
Software Requirements Specification

for

<NaMeNaYou>

Version 1.0 approved

Prepared by <author>

<organization>

<date created>

1. Introduction.....	1
1.1 Purpose	1
1.2 Document Convention.....	2
1.3 Intended Audience and Reading Suggestions.....	2
1.4 Project Scope	3
1.5 Objective(s)/Aim(s)/Target(s)	3
1.6 Nature of End Product	4
1.7 References	4
2. Overall Description.....	5
2.1 Product Perspective	5
2.2 Product Functions.....	5
2.2.1 Index screen	6
2.2.2 Registration	6
2.2.3 Login	6
2.2.4 Dashboard of customer.....	7
2.2.5 Dashboard of driver.....	7
2.2.6 Dashboard of admin	7
2.3 Product Features	8
2.3.1 Developer Portal.....	8
2.3.2 Application Development	8
2.3.3 Hardware Host Device	8
2.4 Operating Environment	9
2.5 User Classes and Characteristics	9
2.6 Design and Implementation Constraints.....	9
2.7 Privacy.....	10
2.8 Hardware Specifications.....	10
2.9 False Positives	10
2.10 User Documentations and conditions	10
2.10.1 Security	10
2.10.2 Assumptions and Dependencies.....	10
3. External Interface Requirements	11
3.1 User Interface	11
3.2 Hardware Interfaces.....	11
3.3 Software interfaces:	11
3.4 Communication Interface	12
4. Product Features / Functional Requirements	13
4.1 Functionality	13
4.1.1 Interactive and Responsive GUI	13
4.1.2 Design and implementation of customer functionalities.....	13
4.1.3 Design and implementation of driver functionalities	14
4.1.4 Store the data in database	15
4.1.5 Maintain user profile.....	15
4.1.6 Email confirmation.....	16
4.2 Behavioral Requirement:.....	17
4.3 Use Cases Diagrams	18
4.3.1 Use case diagram of Customer.....	18
4.3.2 Use case diagram of driver.....	19
4.3.3 Use case diagram of admin	20
4.3.4 Use Cases Description.....	21
5. Technical Requirements.....	76
5.1 Usability	76
5.1.1 Graphical User Interface	76
5.1.2 Accessibility.....	76
5.2 Reliability and Availability	76
5.2.1 Back-end Internal Computers.....	76

5.2.2	Internet Service Provider.....	76
5.3	Security.....	77
5.3.1	Data Transfer.....	77
5.3.2	Data Storage.....	77
5.4	Supportability	77
5.4.1	Configuration Management Tool.....	77
5.5	Design Constraints.....	78
5.5.1	Standard Development Tools.....	78
5.5.2	Web Based facility	78
5.6	On-line User Documentation and Help System Requirements	78
5.7	Purchased Components.....	78
5.8	Interfaces	78
6.	Other Non-functional Requirements.....	79
6.1	System access	79
6.2	Performance Requirements.....	79
6.3	Security Requirements.....	79
6.4	Hardware/Software.....	79
6.5	Database	80
6.6	User Interface	80
Appendix A: Glossary.....		81

1. Introduction

In this section the overview and scope overview of the SRS will be discuss in detail. The purpose of the document is to provide complete detail of *“NaMeNaYou”*, list of abbreviations, functions and features of the application. The main objective of this SRS documentation is to provide the detail analysis and gather the content in depth of the problem statement defined by the *“NaMeNaYou”*. Moreover this document provides the high level system features to engage the stakeholders and other participants involved in the development of *“NaMeNaYou”* application. This SRS document provides the requirements of *“NaMeNaYou”* in detail to the developers to develop this application.

1.1 Purpose

The main purpose of this SRS document is analyse and collect all the information and requirements which defines the system properly to the admin, database manager and the end users (who use the system). All the ideas to develop the *“NaMeNaYou”* is combined to predict any sort of information which can be used to gain the better understanding of the application and concepts which can be developed later, all the ideas will be consider to develop the application.

The detailed overview of the application, its goals, parameters and objectives are provided in this SRS document. This document provides the application functional, non-functional requirements, hardware and software requirements to target the description of the application. It defines the flow of the application and the interaction of the user with the system in detail. However, it helps the designers and developers to support in the software development life cycle (SDLC) process.

The purpose of this “*NaMeNaYou*” application is to develop a mobile application in which users can order a taxi cab in the West Africa country of Nigeria and take safe journey to their destinations. Admin will control the flow of users and drivers on the application. The application will also provide the training session, a learning school for the users who want to learn how to be safe on the application.

1.2 Document Convention

The document conventions are as follows:

Font style and Size

- o Headings: Times New Roman (18).
- o Sub-headings: Times New Roman (14).
- o Text: Times New Roman (12).
- o Line Spacing 2.0

1.3 Intended Audience and Reading Suggestions

This document is intended for the developers and designers which will develop the “*NaMeNaYou*” as per the requirements specified in the document. This document includes the scope of this system. That why we need this system? What is the purpose is to develop this product? In this document, we mention the features of this product, functional and non-functional requirements and the environment under which we have to work and also mention what we have learned during working on this project, different use cases, interfaces, our objectives which have to be achieved after the completion of this project, challenges we face in order to complete our product. Last but not least our completeness criteria.

Now, how we are able to complete this document developers gather all requirements in from of documentation and then they develop this product. The project manager will control the entire project including his all partners working on this product. Marketing staff advertise this product that how beneficial this product is and about the ease of product.

1.4 Project Scope

The system will be developed by the team of Engineers which have the knowledge of software development, networking, databases and graphic engineering. Each module of the system will be developed individually and merge in the end. The Quality assurance team will determine the functionality and feature of the systems and find out the bugs, errors and all other flaws that can be produced in future use of the system in any case. The system will be online for user which they can use on their personal computers, laptops and mobile phone devices.

1.5 Objective(s)/Aim(s)/Target(s)

Following are the objectives we want to achieve:

- Design a simple and interactive graphical interface of the system
- Design and implementation of all user portals
- Design and development of database system for overall system
- Design and development of driver module
- Design and development of customer module
- Design and implementation of admin module
- Implementation of support center
- Implementation of 24/7 help center

- Implementation of API's
- Performance Testing
- Quality testing

1.6 Nature of End Product

The end product will be mobile based system and it will require operating environment for a client and server GUI. The system will be operated with mobile phones.

