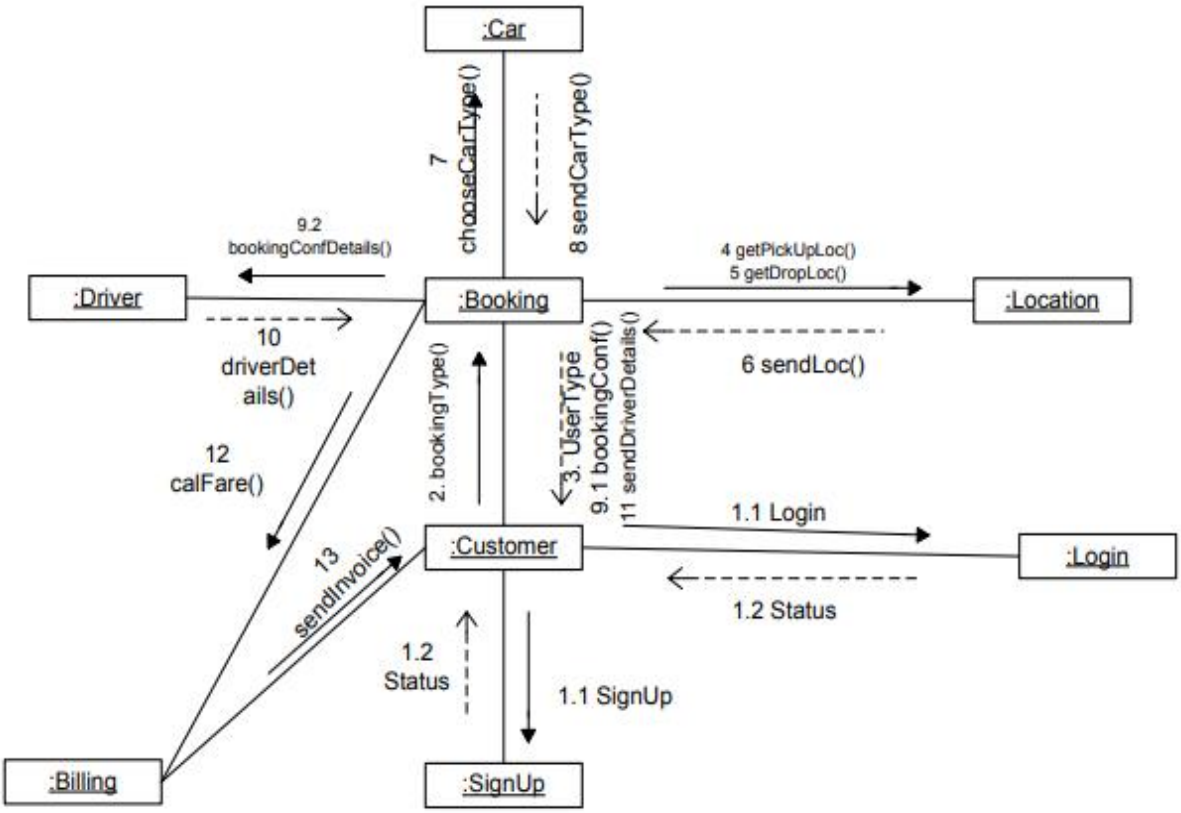


Figure 4.15 : Collaboration Diagram

4.5.7 Dynamic View (Activity Diagram)

In activity diagram, the dynamic view of the system is shown. All the activities are shown concurrently with their respective start and end states.

