|  |
| --- |
| **Requirements and Design Specification Document**  **<Share Ride>**  Advisor:  Aneeka Shayan  Project Team:    Jahanzaib 212370009  Waqar Ahmad 212370019  Submission Date:  **Sept 20, 2023** |

**Document Information**

| **Category** | **Information** |
| --- | --- |
| Customer | GIFT University Gujranwala |
| Project | <Share Ride> |
| Document | Requirement Specifications |
| Document Version | 1.0 |
| Identifier |  |
| Status | Draft |
| Author(s) | <Abu Bakar Siddique,Rizwan Ashraf , M Ansar,M Umer> |
| Approver(s) |  |
| Issue Date |  |
| Document Location |  |
| Distribution |  |

**Definition of Terms, Acronyms and Abbreviations**

*This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.*

| **Term** | **Description** |
| --- | --- |
| CNIC | Computerized National Identity Card |
|  |  |

**Table of Contents**

[**Introduction**](#_1fob9te) **5**

[Purpose](#_3znysh7) 5

[Scope](#_2et92p0) 5

[Definitions, Acronyms, and Abbreviations](#_3dy6vkm) 5

[References](#_4d34og8) 6

[**Specific Requirements**](#_2s8eyo1) **6**

[2.1.1 Business Requirements:](#_17dp8vu) 6

[2.1.2 User Requirements:](#_3rdcrjn) 7

[2.1.3 Functional Requirements:](#_26in1rg) 9

[2.2 Non-Functional Requirements:](#_lnxbz9) 14

[2.2.1 Performance:](#_35nkun2) 14

[2.2.2 Design Constraints](#_1ksv4uv) 15

[2.2.3 Supportability](#_44sinio) 15

[2.2.4 Usability](#_2jxsxqh) 15

[2.2.5 Reliability](#_z337ya) 16

[**Use Cases**](#_1y810tw) **17**

[Use Case Diagram](#_4i7ojhp) 17

[**Detailed System Design**](#_1ci93xb) **20**

[4.1 Work Breakdown Structure](#_3whwml4) 21

# **Introduction**

“Share Ride” is a transport idea based on a mobile application that lets you comfortably travel from one city to another city as well as within the city. A person who has to travel from one city to another has to pay the amount of the complete ride. Existing applications in the market allow the passenger to book the complete ride (Vehicle). In this application we are giving a solution where the passenger will book the seats that he needs and pay only for these seats.

## **Purpose**

The purpose of the Software Requirement Specification (SRS) document is to give a complete overview of our project of “ShareRide” transportation mobile Application which includes its storyline, user interface and tools used for development and implementation.

## **Scope**

This application provides the facilities to driver and owner of vehicle as well as passengers. At the driver’s end the driver will have to add his/her profile with complete detail also he/she will have to attach his/her CNIC picture, driving license and also a passport size picture. Admin will review the profile of the driver and after verification he will approve the profile of the driver within 24 hours. When the profile of the driver will be approved by admin he will be able to post the ride, update seat availability, add departure and destination location and also will be able to see the passenger’s profile and chat with him/her who booked seats for his/her post. The notification of the driver's post will be sent to all the registered passengers. Driver can also remove his/her post of ride and also be able to set his/her profile availability (ON/OFF).Application will also provide the facility to the driver to check his location even if he/she has no internet connection and driver can also view the location of the passenger. Driver will receive her/his payment by hand as well as online (jazz cash). At the passenger’s end the passenger will have to register first by entering his/her complete profile and also has to attach a CNIC picture for security reasons. After registering his/her self he/she will be able to check the posts of drivers and can book available seats and also be able to cancel the ride. Passengers can see the real time location of the driver and also chat with the driver. He/she will be able to check the profile of the driver and also will be able to see top rated drivers. He/she can make payment by hand or online (jazz cash).He/she will also be able to give rating to driver as well as application. He/she can book a vehicle for self-drive but in this case passengers will have to provide some extra detail for security reasons .Admin will have authority to approve or reject the profile of driver, block driver, track driver and can view the profile of driver and passengers.

## **Definitions, Acronyms, and Abbreviations**

Null

## **References**

[**https://indriver.com/en/city/**](https://indriver.com/en/city/)[**https://play.google.com/store/apps/details?id=com.careem.acma&hl=en&gl=US**](https://play.google.com/store/apps/details?id=com.careem.acma&hl=en&gl=US)

# **Specific Requirements**

## 2.1.1 Business Requirements:

|  |  |
| --- | --- |
| **Sr. No.** | **Description** |
| **BR-01** | To ease the process of traveling. |
| **BR-02** | A mobile based application to use the marketplace. |
| **BR-03** | A simple UI based application for users. |
| **BR-04** | A system that will help the travelers to divide the cost of traveling among them. |
| **BR-05** | A system that will help the drivers to generate income. |
| **BR-06** | A system that will help the passengers to travel with a live location. |
| **BR-07** | A system that will help the passengers to see the complete details of the driver. |

## 2.1.2 **User Requirements:**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Stakeholders** | **User Requirements** |
| **1** | **Driver** | **UR-01:** User will be able to register/Login.  **UR-02:** Users can Set his/her profile picture/avatar.  **UR-03:** User will be able to register his/her vehicle.  **UR-04**: User will have to upload his/her CNIC picture.  **UR-05:** User will have to upload his/her driving license.  **UR-06:** Users will have to add price of ride based on per seat.  **UR-07:** User can post for Ride.  **UR-08**: User can log out any time.  **UR-09:** user can set his availability to on or off.  **UR-10:** user can add his/her vehicle for rent for self-drive or with a driver.  **UR-11:** User can view his/her profile.  **UR-12:** User can view the profile of the passenger.  **UR-13:** Users are able to chat with passengers.  **UR-14:** User will have to add destination location while posting ride.  **UR-15:** User will have to add departure location while posting ride.  **UR-16:** User can cancel the ride which he/she posted.  **UR-17:** user can book a seat for any passenger.  **UR-18:** User can add payment method |
| **2** | **Passenger** | **UR-01:** User shall be able to register/Login.  **UR-02:** Users can set their profile picture/avatar.  **UR-03:** user can view his/her profile.  **UR-04:** User will have to upload his/her CNIC picture.  **UR-05:** User can view the profile of the driver.  **UR-06:** Users can book seats.  **UR-07:** Users can book a vehicle for self-driving.  **UR-08:** User can log out any time.  **UR-09:** User will have to set his/her complete profile.  **UR-10:** User can chat with the driver. |
| **3** | **Admin** | **UR-01:** User shall be able to login/logout.  **UR-02:** User shall be able to review the driver’s profile  **UR-03:** User has authority to approve/reject driver’s profile.  **UR-04:** User can track the driver.  **UR-05:** User has authority to block the driver. |