1.7 References

- https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc
- <https://www.namenayoutaxi.com>

2. Overall Description

In this portion, system perspective, system features, operating environment, design constraints, assumptions and dependencies will be explained in detail. This document contains the problem statement that the current system is facing which is hampering the growth opportunities of the company. It further contains a list of the stakeholders and users of the proposed solution. It also illustrates the needs and wants of the stakeholders that were identified in the brainstorming exercise as part of the requirements workshop. It further lists and briefly describes the major features and a brief description of each of the proposed system.

The following SRS contains the detail system perspective from different stakeholders and developers. It provides the detail product functions of “*NaMeNaYou*” with user characteristics permitted constraints, assumptions and dependencies and requirements subsets.

2.1 Product Perspective

This system is not dependent on any other outside system.

- High definition graphic display
- System API's

2.2 Product Functions

The “*NaMeNaYou*” platform is a mobile based application in which users can order a taxi cab in the West Africa country of Nigeria and take safe journey to their destinations. Admin will control the flow of users and drivers on the application. The application will also provide the training session, a learning school for the users who want to learn how to be safe on the application.

“*NaMeNaYou*” application contains following features as mentioned:

- Index screen
- Registration
- Login
- Dashboard of customer
- Dashboard of driver

2.2.1 Index screen

- The users will be able to view login button.
- The users will be able to view registration button.
- The users will be able to connect with the application through facebook.

2.2.2 Registration

- The users will be able to get register on the application.
- The users will be able to enter the first name, last name, email, password, confirm password, mobile number and type of user as driver or customer.
- The users will be able to get register on the application though facebook and google.

2.2.3 Login

- The users will be able to get login on the application.
- The users will be able to login through email.
- The users will be able to through mobile.
- The users will be able to login by entering email and password.
- The users will be able to login by entering mobile number.
- An OTP will be send to the user by which he can get login.
- The users will be able to get login though facebook and google.

2.2.4 Dashboard of customer

- The users will be able to book a ride.
- The users will be able to check booking status.
- The users will be able to calculate fair of the ride.
- The users will be able to view driver history.
- The users will be able to change password.
- The users will be able to give feedback to the driver.

2.2.5 Dashboard of driver

- The driver will be able to view all ride requests.
- The driver will be able to set availability.
- The driver will be able to check online booking.
- The driver will be able to view feedbacks.
- The driver will be able to ride history.

2.2.6 Dashboard of admin

- The admin will be able to get login on the application.
- The admin will be able to view all registered drivers.
- The admin will be able to view all registered users.
- The admin will be able to view requests / queries of users and drivers.
- The admin will be able to handle the calls of users and drivers.
- The admin will be able to search the drivers.
- The admin will be able to assign driver to a customer.

2.3 Product Features

2.3.1 Developer Portal

- **On-boarding:** Activities on-boarding new user to the user guide of system usage
- **Instructions Management:** Manages the activities of customers and drivers
- **Device Management:** Communicate and manages the internet
- **Documentation:** Provides users all necessary documentations for on-boarding, tutorials, how-TOS works and all the guidelines
- **Debugging:** Interface to display all logs generated by “*NaMeNaYou*”
- **Analytics:** Interface to display key statistics from system user devices and backend services
- **User’s Details:** Interface to display the user’s details including their activities
- **Verification:** User will have to be verified before using the system
- **User confirmation:** A confirmation mail will be send to the user’s email when he/she will be logged in the system

2.3.2 Application Development

- The “*NaMeNaYou*” has to be developed in Flutter and it will be usable for both android and iOS
 - The “*NaMeNaYou*” has to be developed in Java for android.
 - The “*NaMeNaYou*” has to be developed in Swift for iOS.

2.3.3 Hardware Host Device

Hardware requirements for insurance on internet will be same for both parties which are as follows:

Processor:	Dual Core
RAM:	2 GB
Hard Disk:	320 GB
NIC:	For each party

2.4 Operating Environment

The system is mobile based and hence will require operating environment for a client and server GUI.

2.5 User Classes and Characteristics

The design of the interface must be minimal (number of clicks for completion of task should be minimum) and user-friendly for anyone to understand. The documentations/walk through must be easy enough for any user to understand and follow.

2.6 Design and Implementation Constraints

The “*NaMeNaYou*” is constrained by the interface to the device of the user. Since there are multiple system and multiple devices manufacturers, the interface will most likely not be the same for every one of them and the system design UI/UX should adapt and adjust on all type of devices. Also, there may be a difference between what features they facilitate each of them provide.

The Internet connection is also a constraint for the system. Since the application fetches data from the database over the internet to get the user’s details, it is crucial that there is an internet connection for the application to function.

2.7 Privacy

The privacy of users will not hurt in the system as the database server will only keep the personal details. The system will save record of all the user's details and it will not be accessed to any other third party.

2.8 Hardware Specifications

Host devices running the system on client devices must run the internet connection.

2.9 False Positives

When there is no internet connection is available for the system then the system may not work properly according to the expectation. So, the internet connection must be required for this system to use it properly.

2.10 User Documentations and conditions

2.10.1 Security

User authentication and authorization is required. Users can only change their own schedules. Personal information must not be revealed due to privacy regulations like HIPAA.

2.10.2 Assumptions and Dependencies

The *“NaMeNaYou”* needs the internet connection, audio and video devices. Since the application must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system as for e.g. Modem, WAN-LAN, Ethernet Cross-Cable.

3. External Interface Requirements

3.1 User Interface

The system interface must be user friendly and conform to consistent and standardized colours and fonts. Help links must be provided. Online documentation must be available and accessible on all pages. The system must meet accessibility standards. Several user interfaces must be available for all users.

3.2 Hardware Interfaces

Since the application must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system as for e.g. Modem, WAN-LAN, Ethernet Cross-Cable.

3.3 Software interfaces:

- The system shall be online, all accounts must exist, and users must log in with.
- Authentication and authorization procedures must be used for the users.
- The system shall communicate with the Configuration to identify all the available components to configure the system.
- The system shall communicate with CRM system to provide support.
- The system shall communicate with internal database system for record management.
- The system shall communicate with social networking module.
- The system shall be verifying the authentication which shall allow the users to use the system.

3.4 Communication Interface

The “*NaMeNaYou*” shall use the HTTPS protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite. The system will access files from its own memory. It will also interface with a database or external files from the internet to get the data. The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for all component of the system.

4. Product Features / Functional Requirements

In this section, we will describe the overall functionality, use cases which can occur in our case, and also how user interacts with our software.