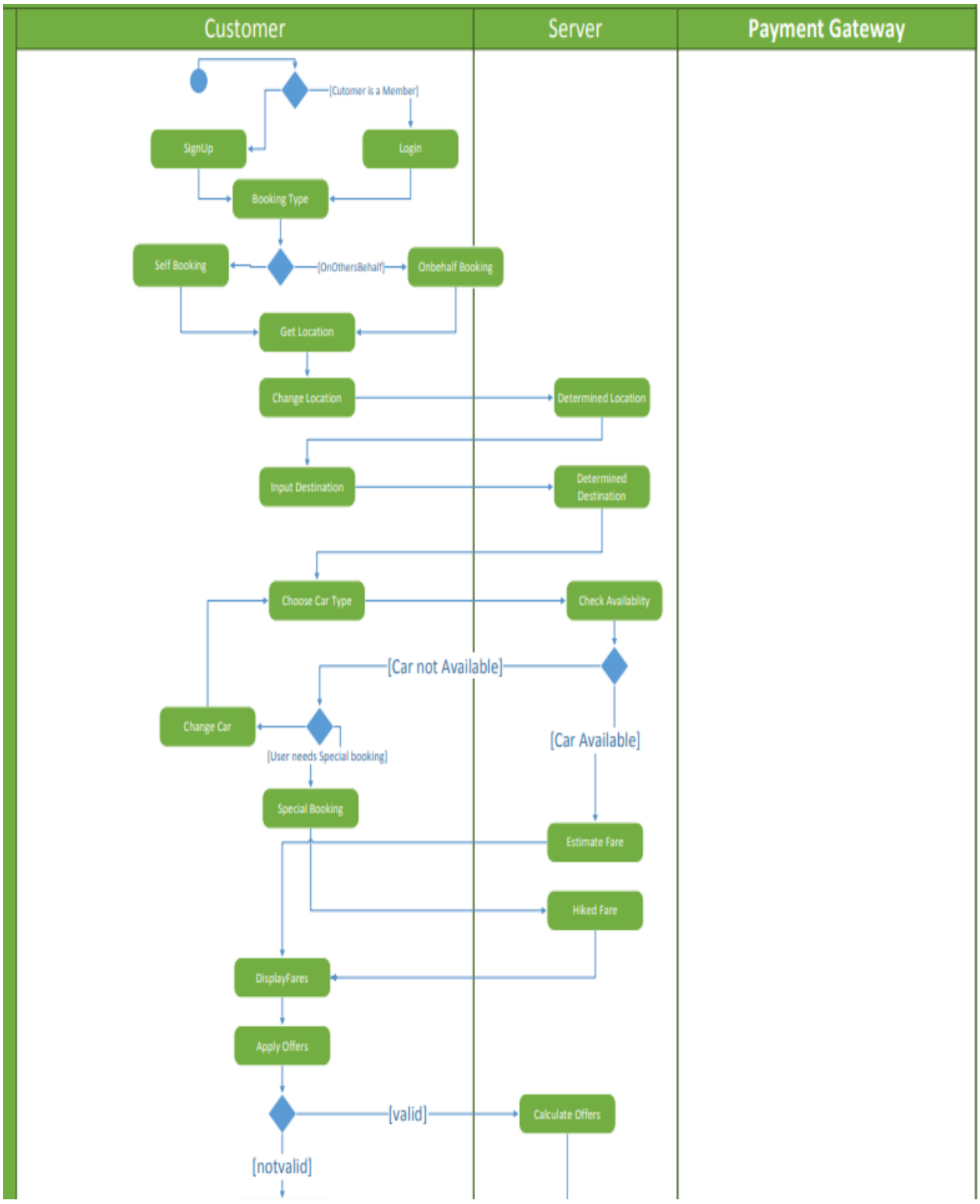


Figure 4.16 : Activity Diagram

4.6 User Interface

4.6.1. Web Portal

4.6.1.1. Log in page

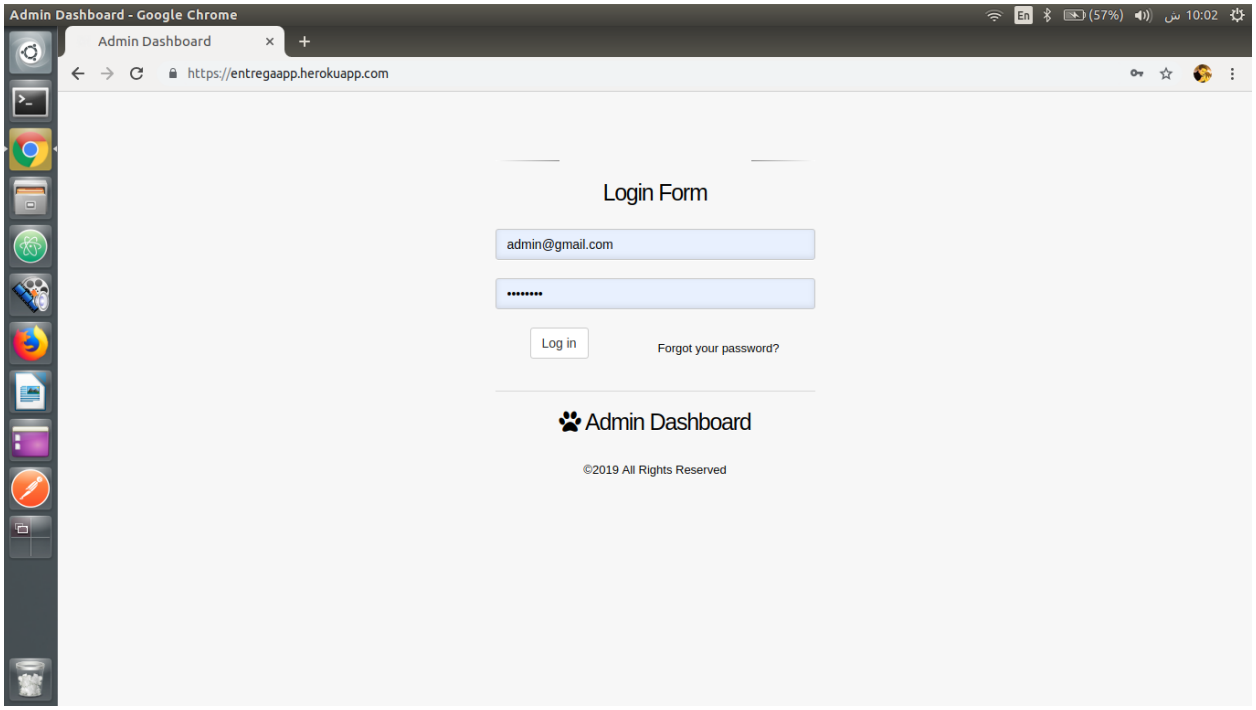


Figure 4.17 : Login page

4.6.1.2. Admin home page

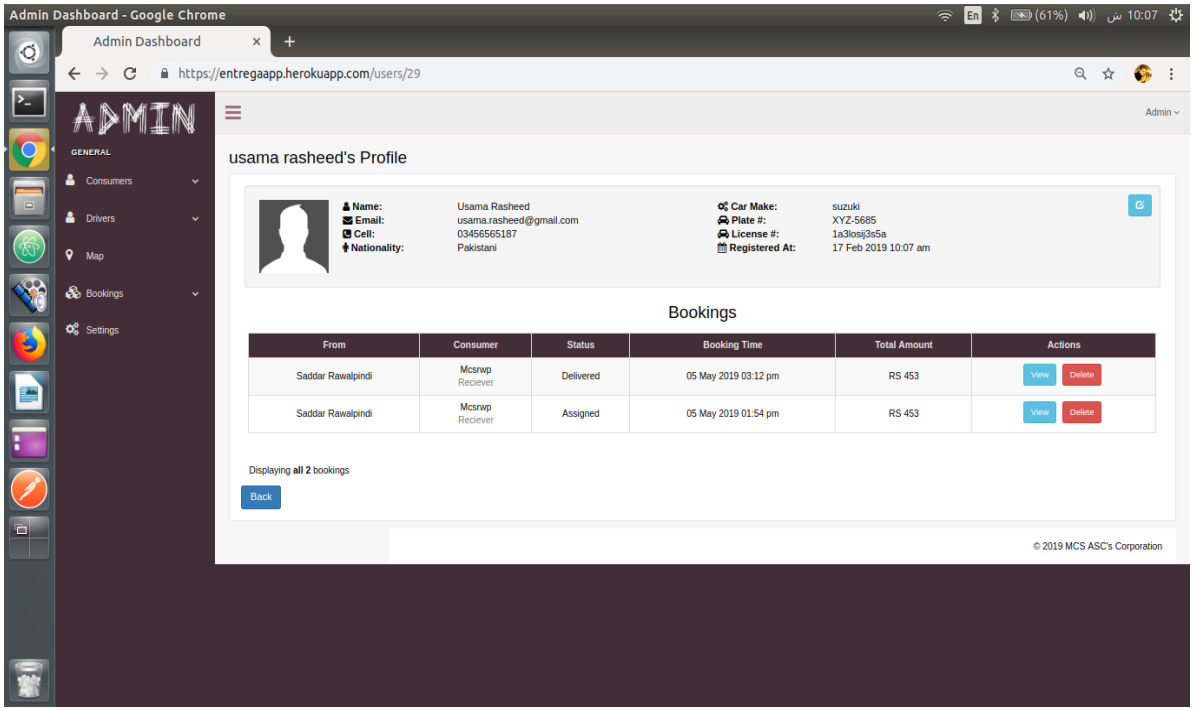


Figure 4.18: Admin homepage

4.6.1.3. Add Administrator

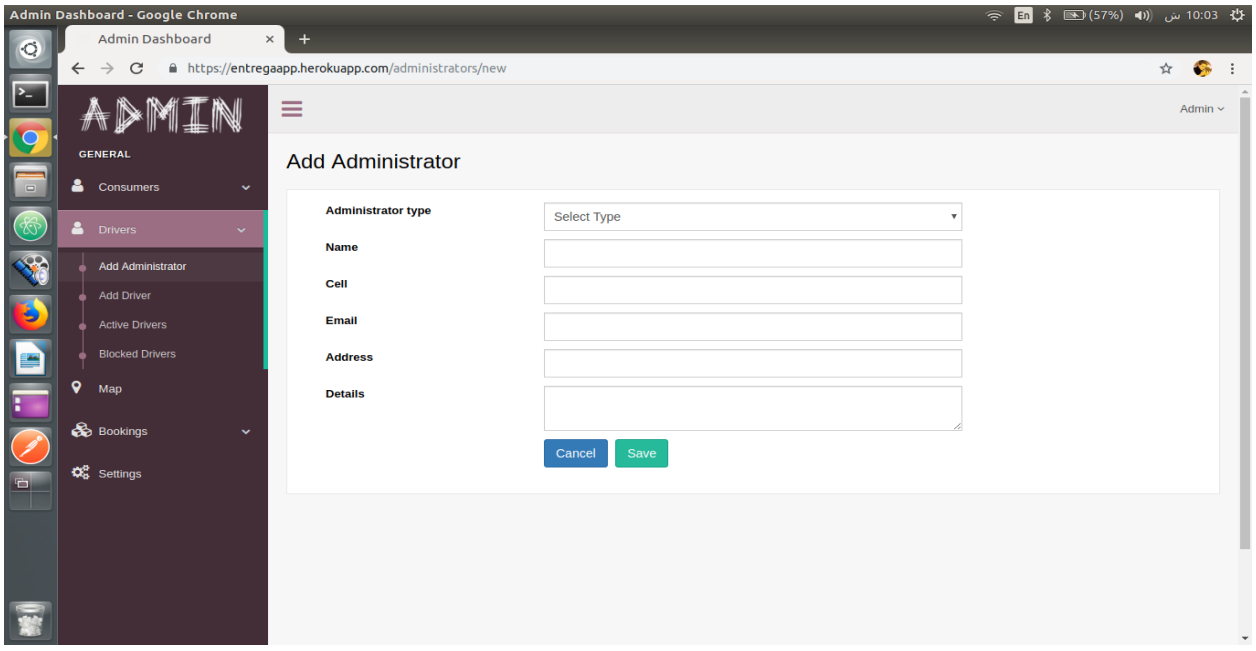


Figure 4.19 : Add administrator

4.6.1.4 Active Customer

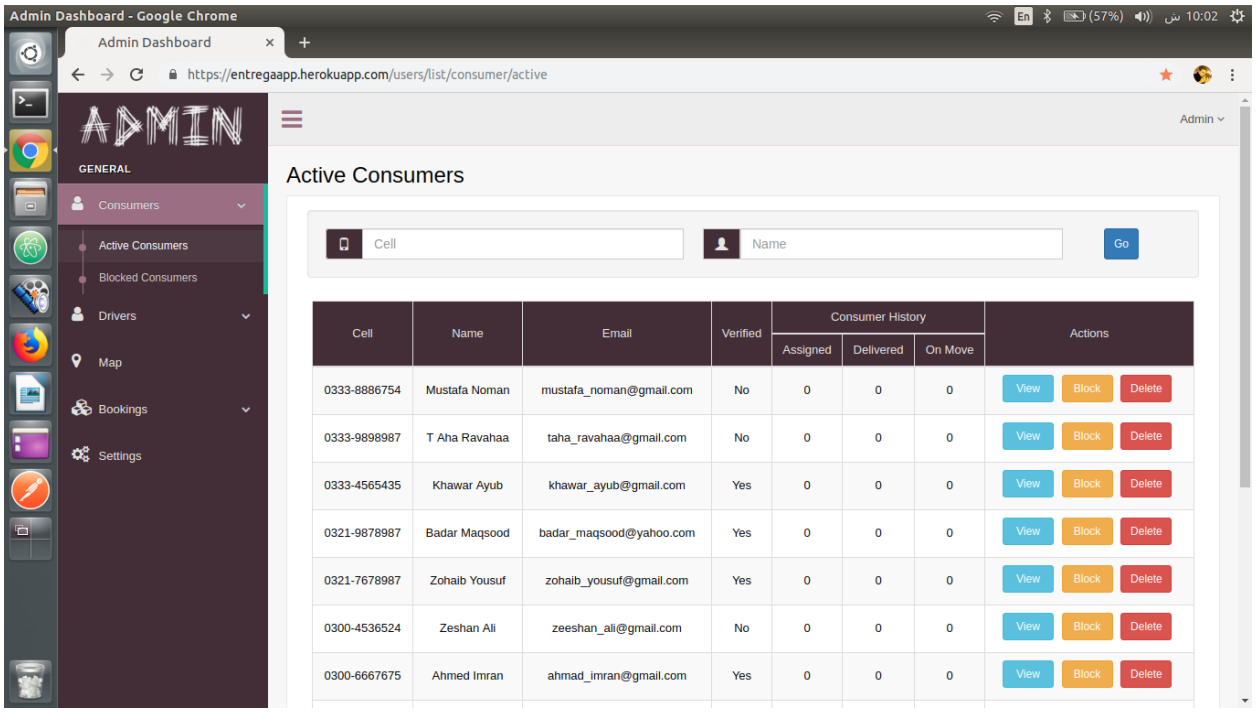


Figure 4.20 : Active Customer

4.6.1.5 Blocked Customer

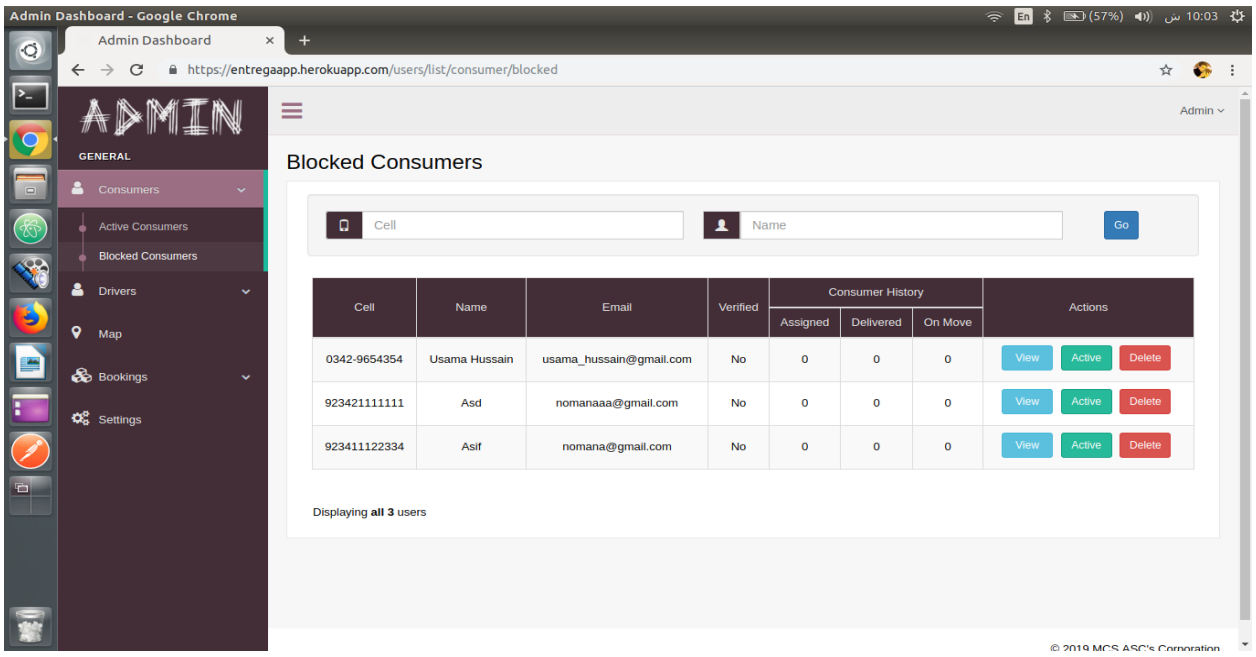


Figure 4.21 : Blocked Customer

4.6.1.6 Add Driver

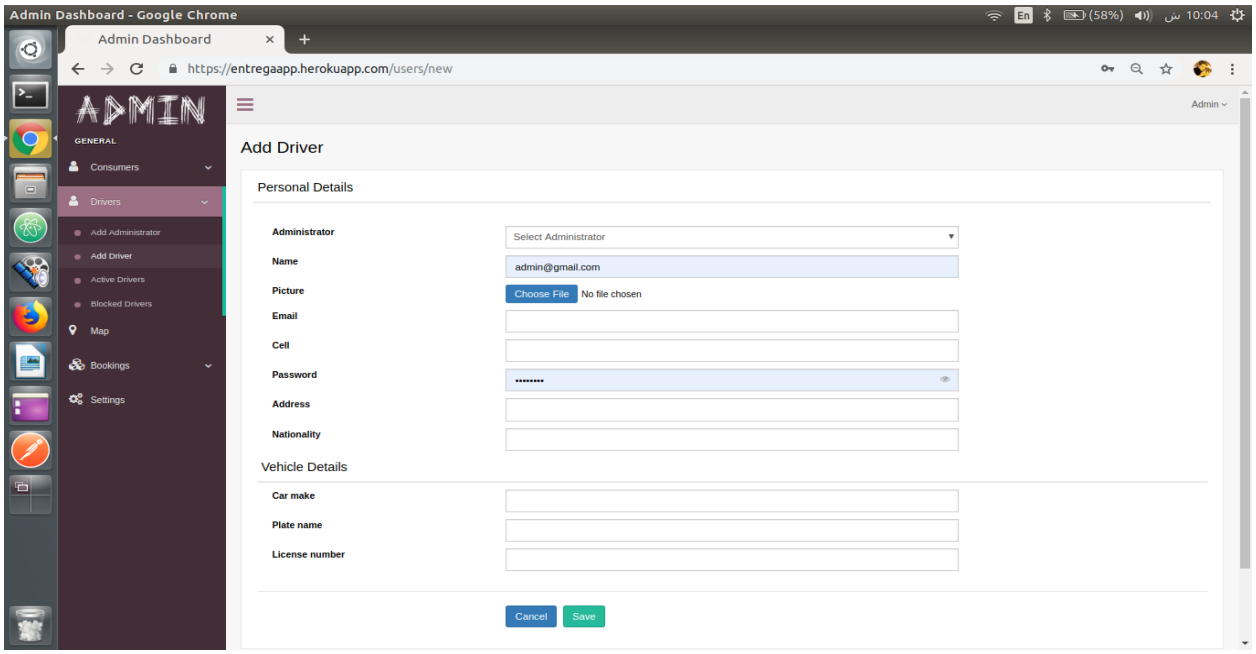


Figure 4.22 : Add Driver

4.6.1.7 Active Driver

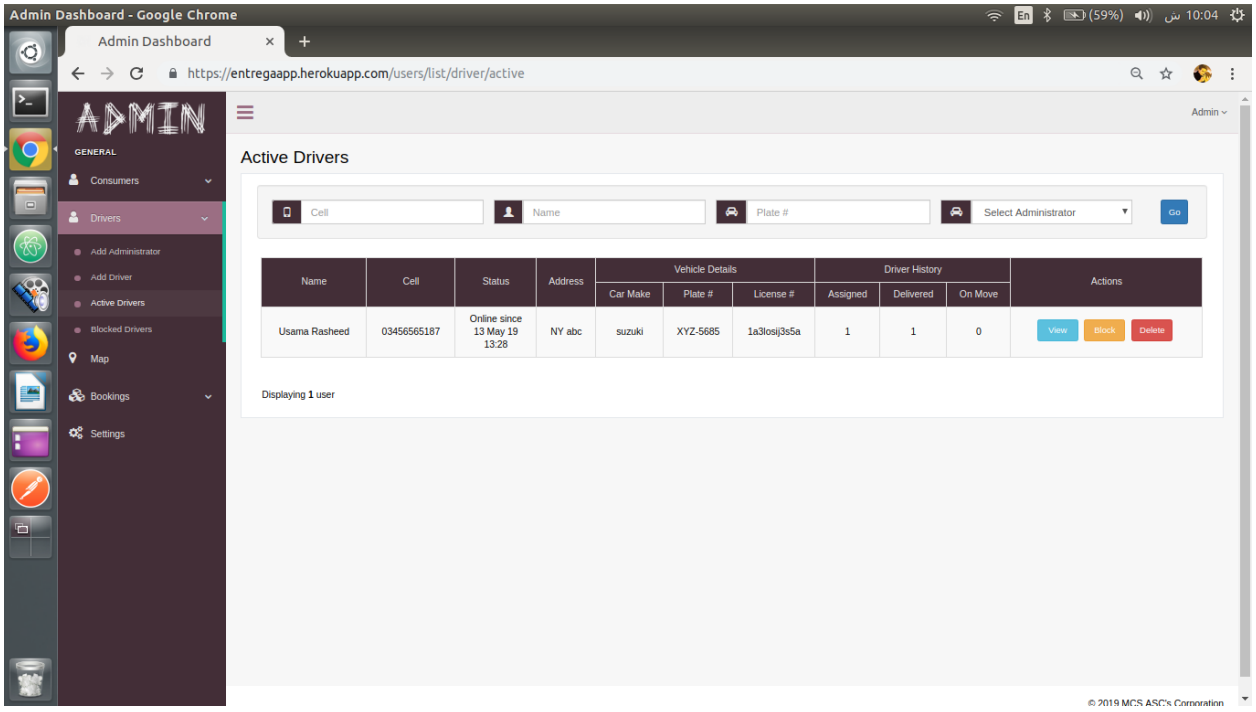


Figure 4.23 : Active Driver

4.6.1.8 Map

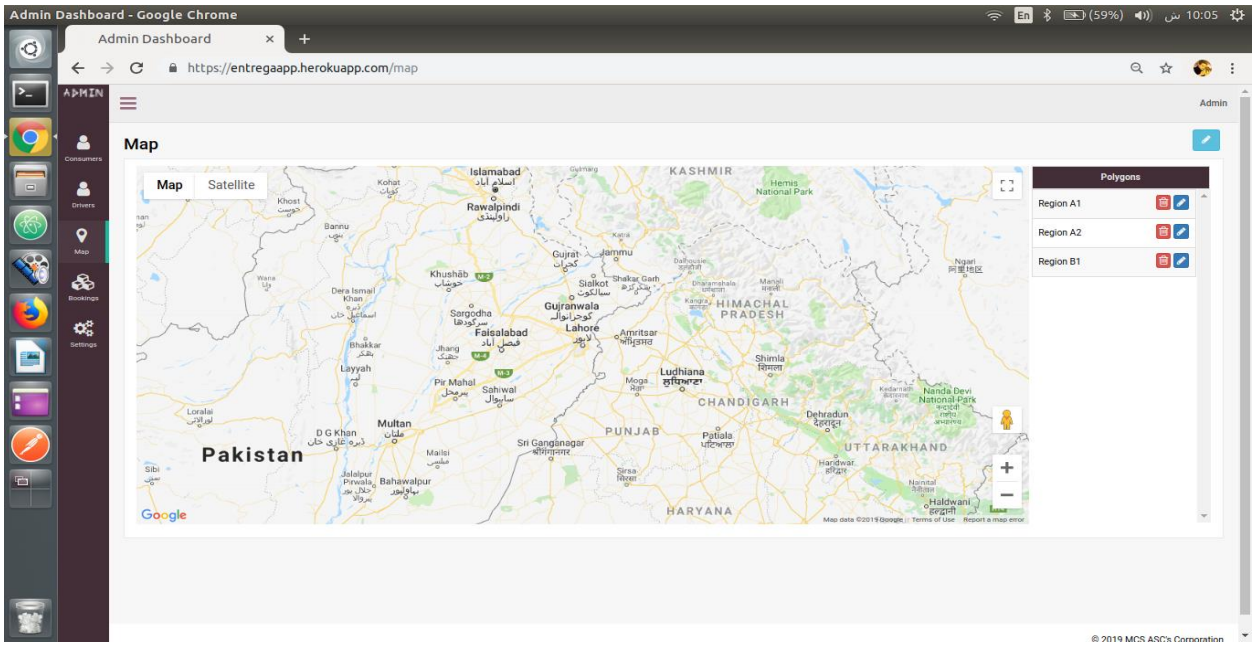


Figure 4.24 : Map

4.6.1.9 Bookings

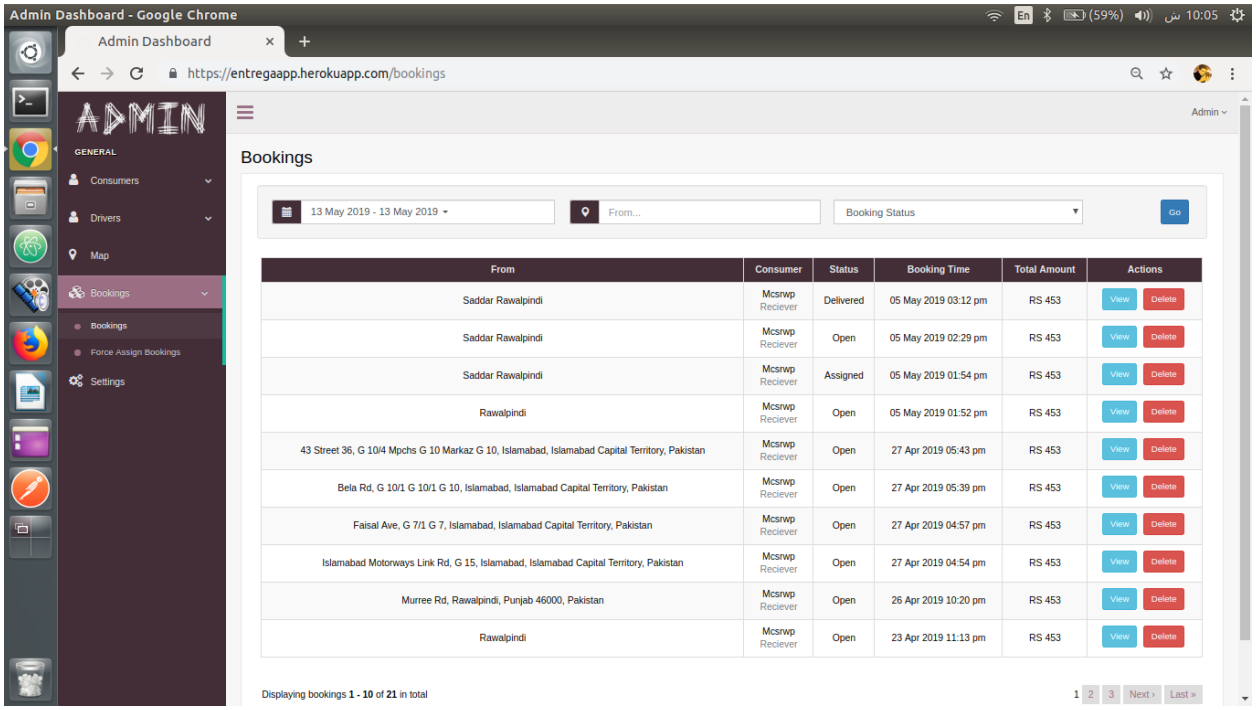


Figure 4.25 : Bookings

4.6.1.10 Force Assign Booking

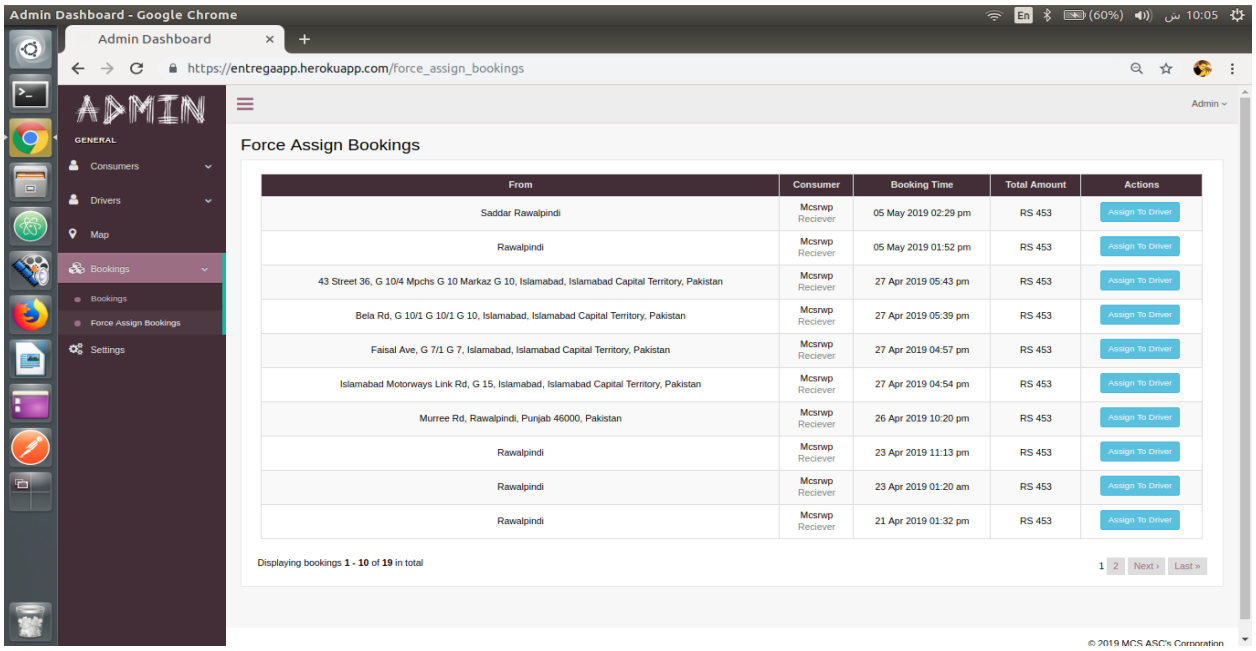


Figure 4.26 : Force Assign Bookings

4.6.1.11 Settings

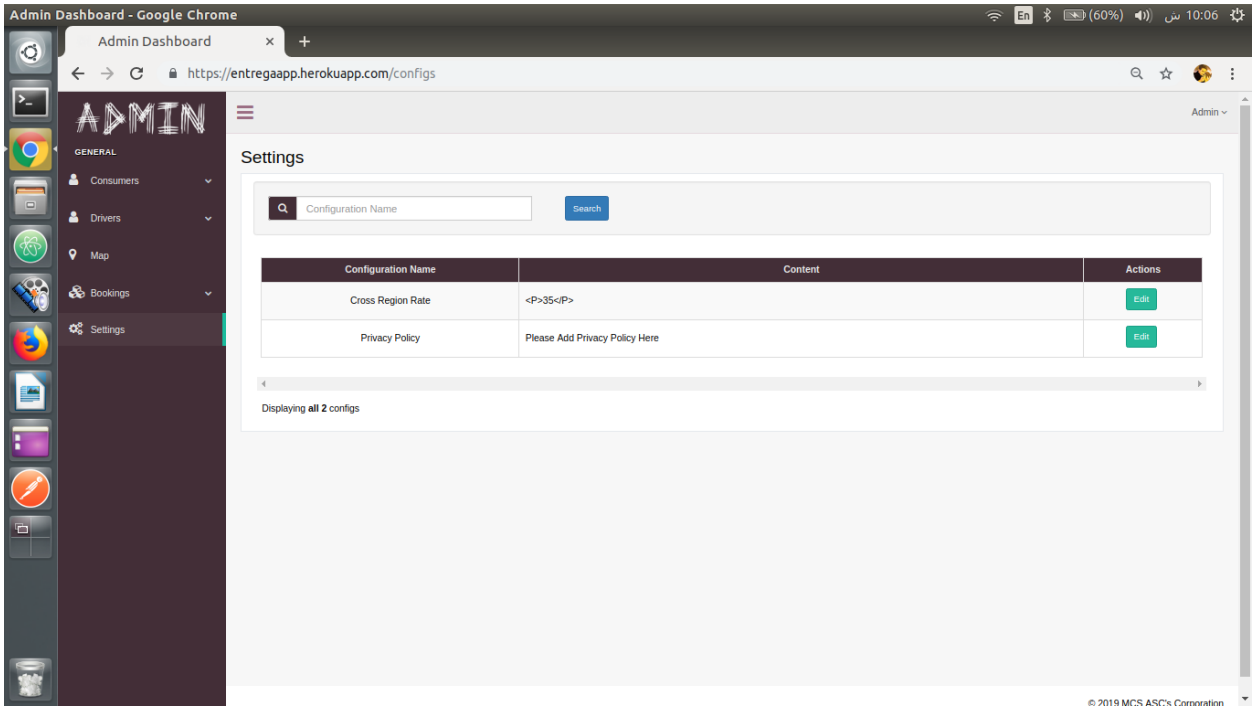


Figure 4.27 : Settings

4.6.2 Android Application

4.6.2.1 Signup Customer

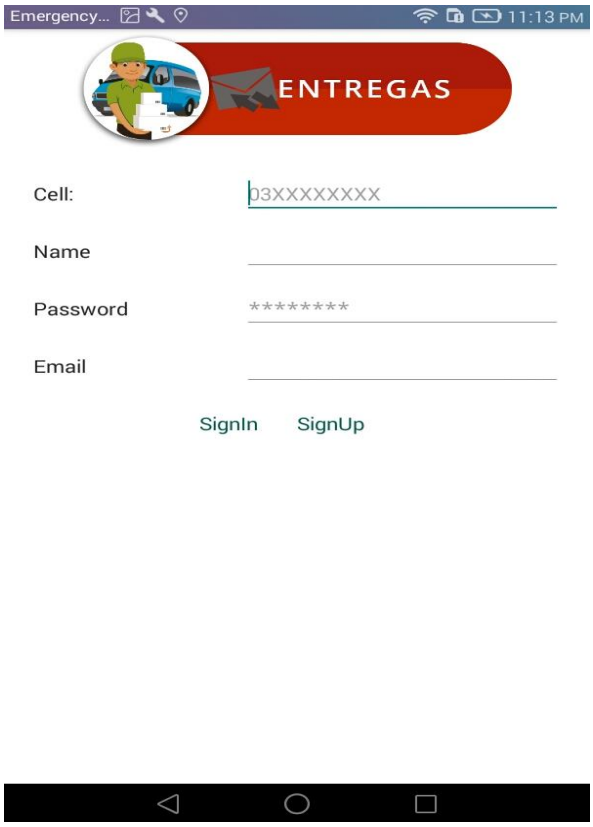


Figure 4.28 : Signup Customer

4.6.2.2 Bookings Customer