## 2.1.3 **Functional Requirements:**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Stakeholders** | **Functional Requirements** |
| **1** | **Driver** | **FR-01:** System will provide the facility to driver register his/her self.  **FR-02:** System will provide the facility to driver login.  **FR-03:** System will provide the facility to driver logout.  **FR-04:** System will provide the facility to the driver to check the profile of Passengers.  **FR-05:** System will provide the facility to the driver to add payment methods online (jazz cash) or by hand.  **FR-06:** System will provide the facility to the driver to edit his/her profile.  **FR-07:** System will provide the facility to the driver to post a ride.  **FR-08:** System will provide the facility to the driver to edit his/her post.  **FR-09:** System will provide the facility to the driver to show his/her history.  **FR-10:** System will provide the facility to the driver to cancel the ride.  **FR-11:** System will provide the facility to the driver to update the availability of seats.  **FR-12:** System will provide the facility to the driver to add his/her vehicle for rent for self-drive or with driver.  **FR-13:** System will provide the facility to the driver to view his/her profile.  **FR-14:** System will provide the facility to drivers to view the attached documents of customers/passengers who want to book a vehicle for self-driving.  **FR-15:** System will provide the facility to drivers to chat with passengers.  **FR-16:** System will provide the facility to the driver to add destination location.  **FR-17:** System will provide the facility to the driver to add departure location.  **FR-18:** System will provide the facility to the driver to view booked seats. |
| **2** | **Passenger** | **FR-01:** System will provide the facility to the passenger to register his/her self.  **FR-02:** System will provide the facility to the passenger to login.  **FR-03:** System will provide the facility to the passenger to logout.  **FR-04:** System will provide the facility to the passenger to check the profile of the driver.  **FR-05:** System will provide the facility to the passenger to check the real time location of the driver.  **FR-06:** System will provide the facility to the passenger to edit his/her profile.  **FR-07:** System will provide the facility to the passenger to book a vehicle for self-driving.  **FR-08:** System will provide the facility to the passenger to chat with the driver.  **FR-09:** System will provide the facility to the passenger to view his/her travel history.  **FR-10:** System will provide the facility to the passenger to book the available seats.  **FR-11:** System will provide the facility to the passenger to view his/her profile.  **FR-12:** System will provide the facility to the passenger to attach complete documents and detail in case of booking the vehicle for self-drive.  **FR-13:** System will provide the facility to the passenger to make payment online (jazz cash). |
| **3** | **Admin** | **FR-01:** System will provide the facility to the admin to review the newly driver’s profile.  **FR-02:** System will provide the facility to the admin to approve/reject newly driver’s profile  **FR-03:** System will provide the facility to the admin to track the driver.  **FR-04:** System will provide the facility to the admin to block the driver. |
| **4** | **System** | **FR-01:** System will be able to auto remove his/her post after 30 minutes of departure time. |

## 2.2 **Non-Functional Requirements:**

## 2.2.1 **Performance:**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Title** | **Description** |
| **PR-01** | **Loading time** | Application should be run in 2 seconds depending on internet speed. |
| **PR-01** | **Crud operations** | System will perform 99% correct CRUD operations. |

## 2.2.2 **Design Constraints**

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Title** | **Description** |
| **DC-01** | **Responsiveness** | System shall be android, iOS based and responsive for users. |

## 2.2.3 **Supportability**

We are using Dart language using Flutter extension so the application will support both android as well as iOS.

• System shall be able to support android devices with version 8.0 and onward.

• System shall be able to support iOS devices with version 10.0 and onward.

## 2.2.4 **Usability**

• The program shall be able to open in 30 to 50 seconds.

• The program shall be able to generate notification to all register users.

• The program shall have a good refresh rate.

## 2.2.5 **Reliability**

• System shall be available 24/7.

• System shall be protected from unauthorized access.

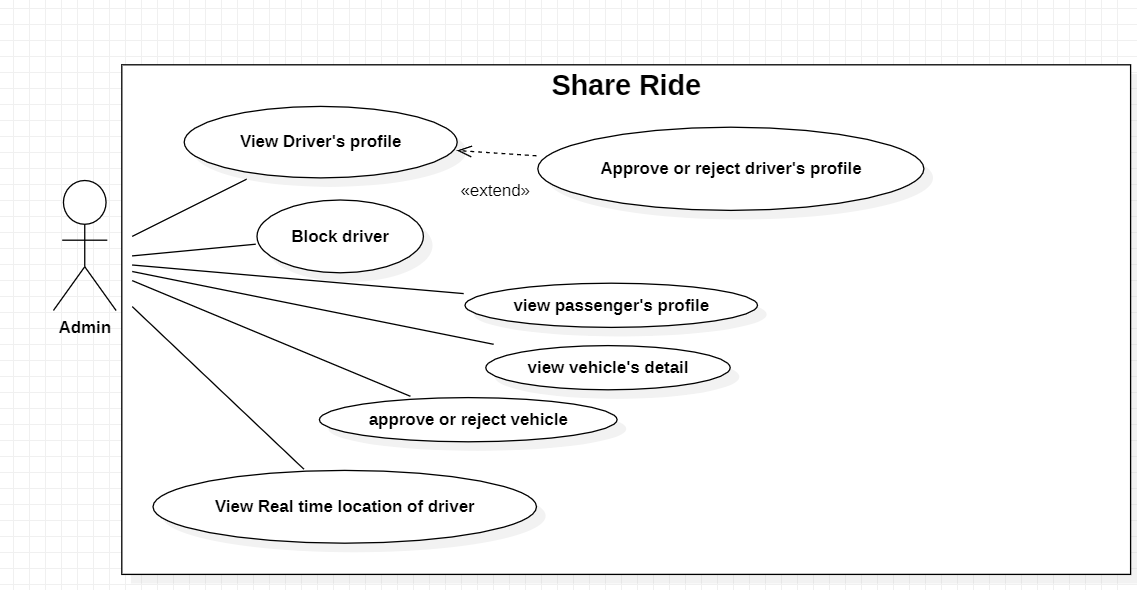
• Systems shall have a less failure rate.