4.1 Functionality

This subsection contains the requirements for the “*NaMeNaYou*”. These requirements are organized by the features discussed in the vision document. Features from vision documents are then refined into use case diagrams to best capture the functional requirements of the system. All these functional requirements can be traced using tractability matrix.

4.1.1 Interactive and Responsive GUI

- Simple and interactive graphical user interface.
- Task completion with minimal clicks.
- Creation of icons and widgets which are displayed to the users and organize them inside a screen window.
- Define functions that will process user and application events.
- Association of specific user events with specific functions.
- Start an infinite event-loop that processes user events. When a user event happens, the event-loop calls the function associated with that event.

4.1.2 Design and implementation of customer functionalities

- The users will be able to view login button.
- The users will be able to view registration button.
- The users and drivers will be able to get register though facebook and google.

- The users will be able to connect with the application through facebook.
- The users will be able to get register on the application.
- The users will be able to enter the first name, last name, email, password, confirm password, mobile number and type of user as driver or customer.
- The users will be able to get login on the application.
- The users will be able to login through email.
- The users will be able to through mobile.
- The users will be able to login by entering email and password.
- The users will be able to login by entering mobile number.
- An OTP will be send to the user by which he can get login.
- The users will be able to book a ride.
- The users will be able to check booking status.
- The users will be able to calculate fair of the ride.
- The users will be able to view driver history.
- The users will be able to change password.
- The users will be able to give feedback to the driver.

4.1.3 Design and implementation of driver functionalities

- The users will be able to view login button.
- The users will be able to view registration button.
- The users will be able to connect with the application through facebook.
- The users will be able to get register on the application.

- The users will be able to enter the first name, last name, email, password, confirm password, mobile number and type of user as driver or customer.
- The users will be able to get login on the application.
- The users will be able to login through email.
- The users will be able to through mobile.
- The users will be able to login by entering email and password.
- The users will be able to login by entering mobile number.
- An OTP will be send to the user by which he can get login.
- The driver will be able to view all ride requests.
- The driver will be able to set availability.
- The driver will be able to check online booking.
- The driver will be able to view feedbacks.
- The driver will be able to ride history.

4.1.4 Store the data in database

- The system will have a database in which all the records of users and their activities will be saved automatically with time to time.

4.1.5 Maintain user profile

- The system shall allow user to create profile and set his credential.
- The system shall authenticate user credentials to view the profile.
- The system shall allow user to update the profile information.

4.1.6 Email confirmation

- The system shall maintain user email information as a required part of user profile.
- The system shall send user confirmation to the user through email.