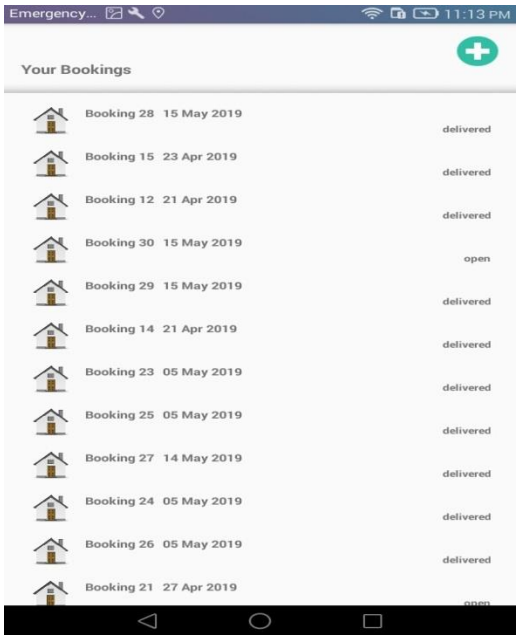


Figure 4.29 : Bookings

4.6.2.3 Details

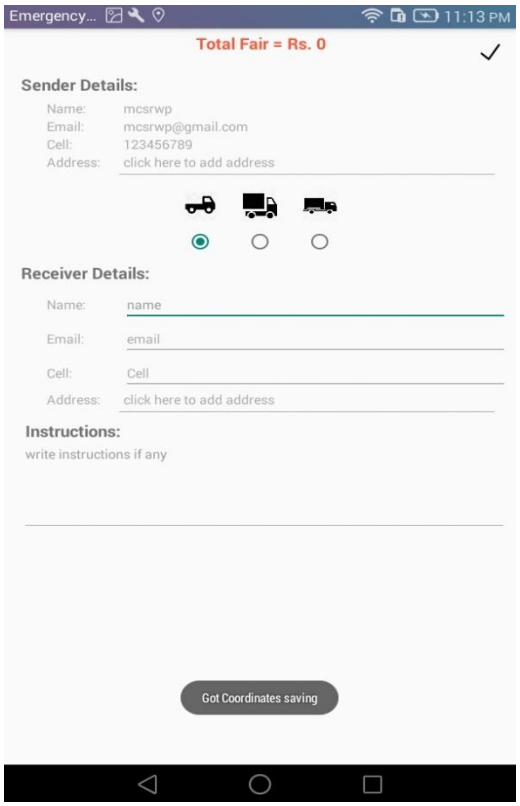


Figure 4.30: Details

4.6.2.4 Map for coordinates

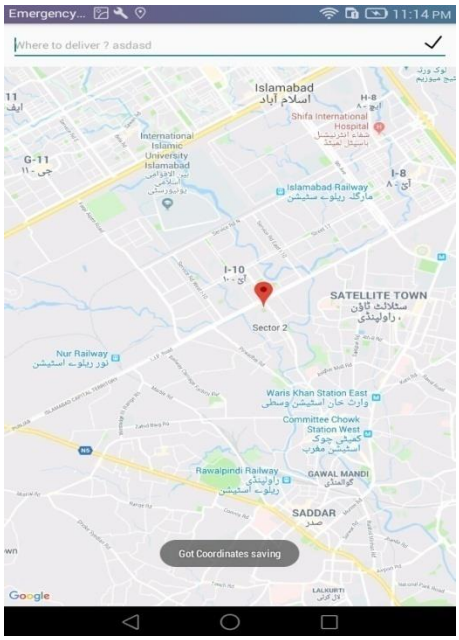


Figure 4.31 : Map Android Application

4.6.2.5 Sender Details

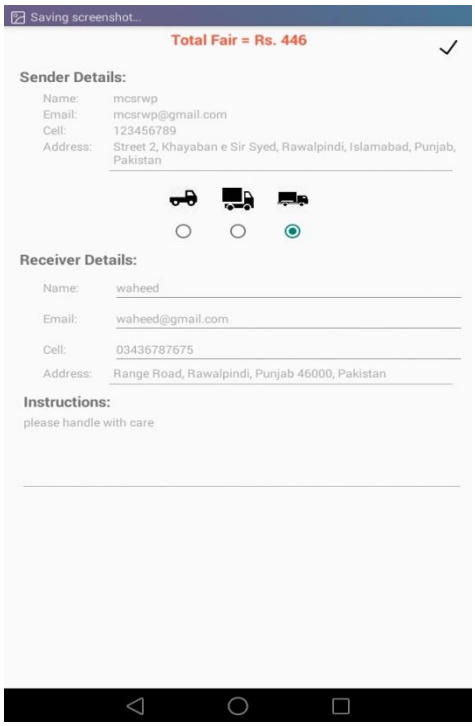


Figure 4.32 : Sender Details

4.6.2.6 Sign In Driver

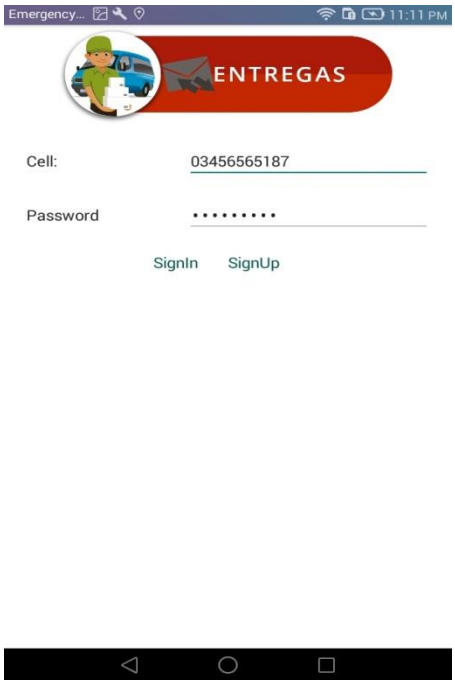


Figure 4.33 : Sign In Driver

4.6.2.7 Booking Details Driver End

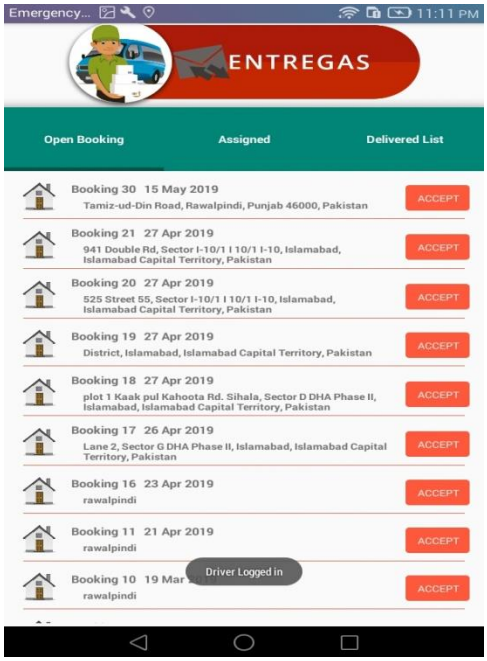


Figure 4.34: Bookings Driver

4.6.2.8 Order Delivered

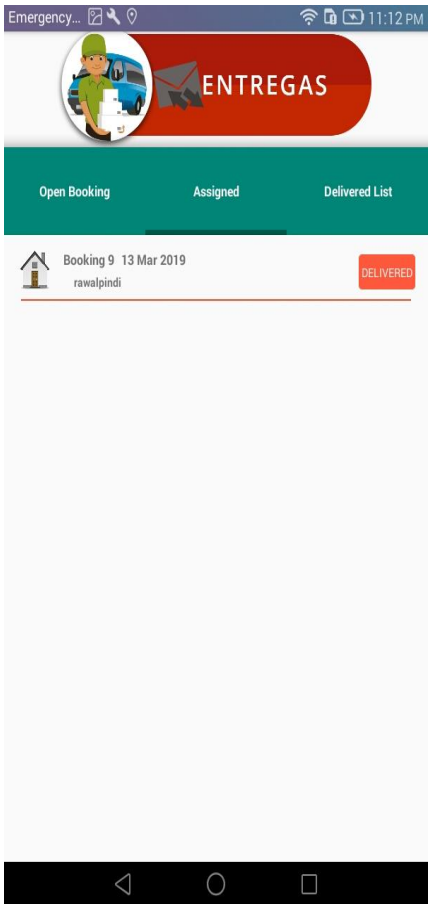


Figure 4.35 : Order Delivered

4.7 Detailed Description of Components

This section describes in detail all the modules of Entrega.

4.7.1 APPLICATION UI COMPONENT

Identification	Name : Application UI Location: Presentation layer of the system architecture.
Type	UI Component
Purpose	The user directly interacts with this component. He/she provides an input for the required action (through this component) and it displays the output respectively. This component fulfills all functional requirements (as specified in SRS document) related to user interaction in the application. Functional requirements are: REQ-1: Application shall be able to notify invalid username and password if it is not found in