• System shall be able to recover itself within 5 to 7 seconds.

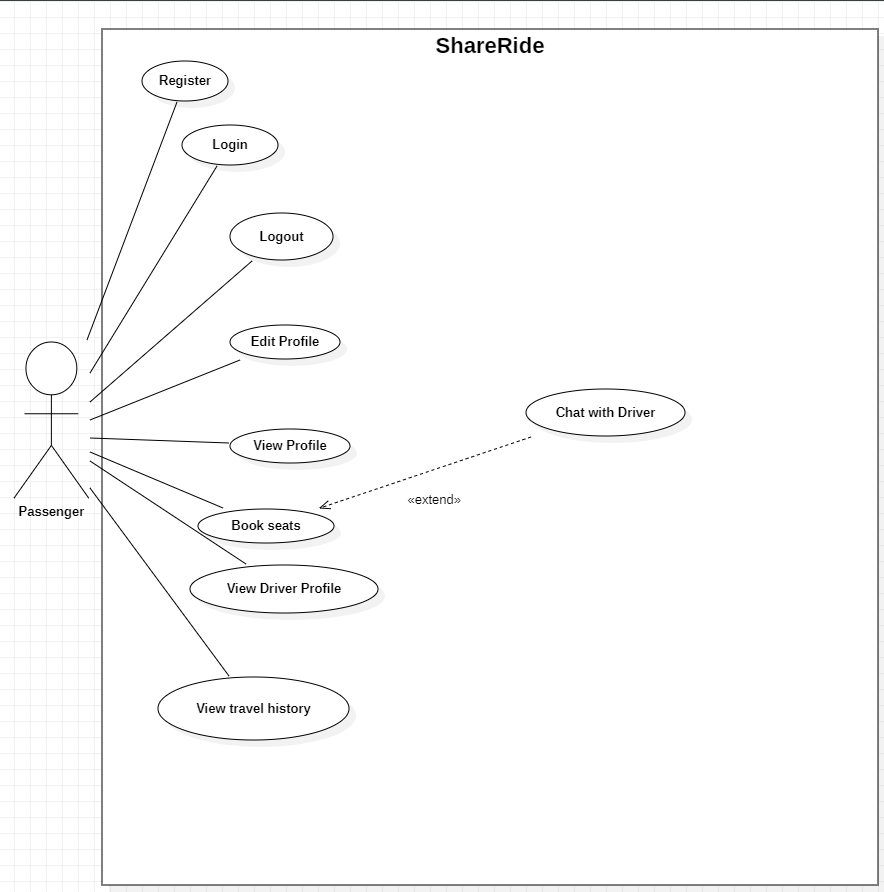
# **Use Cases**

## **Use Case Diagram**

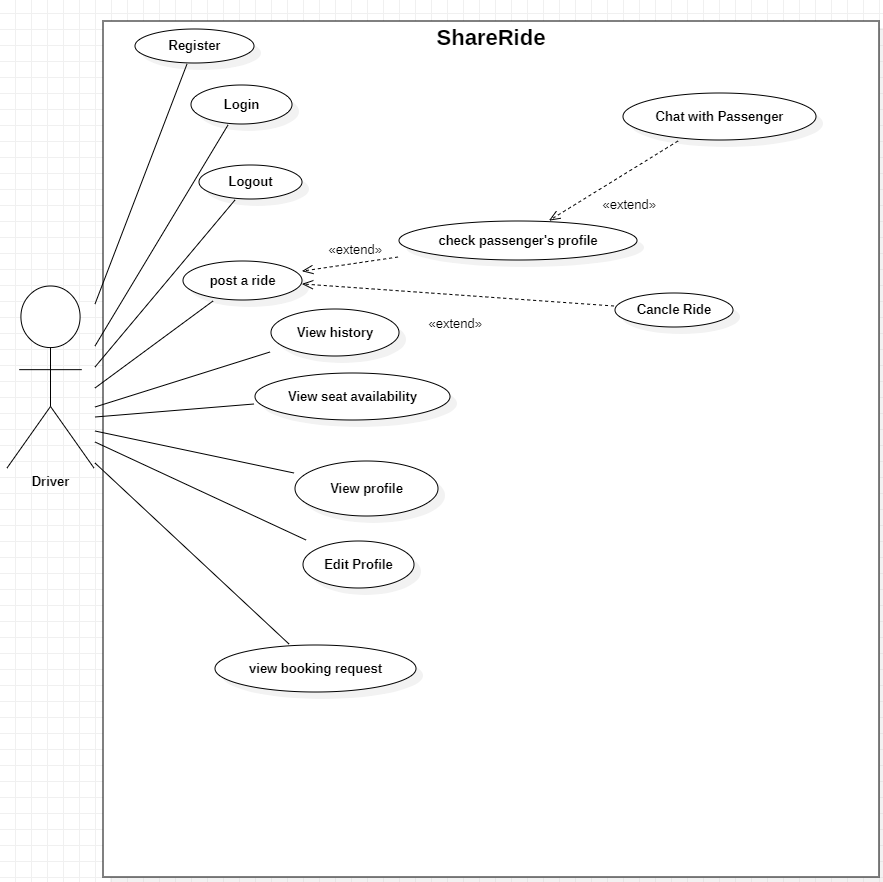
**Admin**

****

**Passenger**

****

**Drtiver**

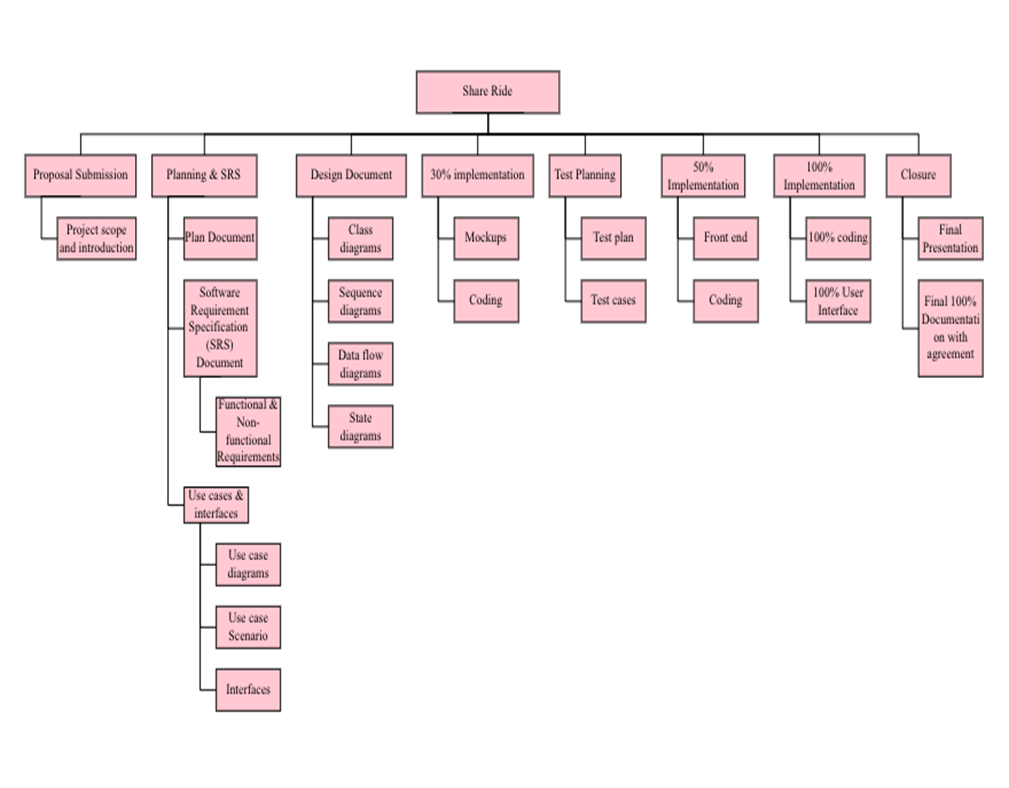
****

## 

# Detailed System Design

[**https://docs.google.com/spreadsheets/d/1ryy2wINnu4kvu-EtLCyWkCru8lP9gsWFPsWyqujKdmc/edit?usp=sharing**](https://docs.google.com/spreadsheets/d/1ryy2wINnu4kvu-EtLCyWkCru8lP9gsWFPsWyqujKdmc/edit?usp=sharing)