4.2 Behavioral Requirement:

Authentication

Authentication
Hi Travis
Service Code/Device
SN

Admin

Descriptions

Participants

Access

Customer
Driver

Modules

List of Modules

- Index
- Registration
- Login
- Driver module
- Customer module
- Admin module

NaMeNaYou

NaMeNaYou

Database

4.3 Use Cases Diagrams

4.3.1 Use case diagram of Customer

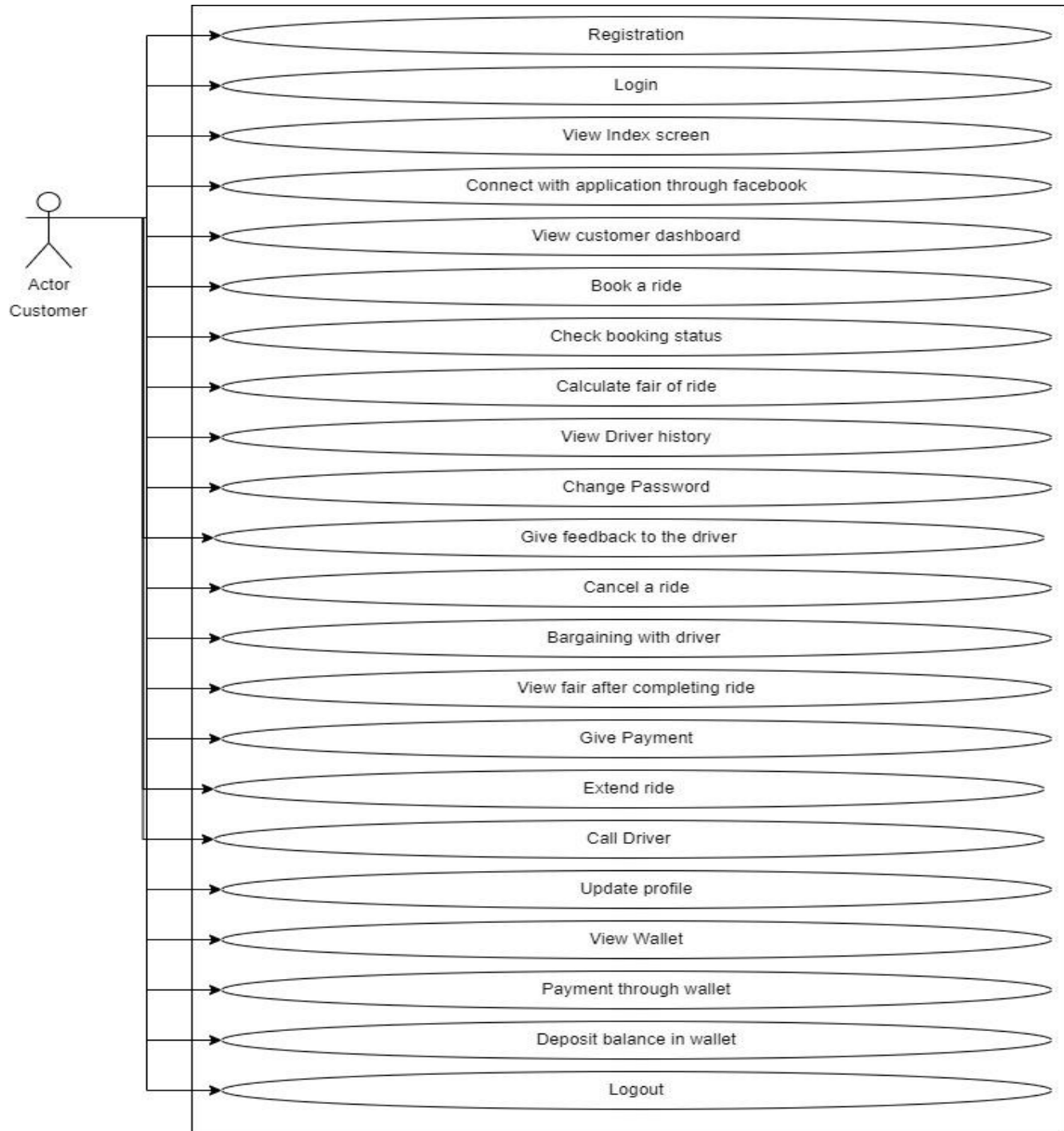


Figure: Use Case diagram of customer

4.3.2 Use case diagram of driver

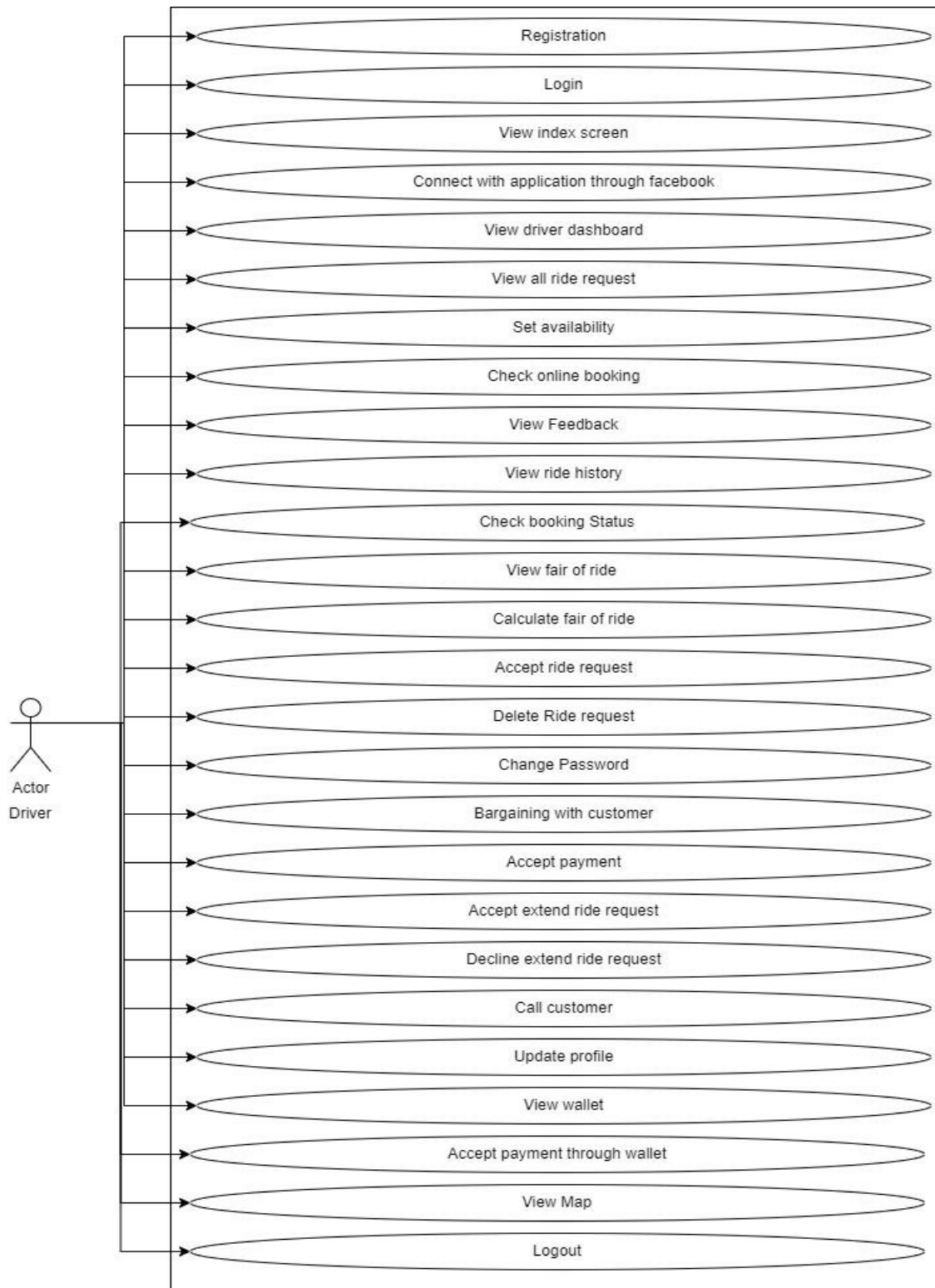


Figure: Use Case diagram of driver

4.3.3 Use case diagram of admin

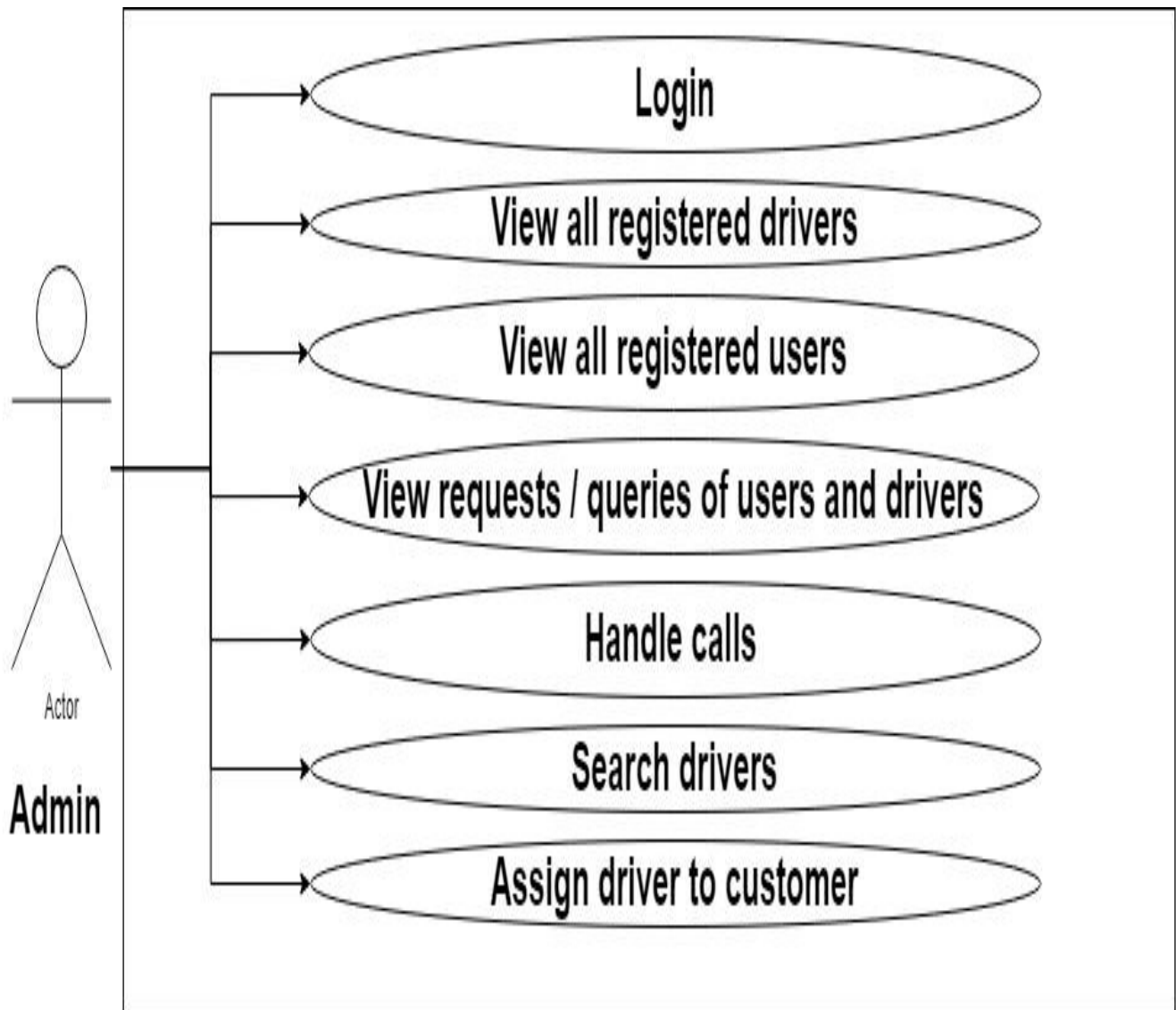


Figure: Use Case diagram of admin

4.3.4 Use Cases Description

Use-Case Name:	Registration	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-001	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the customer so they can get register themselves on the system.	
Precondition:	Customer must have a valid Email ID	
Trigger:	This use case will triggered when customer click on the register button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will type the name, address, home phone number, cell phone, email address.	Step2: System will store the data in database and customer is registered
Alternate Courses:	Alt-Step 3: System is unable to confirm that the username is in the system Alt-Step 4: System is unable to confirm user password Alt-Step 5: System is unable to confirm user email id Alt-Step 6: System sent an error message stating the inaccuracy of user name, email or password	
Conclusion:	Customer will get register in the system.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine valid email address for registration is handled.	

Use-Case Name:	Login	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-002	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can get login themselves on the system.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the login button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will type the username and password.	Step2: System will authenticate the Customer and allow him to login to the application.
Alternate Courses:	Alt-Step 3: System is unable to allow Customer to login.	
Conclusion:	Customer will get login in the system.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View index screen	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-003	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can View index screen on the system.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the View index screen button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the view index screen button.	Step2: System will provide index screen.
Alternate Courses:	Alt-Step 3: System is unable to provide index screen.	
Conclusion:	Customer will View index screen.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Connect with application through facebook	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-004	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can Connect with application through facebook.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the Connect with application through facebook button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the Connect with application through facebook button.	Step2: System will Connect with application through facebook.
Alternate Courses:	Alt-Step 3: System is unable to Connect with application through facebook.	
Conclusion:	Customer will Connect with application through facebook.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View Customer Dashboard	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-005	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can View Customer Dashboard.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on theView Customer Dashboard button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the View Customer Dashboard button.	Step2: System will provide Customer Dashboard.
Alternate Courses:	Alt-Step 3: System is unable to provide Customer Dashboard.	
Conclusion:	Customers will View Customer Dashboard.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Book a ride	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-006	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can Book a ride.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the Book a ride button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the Book a ride button.	Step2: System will allow customer Book a ride.
Alternate Courses:	Alt-Step 3: System is unable to Book a ride.	