database. REQ-2: If the invalid username or invalid password is entered, application should not load next screen and generate an error message. REQ-3: If both username and password are valid, application should load next screen. REQ-10**:** The system shall display information screen of the selected drivers.

	REQ-11**:** The system shall display progress table on request of user. REQ-12**:** The application shall be able to calculate the fare or estimated fare of the ride. REQ-13**:** The system shall allow user to log out from the application. REQ-14**:** All subsequent displayed pages shall contain a logout control. REQ-15**:** The system shall display the main log in page after user logs out.
Function	This component has two major functions; take input from the user and display all application screens. The component takes input from the user in form of keystrokes or other touch events, and provide a graphical output to the user.
Subordinates	This component has two subordinates; one is responsible for input, other for the output. The input subordinate satisfies all functional requirements (mentioned in the SRS document) that require user input: Req 1, Req 2, Req 3 and Req 7. While the output subordinate satisfies all functional requirements (mentioned in the SRS document) that provide output: Req 1, Req 2 , Req 3, Req 5, Req 6, Req 11, Req 12 Req 13.
Dependencies	This component 3.1 ‘Application UI’ interacts with the component ‘Process Data’, whenever a user interacts with the application. This component is dependent on the Application UI whereas no component depends upon this component. This component gets and stores values by using function calls like

	getInfo(), *setInfo()* etc.
Interfaces	All user interfaces defined in in section 2 are part of it. The user input and output on screens will be shown using these interfaces. It will provide external interface to component 'Process Data' in form of inputs taken by user. Error Messages: 1. No Internet Connection. 2. Invalid Username or password. 3. No record exist 4. Unable to save (Database problem). 5. Unable to retrieve (Database problem).
Resources	Hardware: Keyboard and touch for enabling the user to interact with the application. It will require a user screen on which application will be displayed. The screens will be run by using internal memory i.e. RAM of the device. Software: WPF and XAML forms will be used for displaying.
Processing	Takes user input in form of keystrokes and other touches and shapes the output according to user intent.
Data	Entered values from user, Information String, Name String, Option selected integer etc.