## 4.1 Work Breakdown Structure



**5.Learning Outcomes :**

Completing this project, we will be able to understand:

* The complete function of Firebase.
* A new framework for developing applications (Flutter).
* Installation of android studio,vs code,intellij etc.
* Working with GitHub repositories.
* A new programming language Dart.

**6.Practical Applications :**

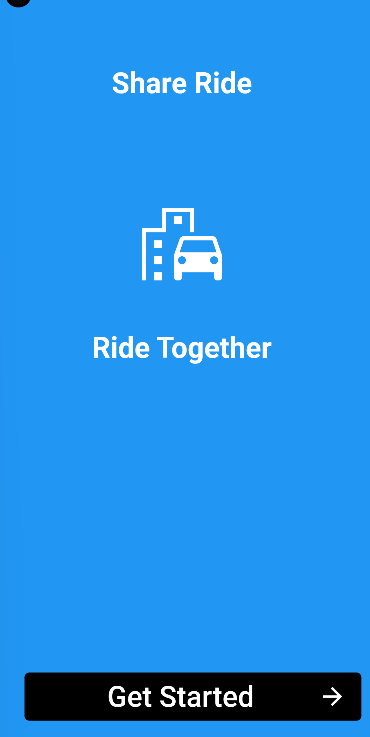
* Uber
* Cream
* In driver
* Carpoolyn

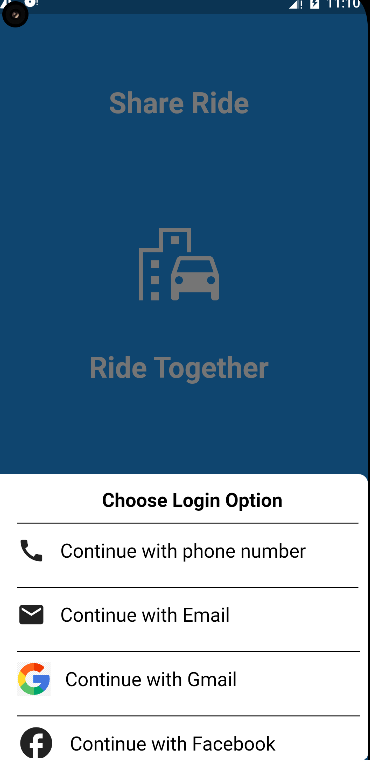
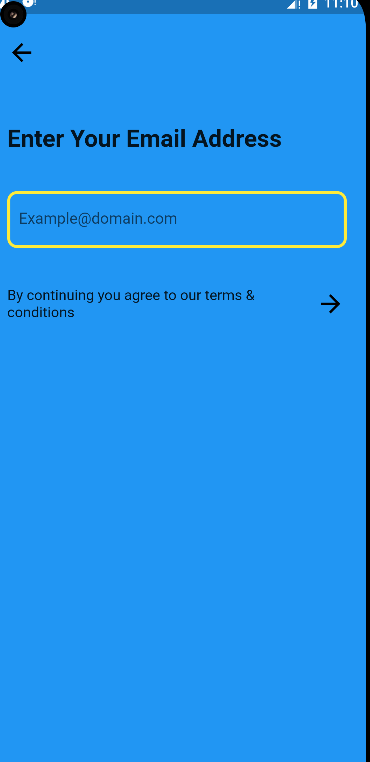
7.**Supporting Information**