Conclusion:	Customer will Book a ride.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Check booking status	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-007	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can Check booking status.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the Check booking status button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the Check booking status button.	Step2: System will allow customer to Check booking status.
Alternate Courses:	Alt-Step 3: System is unable to provide booking status.	
Conclusion:	Customer will Check booking status.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Calculate fair of ride	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-008	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can Calculate fair of ride.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the Calculate fair of ride button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the Calculate fair of ride button.	Step2: System will allow customer to Calculate fair of ride.
Alternate Courses:	Alt-Step 3: System is unable to provide Calculate fair of ride option.	
Conclusion:	Customer will Calculate fair of ride.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View driver history	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-009	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can View driver history.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the View driver history button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the View driver history button.	Step2: System will allow customer to View driver history.
Alternate Courses:	Alt-Step 3: System is unable to provide driver history.	
Conclusion:	Customer will View driver history.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Change Password	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-010	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can Change Password.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the Change Password button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the Change Password button.	Step2: System will allow customer to Change Password.
Alternate Courses:	Alt-Step 3: System is unable to Change Password.	
Conclusion:	Customer will Change Password.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Give feedback to the driver	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-011	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can Give feedback to the driver.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the Give feedback to the driver button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the Give feedback to the driver button.	Step2: System will allow customer to Give feedback to the driver.
Alternate Courses:	Alt-Step 3: System is unable to allow customer Give feedback to the driver.	
Conclusion:	Customer will Give feedback to the driver.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Cancel a ride	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-012	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can Cancel a ride.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the Cancel a ride button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the Cancel a ride button.	Step2: System will allow customer to Cancel a ride.
Alternate Courses:	Alt-Step 3: System is unable to allow customers to Cancel a ride.	
Conclusion:	Customer will Cancel a ride.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Bargaining with driver	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-013	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can Bargaining with driver.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the Bargaining with driver button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the Bargaining with driver button.	Step2: System will allow customer to Bargaining with driver.
Alternate Courses:	Alt-Step 3: System is unable to allow customers to Bargaining with driver.	
Conclusion:	Customer will Bargaining with driver.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View fair after completing ride	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-014	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can View fair after completing ride.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the View fair after completing ride button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the View fair after completing ride button.	Step2: System will allow customer to View fair after completing ride.
Alternate Courses:	Alt-Step 3: System is unable to allow customers to View fair after completing ride.	
Conclusion:	Customer will View fair after completing ride.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Give payment	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-015	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can Give payment.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the Give payment button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the Give payment button.	Step2: System will allow customer to Give payment.