4.7.2 BACKEND SERVER UI COMPONENT

Identification	Name: Back End Server *Location:* Presentation layer of the system architecture
Type	UI component
Purpose	The user directly interacts with this component. He/she provides an input for the required action (through this component) and it displays the output respectively. This component fulfills all functional requirements (as specified in SRS Document) related to user interaction in the application. Functional requirements are: REQ-1: Web Application should be able to notify invalid username and password if it is not found in database. REQ-2: If the invalid username or invalid password is entered, web portal should not load next screen and generate an error message. REQ-3: If both username and password are valid, web portal should load next screen. REQ-4: A login failure shall redisplay the login method with all fields blank. REQ-5: If both username and password are valid, portal shall load next

	screen. REQ-6: The system shall allow admin to delete any user. REQ-7: The system shall display a dialog box to reconfirm deletion of user when admin requests to delete a user. REQ-8: The system shall delete information of the user from the database. REQ-9: The system shall allow Admin to select User (Driver, Customer) from the list available on his home screen. REQ-10: The system shall display information screen of the selected drivers. REQ-11: The system shall display progress table on request of user. REQ-12: The web application should give the option to record the remarks of the user about the progress of the users (customer and driver) in the database after displaying the progress. REQ-13: The application should give the option of logout on different screens so that users can logout of the application at any time.
Function	This component has two major functions; take input from the user and display all application screens. The component takes input from the user in form of keystrokes or other mouse events, and provide a graphical output to the user.
Subordinates	This component has two subordinates; one is responsible for input, other

	for the output. The input subordinate satisfies all functional requirements (mentioned in the SRS Document) that require user input: Req 1, Req 2 and Req 3. While the output subordinate satisfies all functional requirements (mentioned in the SRS Document) that provide output: Req 4, Req 5 , Req 6, Req 7, Req 8, Req 9, Req 10, Req 12, Req 13.

4.7.3 PROCESS DATA

This component has three sub-components.

4.7.3.1 CUSTOMER MANAGEMENT COMPONENT

Identification	Name: User Management *Location:* Presentation layer of the system architecture
Type	Sub-Component
Purpose	Following functional requirements mentioned in SRS are fulfilled by this sub-component: REQ-1: Application should be able to notify invalid username and password if it is not found in database. REQ-2: If the invalid username or invalid password is entered, application should not load next screen and generate an error message. REQ-3: If both username and password are valid, application should load

	next screen. REQ-4: The application should be able to identify blank required fields for addition and should highlight them. REQ-5**:** The application shall open the corresponding vehicle allotted screens selected from the first menu. REQ-7**:** The application shall be able to calculate the fare or estimated fare of the ride. REQ-14: The application should show the detailed ride history when asked by the customer.
Function	For authentication of user at time of login, it will take input from the user and check its credentials in the database and then validates the user if credentials are matched.
Subordinates	It has two subordinates; to update and view user's data and rides history and other one is to book ride and track it. It fulfills Req 1, Req 2, Req 3, Req 4, Req 7 and Req 14 of the functional requirements as listed in the SRS document.
Dependency	This sub-component 3.3.1 'Customer Management' is dependent on component 'Data Control' and sub-component 'Progress Management'.
Interface	Customer is selected from database with the help of component Data Control and displayed on the screen once right credentials are entered Error Messages: 1. Entrega does not operate in this area.

	2. Invalid Entry. 3. Invalid username/password
Resources	Hardware: RAM and Processor of the system will be utilized. Software: C# core libraries , JAVA and JAVA script
Data	Information string, Selected customer String

4.7.3.2 DRIVER MANAGEMENT UI COMPONENT

Identification	Name: Driver Management *Location:* Presentation layer of the system architecture
Type	Sub-Component
Purpose	Following functional requirements mentioned in SRS are fulfilled by this sub-component: REQ-1: Application shall notify invalid username and password if it is not found in database. REQ-2: If the invalid username or invalid password is entered, application shall not load next screen and generate an error message. REQ-3: The application shall load the next screen when valid and authenticates username and passwords are entered. REQ-4: The application shall allow user to select Vehicle, pickup and drop

	off location. REQ-5: The application shall open the corresponding vehicle allotted screens selected from the first menu. REQ-6: The application shall enable the customer to see credentials of the driver who is allotted the ride. REQ-7: The application shall be able to calculate the fare or estimated fare of the ride. REQ-11: The system shall allow user to log out from the application. REQ-12: All subsequent displayed pages shall contain a logout control. REQ-13: The system shall display the main log in page after user logs out.
Function	For authentication of user at time of login, it will take input from the user and check its credentials in the database and then validates the user if credentials are matched.
Subordinates	It has two subordinates; to update and view driver's data and rides history and other one is to accept ride and track it. It fulfills Req 1, Req 2, Req 3, Req 4, Req 5, Req 6, Req 7 and Req 13 of the functional requirements as listed in the SRS document.
Dependency	This sub-component 3.3.1 'Driver Management' is dependent on component 'Data Control' and sub-component 'Progress Management'.
Interface	Driver is selected from database with the help of component Data Control and displayed on the screen once right credentials are entered

	Error Messages: 1. No driver nearby. 2. Invalid Entry. 3. Invalid username/password
Resources	Hardware: RAM and Processor of the system will be utilized. Software: C# core libraries , JAVA and JAVA script
Data	Information string, Selected driver String

4.7.3.3 PROCESS MANAGEMENT COMPONENT

Identification	Name: Process Management *Location:* Presentation layer of the system architecture
Type	Sub-component
Purpose	Following functional requirements mentioned in SRS are fulfilled by this sub-component: REQ-1: Application shall notify invalid username and password if it is not found in database. REQ-2: If the invalid username or invalid password is entered, application shall not load next screen and generate an error message.

	REQ-3: The application shall load the next screen when valid and authenticates username and passwords are entered. REQ-4: The application shall allow user to select Vehicle, pickup and drop off location. REQ-5: The application shall open the corresponding vehicle allotted screens selected from the first menu. REQ-6: The application shall enable the customer to see credentials of the driver who is allotted the ride. REQ-7: The application shall be able to calculate the fare or estimated fare of the ride. REQ-11: The system shall allow user to log out from the application. REQ-12: All subsequent displayed pages shall contain a logout control. REQ-13: The system shall display the main log in page after user logs out. REQ-14: The application should show the detailed ride history when asked by the customer.
Function	This sub-component will control all the progress functionality of the customer and driver and also add the remarks of the Users along with the progress.
Subordinates	It has three subordinates, update data, retrieve data and calculate data.

	It fulfills Req 1, Req 2, Req 3, Req 4, Req 5, Req 6, Req 7, Req 13 and Req 14 of the functional requirements as listed in the SRS document.
Dependency	This sub-component 'Process Management" is dependent on component 'Data Control' and sub-component 'Customer management' and 'Driver Management' .
Interface	Saves data to the database with the help of component 'Data Control'. Error Messages: 1. Cannot save remarks (Database Problem) 2. Cannot fetch any progress. (Database Problem) 3. No records exist.
Resources	Hardware: RAM and Processor of the system will be utilized. Software: C# core libraries , JAVA and JAVA script.
Data	Information string, Selected driver String
Processing	Communicates with other components and calculates percentage improvement and daily progress using formulas and algorithms (learning curve etc.).

4.7.4 DATACONTROL COMPONENT

Identification	Name: Data Control

	Location: Presentation layer of the system architecture
Type	Database Component
Purpose	Following functional requirement mentioned in SRS are fulfilled by this sub-component: REQ-1: Application shall notify invalid username and password if it is not found in database. REQ-2: If the invalid username or invalid password is entered, application shall not load next screen and generate an error message. REQ-3: The application shall load the next screen when valid and authenticates username and passwords are entered. REQ-5: The application shall open the corresponding vehicle allotted screens selected from the first menu. REQ-6: The application shall enable the customer to see credentials of the driver who is allotted the ride. REQ-7: The application shall be able to calculate the fare or estimated fare of the ride. REQ-11: The system shall allow user to log out from the application. REQ-12: All subsequent displayed pages shall contain a logout control. REQ-13: The system shall display the main log in page after user logs out. REQ-14: The application should show the detailed ride history when asked by

	the customer.
Function	Function of this component is to handle the database transactions i.e. add, update, delete and select information from the database.
Subordinates	It has 2 subordinates, set and get. 'Set' does the modifying part of the database and it will fulfill the above mentioned Functional requirements of the Functional Requirements mentioned in SRS document
Interfaces	SQL database server in which all the data will be saved. It provides external interface to component 'Process Data' in form of service of data management this component offers to it.
Resources	Hardware: MySQL on Azure, RAM and Processor of the system will be utilized. Software: SQLYog , SQL queries and C# core libraries.
Processing	Transaction is performed in order to retrieve data into the database. The component receives a query in form of an input strings from other components. The query is then executed and transaction is performed either to retrieve data from the database or update it.
Data	Information String, AuthenticalBool, user intetc .

4.8 Reuse and Relationships

Entrega is based on Careem and Uber cab service and the difference between Entrega and other similar application is that Entrega is specifically for loading vehicles and such system has not

been developed anywhere in Pakistan and 90% of the outer world. It can be evolved into a bigger and more complex system with more features and functionality. It can also be used as a big income project by launching it in industry all around the World and extending the scope and outcome as well as results of the product. There are a lot of application which are specifically designed for the Taxi booking or cab Booking like UBER as well as the algorithm are based on the same functionalities based on which the search and the booking can be implemented as well as the structure of the system can be more optimized if there is the proper workflow of execution to have the structural and the functional domain models of the analytics as well as the information about the system is based on the completion of the core functional analysis of the navigation based application which provides the complete and secure services to the end users.

4.9 Design Decisions and tradeoffs

Applications like careem, Uber usually don't have a typical application design pattern. They need to run as fast as possible or locked in sync with the refresh of the display. Using an event based model doesn't work well for this type of development since it needs to grab the frame of data when it wants, regardless of what application is doing and if it isn't there, it'll catch it next time around. It cannot block the thread that does this update/query cycle.

So we are using MVC (Model View Controller) framework. MVC frameworks are libraries that can be included alongside JavaScript to provide a layer of abstraction on top of the core language. Their goal is to help structure the code-base and separate the concerns of an application into three parts:

1. **Model** - Represents the data of the application. This matches up with the type of data a web application is dealing with, such as a user, video, picture or comment. Changes made to the model notify any subscribed parties within the application.
2. **View** - The user interface of the application. Most frameworks treat views as a thin adapter that sits just on top of the DOM. The view observes a model and updates itself should it change in any way.
3. **Controller** - Used to handle any form of input such as clicks or browser events. It's the controller's job to update the model when necessary (i.e. if a user changes their name).

Not all frameworks follow the MVC pattern. You may see some frameworks utilize a variation of the MVC pattern such as MVVM or MVP.