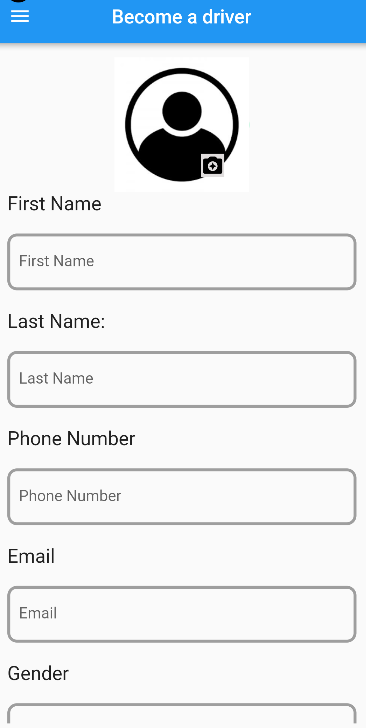
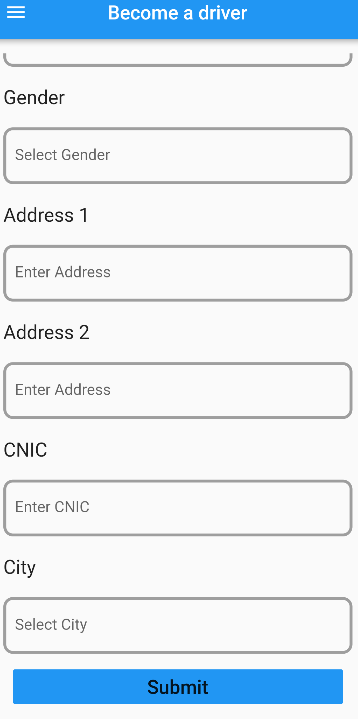
**Prototypes:**

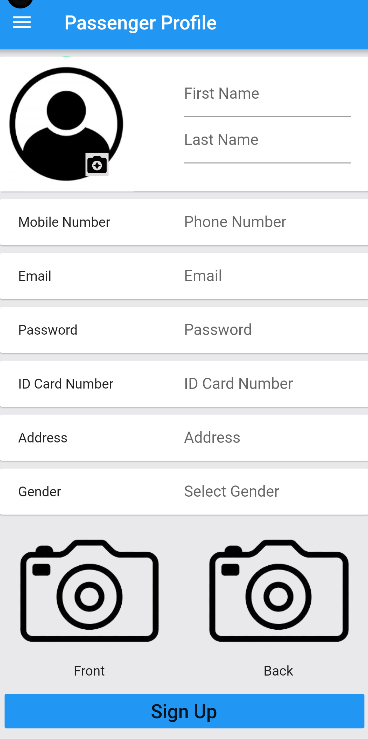
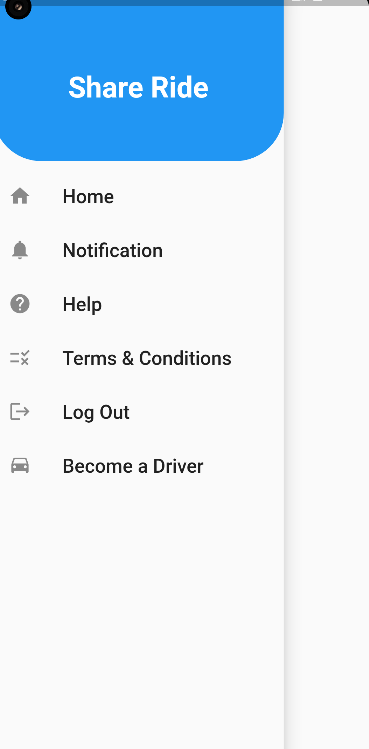
** **