Alternate Courses:	Alt-Step 3: System is unable to allow customers to Give payment.	
Conclusion:	Customer will Give payment.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Extend ride	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-016	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can Extend ride.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the Extend ride button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the Extend ride button.	Step2: System will allow customer to Extend ride.
Alternate Courses:	Alt-Step 3: System is unable to allow customers to Extend ride.	
Conclusion:	Customer will Extend ride.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Call driver	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-017	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can Call driver.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the Call driver button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the Call driver button.	Step2: System will allow customer to Call driver.
Alternate Courses:	Alt-Step 3: System is unable to allow customers to Call driver.	
Conclusion:	Customer will Call driver.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Update profile	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-018	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can Update profile.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the Update profile button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the Update profile button.	Step2: System will allow customer to Update profile.
Alternate Courses:	Alt-Step 3: System is unable to allow customers to Update profile.	
Conclusion:	Customer will Update profile.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View Wallet	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-019	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can View Wallet.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the View Wallet button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the View Wallet button.	Step2: System will allow customer to View Wallet.
Alternate Courses:	Alt-Step 3: System is unable to allow customers to View Wallet.	
Conclusion:	Customer will View Wallet.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Payment through wallet	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-020	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can Payment through wallet.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the Payment through wallet button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the Payment through wallet button.	Step2: System will allow customer to Payment through wallet.
Alternate Courses:	Alt-Step 3: System is unable to allow customers to Payment through wallet.	
Conclusion:	Customer will Payment through wallet.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Deposit balance in wallet	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-021	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Customer	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Customer so they can Deposit balance in wallet.	
Precondition:	Customer must be registered on the application.	
Trigger:	This use case will triggered when Customer click on the Deposit balance in wallet button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Customer will click on the Deposit balance in wallet button.	Step2: System will allow customer to Deposit balance in wallet.
Alternate Courses:	Alt-Step 3: System is unable to allow customers to Deposit balance in wallet.	
Conclusion:	Customer will Deposit balance in wallet.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Logout		Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-022		
Priority:	High		
Source:	Functional Requirement Document		
Primary System Actor:	Customer		
Primary Business Actor:	None		
Other Participating Actors:	Database		
Other Interested Stakeholders:	None		
Description:	This use-case will be used by the Customer so they can Logout.		
Precondition:	Customer must be registered on the application.		
Trigger:	This use case will triggered when Customer click on the Logout button.		
Typical Course Of Events:	Actor Action	System Response	
	Step1: Customer will click on the Logout button.	Step2: System will allow customer to Logout.	
Alternate Courses:	Alt-Step 3: System is unable to allow customers Logout.		
Conclusion:	Customer will Logout.		
Business Rules:	The data will be stored in the database.		
Open Issues:	1. Need to determine how to perform this functionality.		