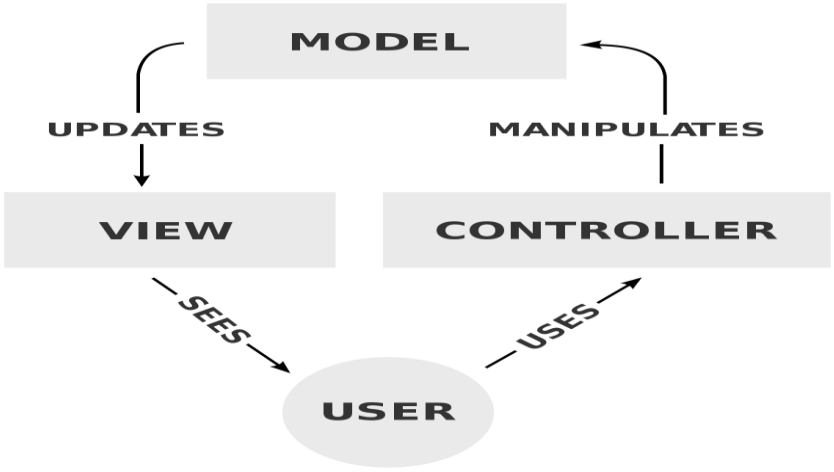


Figure 4.36: MVC

MVC facilitates a clear separation of the development of graphical user interface (View) from the development of the application logic (Model). The Model is a value converter i.e.; responsible for exposing the data objects from the model in such a way that those objects are easily managed and consumed. In this way, the Model is more model than View, handles most of it not all of the view's display logic. It also implement a mediator pattern) organizing access to the back-end logic around set of use cases supported by the view.

Chapter 5

5. SYSTEM IMPLEMENTATION

Entrega is aandroid application which is used to provide the customers with loading vehicles service. It allows the user to easily make a ride and input the dimension of the product to be transported. It allows driver to get location of the customer and earn more in less time and become financially strong. All the record of the user and the driver is stored in back-end server which is made on ROR frame work and is implemented on the basis of MVC framework.

5.1 PSEUDO CODE FOR COMPONENT

5.1.1 Application UI

```
if Login is Successful
Begin
Show Welcome_Message
Show Add_Button
Show Customer(user or Driver)_Table
End
Else
Print ‘Invalid Username/Password’
```

5.1.2 Ride Booking Management

```
begin
Start_BookingRide
Start searching ride(search_location)
Allot_Ride(Driver_Information)
Fare_ride(cal_fare())
```

Remark_user(customer or driver)(Remarks())

End_ride(Finish_ride)

End

5.1.3 User (customer and driver) Management

begin

Show user list

if user selects add user option then ask for user input

if user selects a user then open user options

if user selects edit option then ask for user input

if user selects delete option then delete user from database

if user selects rides then show rides menu

end

5.1.4 Progress Management

begin

retrieve progress from database upon user request

store progress into database

end

5.1.5 User Management

begin

Show user list

if user selects add user option then ask for user input

if user selects a user then open user options

if user selects edit option then ask for user input

if user selects delete option then delete patient from database

```
if user selects exercise then show exercise menu

end
```

5.1.6 Data Control

```
begin

execute DML statements like add, delete, edit, update, select etc.

end
```

Chapter 6

6. Analysis and Evaluation

6.1 Introduction

This test plan document describes the appropriate strategies, process and methodologies used to plan, execute and manage testing of “Entrega". The test plan will ensure that Entrega meets the customer requirements at an accredited level.

Manual Testing will be followed which includes testing a software manually, i.e. without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. Each unit will be tested separately and then will be integrated with other units. Therefore, unit testing and integration testing will be followed. For each unit, black box testing is done and for combined units acceptance testing is done.

The test scope includes the testing of all functional, application performance and use cases requirements listed in the requirement document.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed.

This document includes the plan, scope, approach and procedure the testing of snap assistant. The pass/fail criteria of the test items are also defined. The document tracks the necessary information required to effectively define the approach to be used in the testing of the product.

6.2 Approach

Acceptance test will be executed based on this acceptance test plan. And after all test cases are executed, a test report will be summarized to show the quality of Snap Assistant. Following test approaches will be used in test execution:

1. **Unit test:**Developers are responsible for unit testing. The implementation of each module and individual component will be verified separately.

2. **Integration test:** After the unit test is passed above the defined quality threshold, testers will execute the integration test cases. After all the modules are integrated, it is crucial to test the product as a black-box.

3. **Positive and negative testing design technique**: This approach will be combined with unit test and integration test. Test cases are designed in obvious scenarios, which ensure that all functional requirements are satisfied. Different test cases will also be covered to show how the system reacts with invalid operations.

6.3 Features to be tested

Following features are tested:

1. Signup web portal.
2. Authentication of users (driver and customer).
3. Edit/view profiles of users (driver and customer).
4. Add users (customer and driver).
5. Delete driver.
6. Delete customer.
7. Signup android application.
8. Signing In of users android application.
9. Location tracking.
10. Dimension input and vehicle selection.
11. Ride booking and driver allocation.
12. Track ride (Admin).
13. Assign ride to driver.
14. Admin view driver's progress.
15. Feedback.
16. Update user's information.
17. Send notification.

18. Read notification.
19. Integration testing.

6.4 Item Pass/Fail Criteria

Details of the test cases are specified in section test deliverables. Following the principles outlined below, a test item would be judged as pass or fail.

1. Preconditions are met
2. Inputs are carried out as specified
3. The result works as what specified in output =>Pass
4. The system doesn't work or not the same as output specification =>Fail.

6.5 Test Items

1. Develop test cases.
2. Execute tests based on the developed test cases for the software.
3. Report defects from the executed test cases if any.
4. Provide complete test report.
5. Incorporate or manage changes later in the stage of the project development.

6.6 Test Deliverables

Following are the Deliverables as per this Plan:

1. Test cases
2. Output from tools

6.7 Suspension Criteria and Resumption Requirements

Any bugs found can be fixed by developers quickly and no need to start the testing process from the beginning. However, when major bugs will block some test cases as they are interdependent and the testing has to be paused. The test will restart from the very beginning until the major error is solved.

6.8 Staffing and Training Needs

1. Basics knowledge of testing strategies and techniques is needed for the testing of the project.

2. Techniques such as Black Box testing, integration testing should be known to developers.

3. All the developers will be testing each other's work and will be actively participating in the development and testing of the project simultaneously.

6. 9 Schedule

6.9.1 Important Dates

1. Unit testing and integration testing will be finished by 1st May, 2019.

2. Acceptance Testing will be performed right after the Development process completes that is in the middle of May.

6.10 Risks and contingencies

6.10.1 Schedule Risk

1.The project might get behind schedule so in order to complete the project in time we will need to increase the hours/day that the project is being worked on.

6.10.2 Operational Risks

Operational risks will be eliminated by Scheduling daily meetings and regular deadlines to meet the goals of the project as well as provide proper communication within the group.

6.10.3 Technical risks:

1. Technical risks will be eliminated by keeping the once defined requirements constant.

6.10.4 Programmatic Risks:

1. In case of a programmatic risk the scope of the project will be limited in order to stay inside the constraints of the project.

6.11 Test Cases

6.11.1 Unit and Component level Testing

Following are the Test Cases:

Test Case Number	01
Test Case Name	Signup web portal
Description	All the users are able to fill a form for signup
Testing Technique	Component testing, Black Box Testing
Preconditions	Web portal should be open
Input Values	Enter name, age, address, contact no and other fields required and click “signup”
Steps	• Open the web portal • Main Screen is open. • Enter your details • Click on “signup” button.
Expected output	User should be registered as a legitimate user if the required fields are entered correctly.
Actual output	User is registered successfully.
Status	Test case passed successfully.

Table 6.1: Signup Web Portal

Test Case Number	02
Test Case Name	Authentication of users (driver And customer)
Description	Admin will authenticate users (driver And customer)
Testing Technique	Component testing, Black Box Testing
Preconditions	Web portal should be open and admin should be successfully logged in.
Input Values	Enter user’s name, role , email and password
Steps	• Open the web portal. • Login as admin. • Enter user details • Click on the ‘Authenticate’ button.
Expected output	User should be added to the database.
Actual output	User is successfully authenticated
Status	Test case passed successfully.

Table 6.2: Authentication of users

Test Case Number	03

Test Case Name	Edit/view profiles of users (driver And customer)
Description	Admin will view/edit profile of Users (driver And customer)
Testing Technique	Component testing, Black Box Testing
Preconditions	Web portal should be open and admin should already be logged in.
Input Values	Enter name, password and click 'Edit' or 'View' button
Steps	• Open the web portal. • Login as admin. • Click edit or view. • Edit information • Click on the 'Save' button.
Expected output	Edited information should be updated to the database.
Actual output	Database is successfully updated.
Status	Test case passed successfully.

Table 6.3: View profile of users

Test Case Number	04
Test Case Name	Add users (customer and driver)
Description	Admin can add users
Testing Technique	Component testing, Black Box Testing
Preconditions	Web portal should be open and admin should be successfully logged in.
Input Values	• Enter username, password and click 'Log In' • Driver name, address, number and all other required inputs
Steps	• Open the web portal. • Login as admin. • Open the driver Portal/customer Portal. • Add driver/Add Customer. • Update details
Expected output	User should be able to register.
Actual output	User gets successfully Registered.
Status	Test case passed successfully.

Table 6.4: Add users

Test Case Number	05
Test Case Name	Delete driver
Description	Admin can delete driver
Testing Technique	Component testing, Black Box Testing
Preconditions	Web portal should be open and admin should be successfully logged in.
Input Values	• Enter username, password and click 'Log In' • Driver name, address, number and all other required inputs

Steps	• Open the web portal. • Login as admin. • Open the driver Portal. • Delete driver. • Update details
Expected output	Driver should be deleted.
Actual output	Profile gets deleted successfully.
Status	Test case passed successfully.

Table 6.5: Delete Driver

Test Case Number	06
Test Case Name	Delete customer
Description	Admin can delete customer upon complain
Testing Technique	Component testing, Black Box Testing
Preconditions	Web portal should be open and admin should be successfully logged in.
Input Values	• Enter username, password and click 'Log In' • Customer name, address, number and all other required inputs
Steps	• Open the web portal. • Login as admin. • Open the customer Portal. • Delete Customer. • Update details
Expected output	Customer should be deleted.
Actual output	Profile is deleted successfully.
Status	Test case passed successfully.