** **

** **

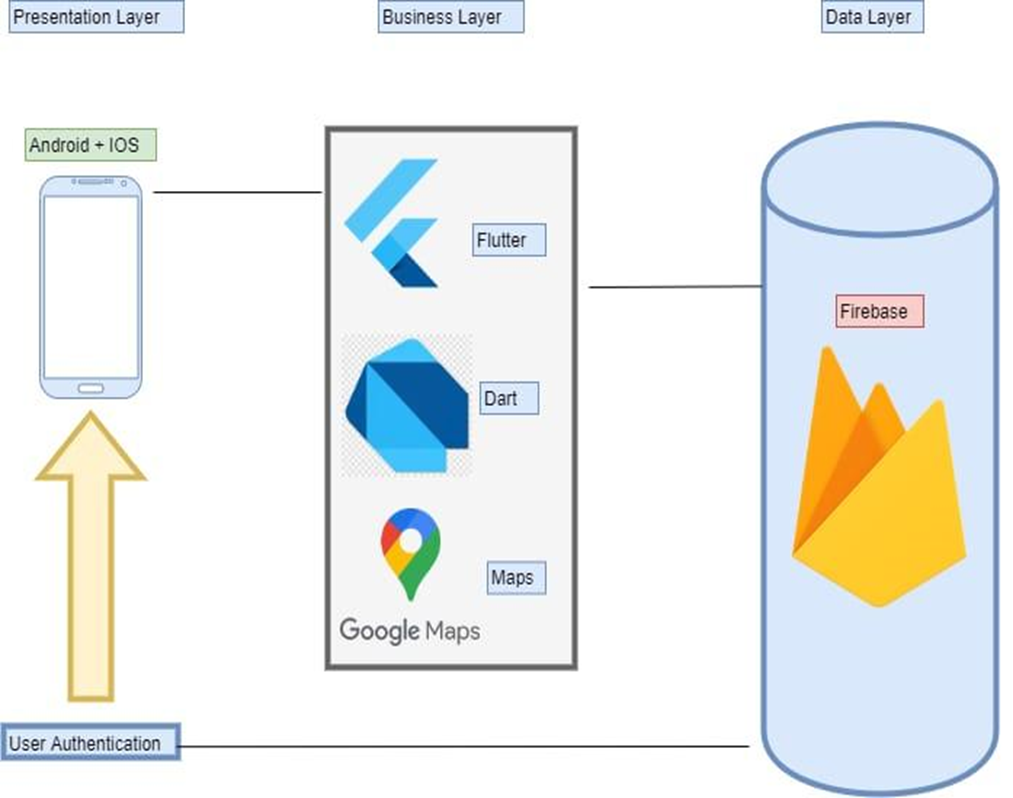
** **

** **

** **

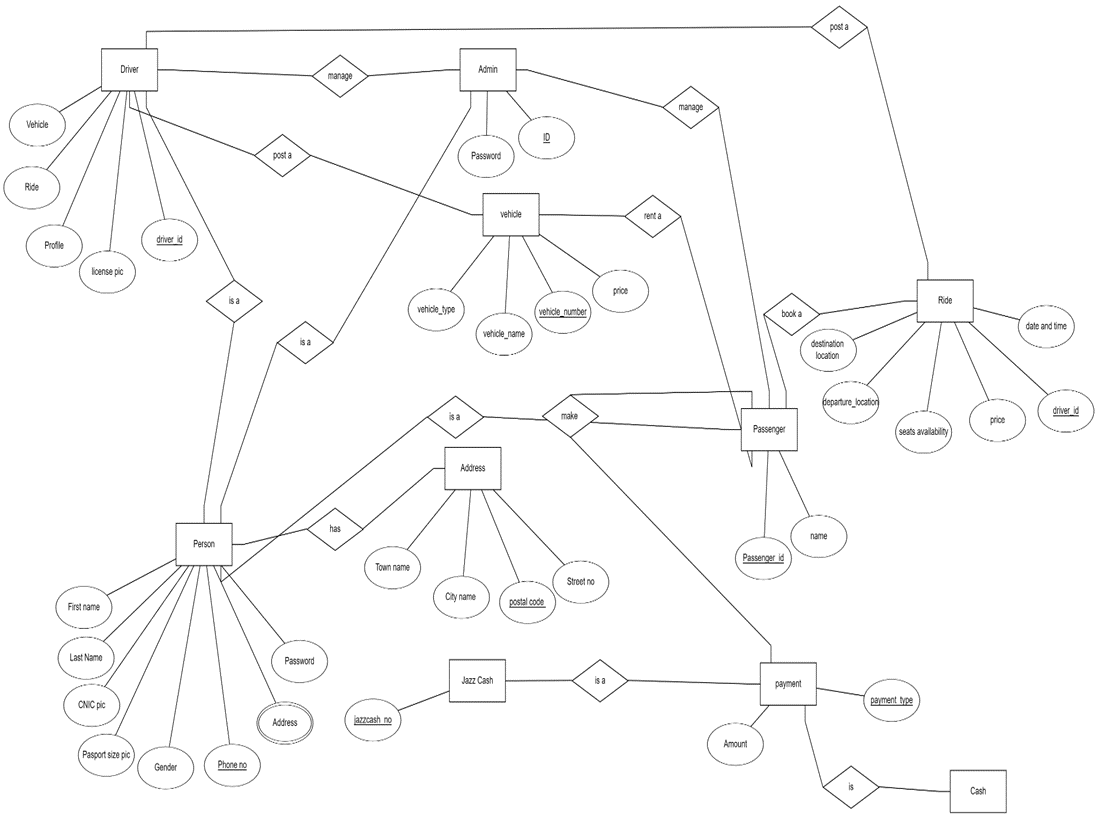
**Software Design Specification**

**System Architecture:**