Use-Case Name:	Registration	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-023	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Driver so they can get register themselves on the system.	
Precondition:	Driver must have a valid Email ID	
Trigger:	This use case will triggered when Driver click on the register button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Driver will type the name, address, home phone number, cell phone, email address.	Step2: System will store the data in database and customer is registered
Alternate Courses:	Alt-Step 3: System is unable to confirm that the username is in the system Alt-Step 4: System is unable to confirm user password Alt-Step 5: System is unable to confirm user email id Alt-Step 6: System sent an error message stating the inaccuracy of user name, email or password	
Conclusion:	Driver will get register in the system.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine valid email address for registration is handled.	

Use-Case Name:	Login	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-024	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Driver so they can get login themselves on the system.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the login button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Driver will type the username and password.	Step2: System will authenticate the Driver and allow him to login to the application.
Alternate Courses:	Alt-Step 3: System is unable to allow Driver to login.	
Conclusion:	Driver will get login in the system.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View index screen	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-025	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Driver so they can View index screen on the system.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the View index screen button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Driver will click on the view index screen button.	Step2: System will provide index screen.
Alternate Courses:	Alt-Step 3: System is unable to provide index screen.	
Conclusion:	Driver will View index screen.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Connect with application through facebook	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-026	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the Driver so they can Connect with application through facebook.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the Connect with application through facebook button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Driver will click on the Connect with application through facebook button.	Step2: System will Connect with application through facebook.
Alternate Courses:	Alt-Step 3: System is unable to Connect with application through facebook.	
Conclusion:	Driver will Connect with application through facebook.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View driver dashboard	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-027	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can View driver dashboard.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the View driver Dashboard button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the View driver Dashboard button.	Step2: System will provide driver Dashboard.
Alternate Courses:	Alt-Step 3: System is unable to provide driver Dashboard.	
Conclusion:	Driver will View driver Dashboard.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View all ride request	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-028	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can View all ride request.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the View all ride request button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the View all ride request button.	Step2: System will allow driver to View all ride request.
Alternate Courses:	Alt-Step 3: System is unable to provide View all ride request.	
Conclusion:	Driver will View all ride request.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Set Availability	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-029	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can Set Availability.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the Set Availability button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the Set Availability button.	Step2: System will allow driver to Set Availability.
Alternate Courses:	Alt-Step 3: System is unable to Set Availability.	
Conclusion:	Driver will Set Availability.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Check online booking	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-030	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can Check online booking.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on theCheck online booking button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the Check online booking button.	Step2: System will allow driver to Check online booking.
Alternate Courses:	Alt-Step 3: System is unable to Check online booking.	
Conclusion:	Driver will Check online booking.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View feedback	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-031	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can View feedback.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the View feedback button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the View feedback button.	Step2: System will allow driver to View feedback.
Alternate Courses:	Alt-Step 3: System is unable to provide feedback.	
Conclusion:	Driver will View feedback	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View ride history	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-032	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can View ride history.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the View ride history button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the View ride history button.	Step2: System will allow driver to View ride history.
Alternate Courses:	Alt-Step 3: System is unable to provide ride history.	
Conclusion:	Driver will View ride history	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Check booking status	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-033	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can Check booking status.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the Check booking status button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the Check booking status button.	Step2: System will allow driver to Check booking status.
Alternate Courses:	Alt-Step 3: System is unable to provide booking status.	
Conclusion:	Driver will Check booking status	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View fair of ride	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-034	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can View fair of ride.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the View fair of ride button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the View fair of ride button.	Step2: System will allow driver to View fair of ride.
Alternate Courses:	Alt-Step 3: System is unable to provide fair of ride.	
Conclusion:	Driver will View fair of ride	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Calculate fair of ride	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-035	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can Calculate fair of ride.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the Calculate fair of ride button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the Calculate fair of ride button.	Step2: System will allow driver to Calculate fair of ride.
Alternate Courses:	Alt-Step 3: System is unable to allow driver to Calculate fair of ride.	
Conclusion:	Driver will Calculate fair of ride	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Accept ride request	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-036	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can Accept ride request.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the Accept ride request button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the Accept ride request button.	Step2: System will allow driver to Accept ride request.
Alternate Courses:	Alt-Step 3: System is unable to Accept ride request.	
Conclusion:	Driver will Accept ride request	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Delete ride request	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-037	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can Delete ride request.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the Delete ride request button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the Delete ride request button.	Step2: System will allow driver to Delete ride request.
Alternate Courses:	Alt-Step 3: System is unable to Delete ride request.	
Conclusion:	Driver will Delete ride request	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Change Password	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-038	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can Change Password.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the Change Password button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the Change Password button.	Step2: System will allow driver to Change Password t.
Alternate Courses:	Alt-Step 3: System is unable to Change Password.	
Conclusion:	Driver will Change Password	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Bargaining with customer	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-039	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can Bargaining with customer.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the Bargaining with customer button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the Bargaining with customer button.	Step2: System will allow driver to Bargaining with customer.
Alternate Courses:	Alt-Step 3: System is unable to Bargaining with customer.	
Conclusion:	Driver will Bargaining with customer	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Accept Payment	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-040	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can Accept Payment.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the Accept Payment button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the Accept Payment button.	Step2: System will allow driver to Accept Payment.
Alternate Courses:	Alt-Step 3: System is unable to Accept Payment.	
Conclusion:	Driver will Accept Payment	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Accept extend ride request	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-041	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can Accept extend ride request.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the Accept extend ride request button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the Accept extend ride request button.	Step2: System will allow driver to Accept extend ride request.
Alternate Courses:	Alt-Step 3: System is unable to Accept extend ride request.	
Conclusion:	Driver will Accept extend ride request	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Decline extend ride request	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-042	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can Decline extend ride request.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the Decline extend ride request button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the Decline extend ride request button.	Step2: System will allow driver to Decline extend ride request.
Alternate Courses:	Alt-Step 3: System is unable to Decline extend ride request.	
Conclusion:	Driver will Decline extend ride request	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Call customer	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-043	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can Call customer.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the Call customer button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the Call customer button.	Step2: System will allow driver to Call customer.
Alternate Courses:	Alt-Step 3: System is unable to Call customer.	
Conclusion:	Driver will Call customer	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Update profile	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-044	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can Update profile.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the Update profile button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the Update profile button.	Step2: System will allow driver to Update profile.
Alternate Courses:	Alt-Step 3: System is unable to Update profile.	
Conclusion:	Driver will Update profile	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View wallet	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-045	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can View wallet.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the View wallet button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the View wallet button.	Step2: System will allow driver to View wallet.
Alternate Courses:	Alt-Step 3: System is unable to provide wallet.	
Conclusion:	Driver will View wallet	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Accept payment through wallet	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-046	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can Accept payment through wallet.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the Accept payment through wallet button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the Accept payment through wallet button.	Step2: System will allow driver to Accept payment through wallet.
Alternate Courses:	Alt-Step 3: System is unable to Accept payment through wallet.	
Conclusion:	Driver will Accept payment through wallet	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View map	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-047	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Driver	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the driver so they can View map.	
Precondition:	Driver must be registered on the application.	
Trigger:	This use case will triggered when Driver click on the View map button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: driver will click on the View map button.	Step2: System will allow driver to View map.
Alternate Courses:	Alt-Step 3: System is unable to View map.	
Conclusion:	Driver will View map	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Logout		Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-048		
Priority:	High		
Source:	Functional Requirement Document		
Primary System Actor:	Driver		
Primary Business Actor:	None		
Other Participating Actors:	Database		
Other Interested Stakeholders:	None		
Description:	This use-case will be used by the driver so they can Logout.		
Precondition:	Driver must be registered on the application.		
Trigger:	This use case will triggered when Driver click on the Logout button.		
Typical Course Of Events:	Actor Action	System Response	
	Step1: driver will click on the Logout button.	Step2: System will allow driver to Logout.	
Alternate Courses:	Alt-Step 3: System is unable to Logout.		
Conclusion:	Driver will Logout		
Business Rules:	The data will be stored in the database.		
Open Issues:	1. Need to determine how to perform this functionality.		

Use-Case Name:	Login	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-049	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Admin	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the admin so they can login.	
Precondition:	Admin must be registered on the application.	
Trigger:	This use case will triggered when admin click on the login button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Admin will click on the login button.	Step2: System will allow admin to login.
Alternate Courses:	Alt-Step 3: System is unable to allow admin to login on application.	
Conclusion:	Admin will login	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View all registered drivers		Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-050		
Priority:	High		
Source:	Functional Requirement Document		
Primary System Actor:	Admin		
Primary Business Actor:	None		
Other Participating Actors:	Database		
Other Interested Stakeholders:	None		
Description:	This use-case will be used by the admin so they can view all registered drivers.		
Precondition:	Admin must be registered on the application.		
Trigger:	This use case will triggered when admin click on the view all registered drivers button.		
Typical Course Of Events:	Actor Action	System Response	
	Step1: Admin will click on the view all registered drivers button.	Step2: System will allow admin to view all registered drivers.	
Alternate Courses:	Alt-Step 3: System is unable to allow admin to view all registered drivers on application.		
Conclusion:	Admin will view all registered drivers		
Business Rules:	The data will be stored in the database.		
Open Issues:	1. Need to determine how to perform this functionality.		