Table 6.6: Delete Customer

Test Case Number	07
Test Case Name	Signup Android Application
Description	All the users are able to fill a form for signup
Testing Technique	Component testing, Black Box Testing
Preconditions	Web portal should be open
Input Values	Enter name, age, address, contact no and other fields required and click "Signup"
Steps	• Open the Entrega application • Main Screen is open. • Click on Signup. • Enter your details. • Click on "Signup" button.
Expected output	User should be registered as a legitimate user if the required

	fields are entered correctly.
Actual output	User is registered successfully.
Status	Test case passed successfully.

Table 6.7: Signup android application

Test Case Number	08
Test Case Name	Signing in of users android application
Description	All the users will input credentials and sign in on application only one time.
Testing Technique	Component testing, Black Box Testing
Preconditions	Web portal should be open
Input Values	• Enter user name/registered id • Password
Steps	• Open the Entrega application • Main Screen is open. • Enter your details. • Click on "Sign In" button.
Expected output	User should get logged in to his account and it should be one time procedure
Actual output	User is signed In successfully.
Status	Test case passed successfully.

Table 6.8: Sign inandroid application

Test Case Number	09
Test Case Name	Location tracking
Description	Customer Inputs the location for pick up and drop of the delivery item
Testing Technique	Component testing, Black Box Testing
Preconditions	Entrega Application should be open and customer should be Signed In.
Input Values	• Enter Pick up location • Enter drop of location
Steps	• Open the Entrega application • Main Screen is open. • Enter your pick up and drop of location. • Click on Book Ride to continue
Expected output	Pick up and drop of location should be pinned
Actual output	Location pinned successfully.
Status	Test case passed successfully.

Table 6.9: Location Tracking

Test Case Number	10
Test Case Name	Dimension input and vehicle selection
Description	Customer Inputs the dimension of the item or selects the required vehicle
Testing Technique	Component testing, Black Box Testing
Preconditions	Entrega Application should be open and customer should be Signed In.
Input Values	• Length Breath and height In Inches
Steps	• Open the Entrega application • Main Screen is open. • Enter your pick up and drop of location. • Click on book tide to continue • Add dimension or select the vehicle required. • Click to continue
Expected output	Dimension or vehicle is selected successfully
Actual output	Dimension Input Successful.
Status	Test case passed successfully.

Table 6.10: Dimension input and vehicle selection

Test Case Number	11
Test Case Name	Ride booking and driver allocation
Description	Customer books ride and driver is allotted to the customer who is nearest to the customer and has the vehicle required by customer.
Testing Technique	Component testing, Black Box Testing
Preconditions	Entrega application should be open and customer should be signed in. The pickup location drop of location should be inputted and dimension should be given as well.
Input Values	• Enter pick up location • Enter drop of location • Dimension
Steps	• Click on book ride to continue.
Expected output	Ride is booked and driver is allotted according to the requirement of the customer and the driver which is nearest to the customer.
Actual output	Order successfully placed.
Status	Test case passed successfully.

Table 6.11: Ride booking and driver allocation

Test Case Number	12
Test Case Name	Track Ride (Admin)
Description	Admin can track any ride.
Testing Technique	Component testing, Black Box Testing

Preconditions	Web portal should be open and admin should be Signed In.
Input Values	User ID,user name etc
Steps	• Click on user portal(customer /driver portal) • Search user by user ID • Track the ride and activity
Expected output	Admin is able to track activates of users
Actual output	Ride Tracking Successful.
Status	Test case passed successfully.

Table 6.12: Track ride (Admin)

Test Case Number	13
Test Case Name	Assign ride to driver
Description	Admin can assign ride to any driver nearest to the customer
Testing Technique	Component testing, Black Box Testing
Preconditions	Web portal should be open and admin should be Signed In.
Input Values	• Location of driver.
Steps	Click on The driver near the customer. Click on assign ride to the driver to assign the ride
Expected output	Ride is assigned successfully.
Actual output	Ride assigned to the driver.
Status	Test case passed successfully.

Table 6.13: Assign ride to driver

Test Case Number	14
Test Case Name	Admin View drivers progress.
Description	Admin has the record of all the rides by a driver and total fare of all those rides.
Testing Technique	Component testing, Black Box Testing
Preconditions	Web portal should be open and admin should be Signed In.
Input Values	• Driver id. • Driver credentials.
Steps	• Login to Web Portal • Open the Driver Portal. • Input Driver unique id. • Click on Search. • All the information will be displayed.
Expected output	All information and rides of driver are shown successfully.
Actual output	View of progress is successful.,
Status	Test case passed successfully.

Table 6.14: Admin view drivers progress

Test Case Number	15
Test Case Name	Feedback
Description	Customer and driver can both give feedback after every ride.
Testing Technique	Component testing, Black Box Testing
Preconditions	• Android application is open. • Ride is booked and has just ended.
Input Values	Rate from once star to five stars with writing the descriptions.
Steps	• Click on the feedback button. • Allot the deserved stars from 1 to 5. • Enter comments in feedback. • Click 'submit'
Expected output	Feedback submitted.
Actual output	Feedback is submitted successfully.
Status	Test case passed successfully.

Table 6.15: Feedback

Test Case Number	16
Test Case Name	Update users information (driver)
Description	Admin can update the information of the users.
Testing Technique	Component testing, Black Box Testing
Preconditions	• Admin is logged in
Input Values	Enter username, password and click 'log in'
Steps	• Open web portal • Log in as admin • Open the drivers portal. • Click on update information. • Enter the update and click on save.
Expected output	Information will be updated
Actual output	Information is updated
Status	Test case passed successfully.

Table 6.16: Update driver information

Test Case Number	17
Test Case Name	Send notification

Description	Admin can send notification to the entire user.
Testing Technique	Component testing, Black Box Testing
Preconditions	• Web portal is opened. • Admin is logged in
Input Values	Contact of all users.
Steps	• Open web portal • Login as admin. • Click on 'Send notification' • Add the contacts or user ID of the users. • Send Notification
Expected output	Notification will be send
Actual output	Notification are sent successfully
Status	Test case passed successfully.

Table 6.17: Send notification

Test Case Number	18
Test Case Name	Read notification
Description	Users can read notification
Testing Technique	Component testing, Black Box Testing
Preconditions	• Entrega application is opened. • User is logged in
Input Values	Enter contact no, code and click 'Log In'
Steps	• Open android application. • Login as user. • Click on 'notification'
Expected output	Notification will be displayed
Actual output	Notification are displayed successfully
Status	Test case passed successfully.

Table 6.18: Read notification

6.12 INTEGRATION TESTING

Test Case Number	19
Test Case Name	Driver/customer registration
Description	Testing the integration of 'Register' user interface with its Functionality.
Testing Technique	Component testing, White Box Testing
Preconditions	Admin is logged in.

Input Values	Admin enters driver or customer details and clicks on 'Save' button.
Steps	• Admin logs in • Admin clicks “add driver’ or ‘add customer’ • Enters their details • Clicks on save button
Expected output	Customer or driver details should be saved in database
Actual output	Customer or driver details are saved in database
Status	Test case passed successfully.

Table 6.19: User Registration

Chapter 7

7. Future Work

A system of this magnitude always needs continuous work to evolve. There are a lot of possible changes and additions that can be done to the system to improve its performance and functionalities. The system has been made in a modular fashion which enables integrating new features very easy.

7.1 Extended Scope

1. **More Vehicles:** Implement more than the current number of vehicles by studying the depth parameters required for the implementation. This will help in further evaluation of the performance and improvement in functionalities of Entrega.
2. **Extending Vehicle Type:** Currently Entrega is reserved only to three type of vehicles and which are all related to loading category. In future, as the business flourish we can extend it to cab service too. This will improve the performance, enhance the rating of Entrega and help to bring in more revenue. Moreover extension will result into all facilities under one roof.
3. **Data Analytics**: Applying data analytics to finish the need of selection of vehicle type all the time by customer either by selecting vehicle type or putting in some sort of dimension. We can do data analysis to pre-store the dimension of common thing such as refrigerator, washing machine etc. By which it will help us sending the right vehicle according to the need of driver.
4. **Extending the Service**: We can extend the service to other cities as the progress and business flourish and people are aware of the service we provide. In this way we will be able to facilitate more people and more audience in a wide range of map.

Chapter 8

8. Conclusion

8.1 Overview

The purpose of the application is to enable the proper loading vehicle services so that the regular users of the loading vehicle can easily use the service and enhance their performance. One the other hand it helps the drivers of the loading vehicle to get more rides in a systematic manner and earn more. This will help both the user and the driver to save time, budget, other difficulties and improve the journey experiences.The android application will be available to all the citizens. To get registered as driver the driver needs to visit to the office and perform his/her manual registration and submit the required document and credentials. The application also has a web portal which the owner, administrator and employees of Entrega can use to login. All these can view the driver and the customer and register new driver. Administrator will be able to verify the user after checking his data and credentials. Administrator can set drivers goal, rides, view progress of the ride and give remarks. The administrator can also block the users. The customer can view rides and track it as well.

8.2 Objective Achieved

The Project helped to achieve the objectives of learning software development process/cycle, ruby-on-rails frame work, mvc, react and react native, androiddevelopment,APIs, postman protocol, Ubuntu, other similar concepts, Web Development, handling network issues and integration of databases. It also helped us understand what problems are faced when developing a project in the industry.

Chapter 9

9. BIBLIOGRAPHY

[1] https://www.uberfreight.com/

[2] https://www.thehindubusinessline.com/blink/work/truckers-ubercool-idea/article7256041.ece

[3] https://www.uship.com/

[4] https://facebook.github.io/react-native/

[5] https://www.getpostman.com/

[6] https://rubyonrails.org/

[7] http://releases.ubuntu.com/18.04/

[8] https://www.researchgate.net/post/IEEE_templates_for_Thesis_and_Report

[9] https://www.ieee.org/conferences/publishing/templates.html

[10] https://developer.android.com/studio

[11] https://www.androidauthority.com/best-emulators-for-android-315958/

[1] FYP Thesis 'KiBeOCA' 2014-15 Supervised by Dr.SeemabLatif

[2] FYP Thesis 'Virtual Dressing Room' 2015-16 Supervised by Dr.SeemabLatif

[3] FYP Thesis 'KiRA' 2017-18 Supervised by Asst. Prof Bilal Rauf

[4] FYP Thesis 'MOBILE APP FOR METROPOLITAN COMPLAINT MANAGEMENT SYSTEM

’ 2016-17 Supervised by Col(R) Muhammad Saleem