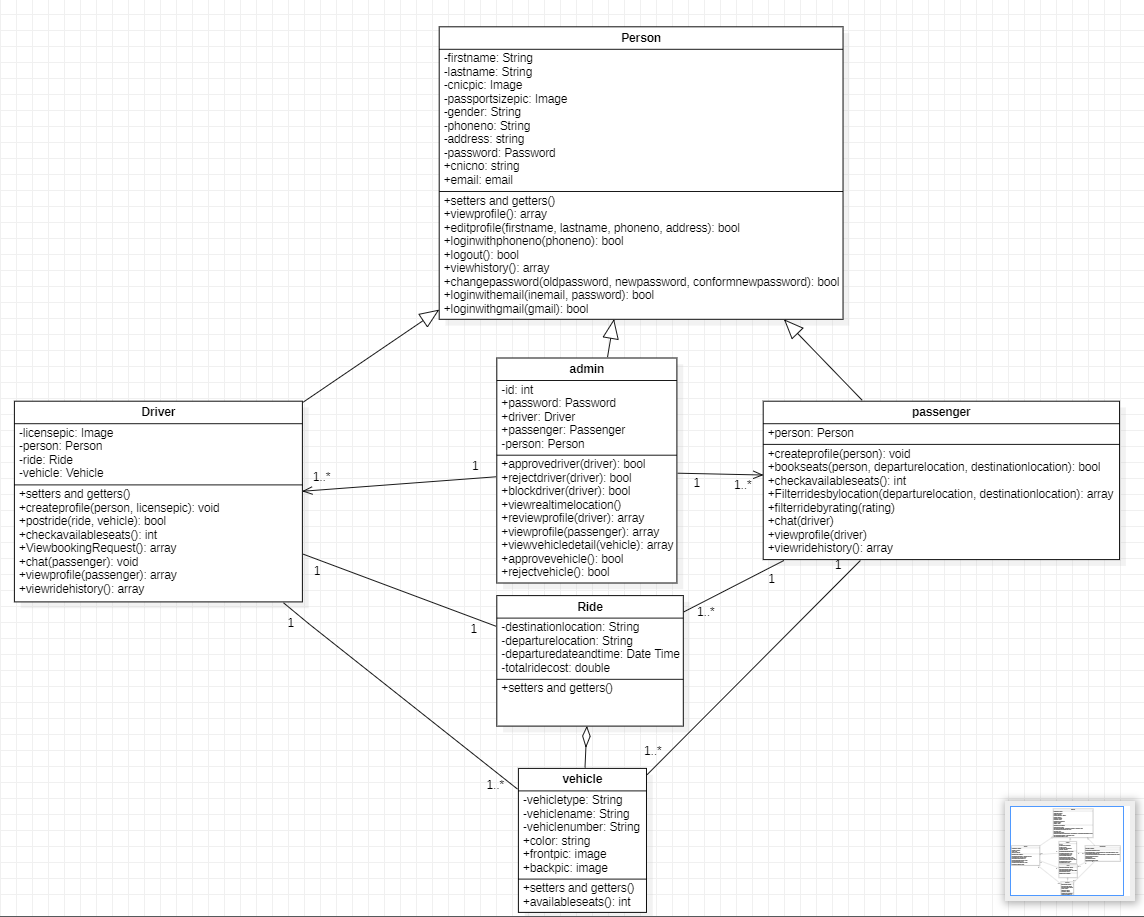
****

**Detailed System Design:**

**ER/EER Diagram:**

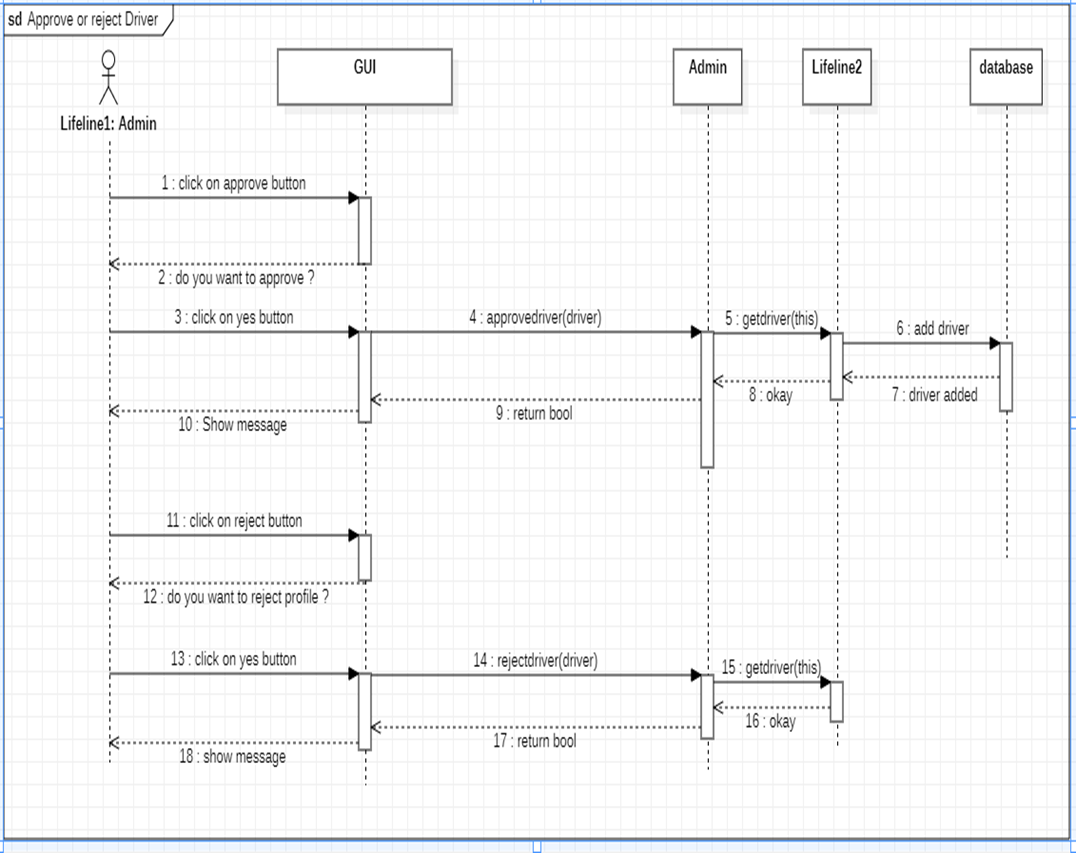
****

**Class Diagram:**

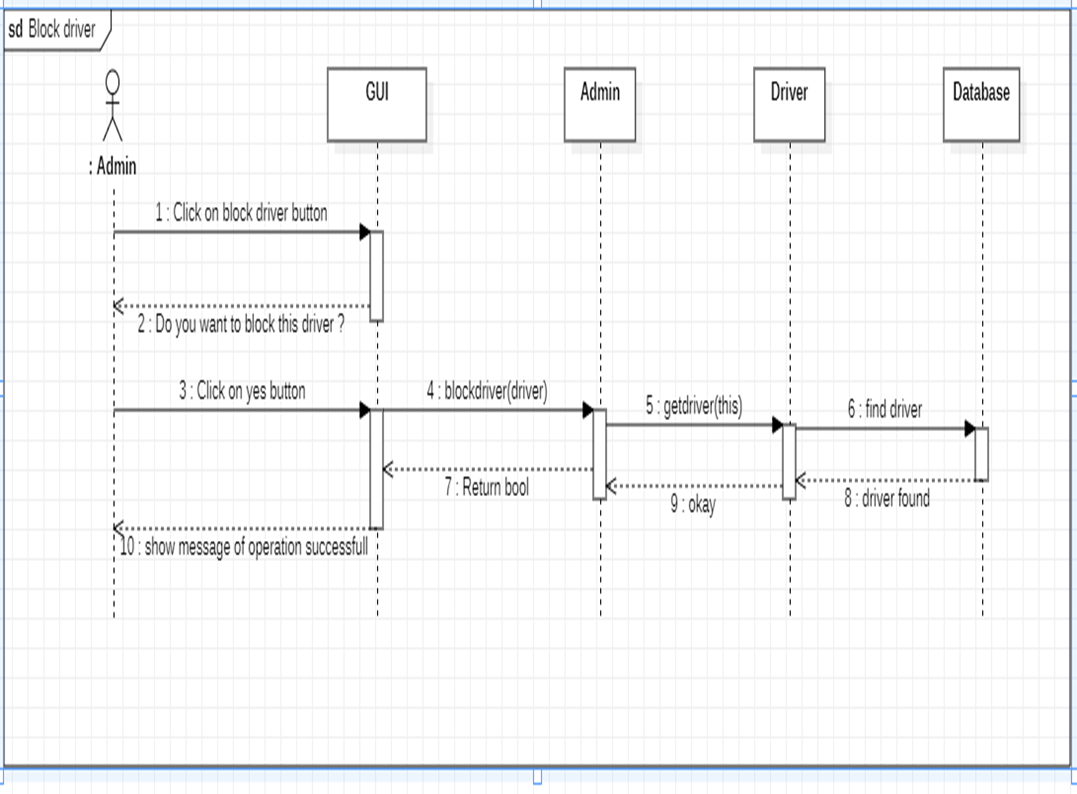