Use-Case Name:	View all registered users	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-051	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Admin	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the admin so they can view all registered users.	
Precondition:	Admin must be registered on the application.	
Trigger:	This use case will triggered when admin click on the view all registered users button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Admin will click on the view all registered users button.	Step2: System will allow admin to view all registered users.
Alternate Courses:	Alt-Step 3: System is unable to allow admin to view all registered users on application.	
Conclusion:	Admin will view all registered users.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	View requests / queries of driver and customer	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-052	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Admin	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the admin so they can view requests / queries of driver and customer.	
Precondition:	Admin must be registered on the application.	
Trigger:	This use case will triggered when admin click on the view requests / queries of driver and customer button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Admin will click on the view requests / queries of driver and customer button.	Step2: System will allow admin to view requests / queries of driver and customer.
Alternate Courses:	Alt-Step 3: System is unable to allow admin to view requests / queries of driver and customer on application.	
Conclusion:	Admin will view requests / queries of driver and customer.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Handle calls	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-053	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Admin	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the admin so they can handle calls.	
Precondition:	Admin must be registered on the application.	
Trigger:	This use case will triggered when admin click on the handle calls button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Admin will click on handle calls button.	Step2: System will allow admin to handle calls.
Alternate Courses:	Alt-Step 3: System is unable to allow admin to handle calls on application.	
Conclusion:	Admin will handle calls.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Search drivers	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-054	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Admin	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the admin so they can search drivers.	
Precondition:	Admin must be registered on the application.	
Trigger:	This use case will triggered when admin click on search drivers button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Admin will click on search drivers button.	Step2: System will allow admin to search drivers.
Alternate Courses:	Alt-Step 3: System is unable to allow admin to search drivers on application.	
Conclusion:	Admin will search drivers.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

Use-Case Name:	Assign driver to customer	Use Case Type Business Requirements <input type="checkbox"/>
Use-Case ID:	Usecase-055	
Priority:	High	
Source:	Functional Requirement Document	
Primary System Actor:	Admin	
Primary Business Actor:	None	
Other Participating Actors:	Database	
Other Interested Stakeholders:	None	
Description:	This use-case will be used by the admin so they can assign driver to customer.	
Precondition:	Admin must be registered on the application.	
Trigger:	This use case will triggered when admin click on assign driver to customer button.	
Typical Course Of Events:	Actor Action	System Response
	Step1: Admin will click on assign driver to customer button.	Step2: System will allow admin to assign driver to customer.
Alternate Courses:	Alt-Step 3: System is unable to allow admin to assign driver to customer on application.	
Conclusion:	Admin will assign driver to customer.	
Business Rules:	The data will be stored in the database.	
Open Issues:	1. Need to determine how to perform this functionality.	

5. Technical Requirements

5.1 Usability

5.1.1 Graphical User Interface

- The system shall provide a uniform look and feel between all the web pages.
- The system shall provide digital details for each profile.
- The system shall provide use of icons and toolbars.

5.1.2 Accessibility

- The system shall provide multi language support.

5.2 Reliability and Availability

5.2.1 Back-end Internal Computers

- The system shall provide storage of all databases on redundant computers with automatic switchover.
- The system shall provide for replication of databases to off-site storage locations.
- The system shall provide RAID V Disk Stripping on all database storage disks.

5.2.2 Internet Service Provider

- The system shall provide a contractual agreement with an internet service provider for T3 access with 99.9999% availability.
- The system shall provide a contractual agreement with an internet service provider who can provide 99.999% availability through their network facilities onto the internet.

5.3 Security

5.3.1 Data Transfer

- The system shall use secure sockets in all transactions that include any confidential user's information.
- The system shall automatically log out all users after a period of inactivity.
- The system shall not leave any cookies on the user's computer containing the user's password.
- The system shall not leave any cookies on the user's computer containing any of the user's confidential information.

5.3.2 Data Storage

- The user's web browser shall never display a user's password. It shall always be echoed with special characters representing typed characters.
- The system's back-end servers shall never display a user's password. The user's password may be reset but never shown.
- The system's back-end servers shall only be accessible to authenticated administrators.
- The system's back-end database shall be encrypted.

5.4 Supportability

5.4.1 Configuration Management Tool

- The source code developed for this system shall be maintained in configuration management tool.

5.5 Design Constraints

5.5.1 Standard Development Tools

- The system shall be built using a standard web page development tool that conforms to either IBM's CUA standards or Microsoft's GUI standards.

5.5.2 Web Based facility

- There are no memory requirements
- The computers must be equipped with web browsers such as internet explorer.
- The data must be stored in such a way that allows the client easy access to it.
- Response time of the system should take no longer than one minute.
- A general knowledge of basic computer skills is required to use the system.

5.6 On-line User Documentation and Help System Requirements

- As the product is online so on-line help system becomes a critical component of the system which shall provide.
- It shall provide specific guidelines to a user for using the system.
- To implement online user help, link and search fields shall be provided.

5.7 Purchased Components

- Not Applicable

5.8 Interfaces

- The protocol used shall be HTTP.
- The Port number used will be 80.

6. Other Non-functional Requirements

6.1 System access

Preconditions include that the system must be online, all accounts must exist, and users must log in with. Authentication and authorization procedures must be used.

6.2 Performance Requirements

The system must be able to service multiple terminal connections simultaneously with at least 75 internet users connected without delay.

The system must be able to process 90% of queries within a fraction of a second.

6.3 Security Requirements

User authentication and authorization is required. Users can only change their own schedules. Personal information must not be revealed due to privacy regulations like HIPAA.

6.4 Hardware/Software

It is anticipated that no special hardware would be required; however, a database management system will be required.

it is anticipated that any web browser could be used and any computer with Internet access could be used by all types of people who will access the system. Servers will be required to store the scheduling data and backups. It seems that no major site adaptations are required.

The system must be online and accessible for Windows, Linux/Unit, or Macintosh systems with a GUI interface through browsers. It must be interoperable with other systems that users are like to need like Adobe Acrobat.

6.5 Database

Database requirements include information collected for customers and drivers. One day for example might consist of many different time slots perhaps spaced 15 minutes apart.

The database should be normalized for space usage efficiency and response time of less than three seconds.

The database must handle a maximum of 2500 users. Any invalid data must be recovered with messages back to the user. 7 days a week for 24 hours each day. A backup plan must be implemented with disaster recovery included.

6.6 User Interface

The system interface must be user friendly and conform to consistent and standardized colours and fonts. Help links must be provided. Online documentation must be available and accessible on all pages. The system must meet accessibility standards. Several user interfaces must be available for all users.

Appendix A: Glossary

Term	Definition
SaaS	Software as a Service
User	who uses the system
24/7	Twenty-four hours a day, seven days a week, all the time.
CRM	User Relationship Management
SDLC	Software Development Lifecycle
GUI	Graphical User Interface