****

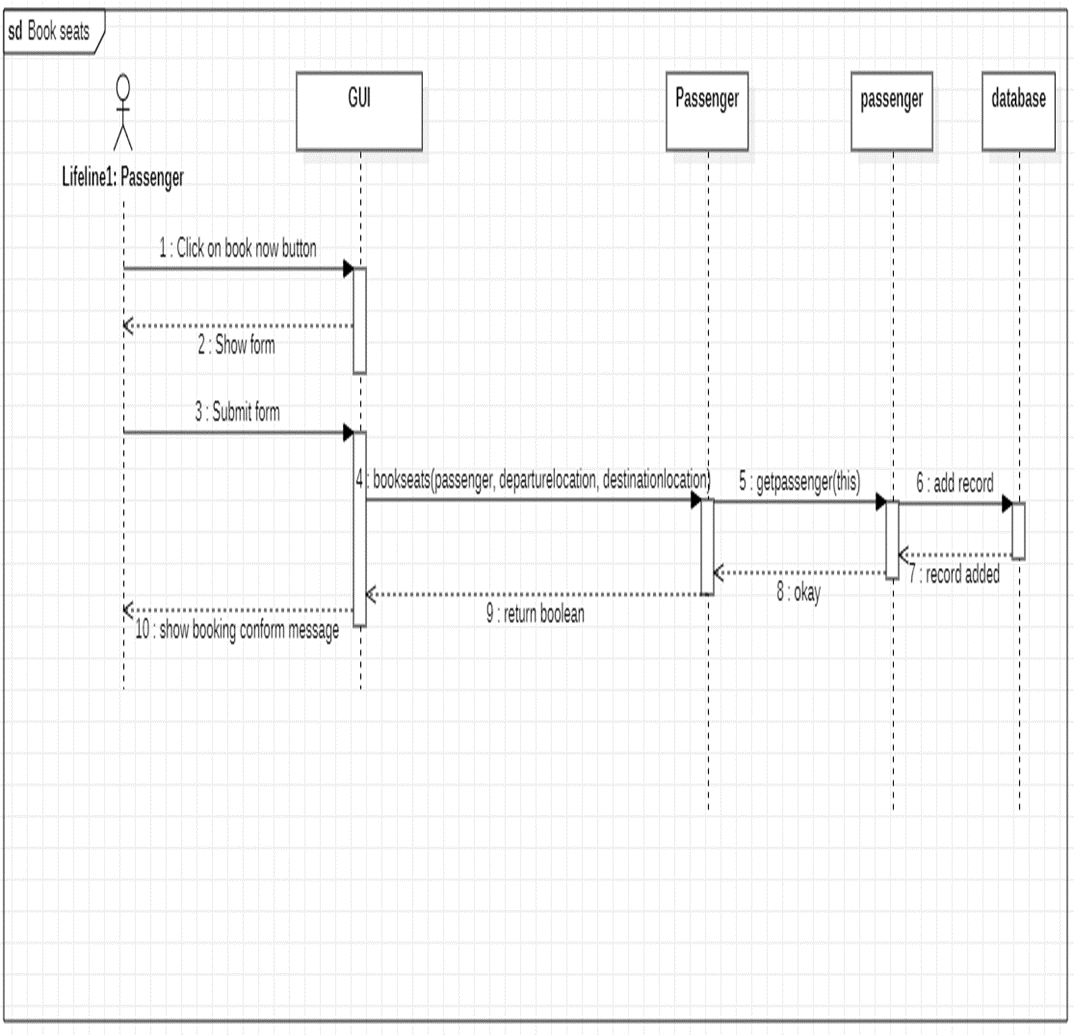
**Sequence diagrams:**

**Approve or reject driver:**

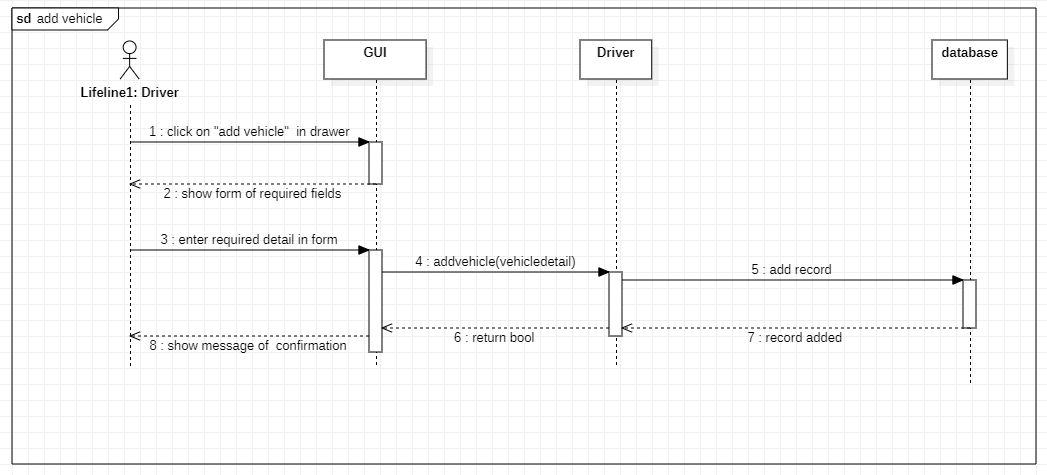
****

**Block Driver:**

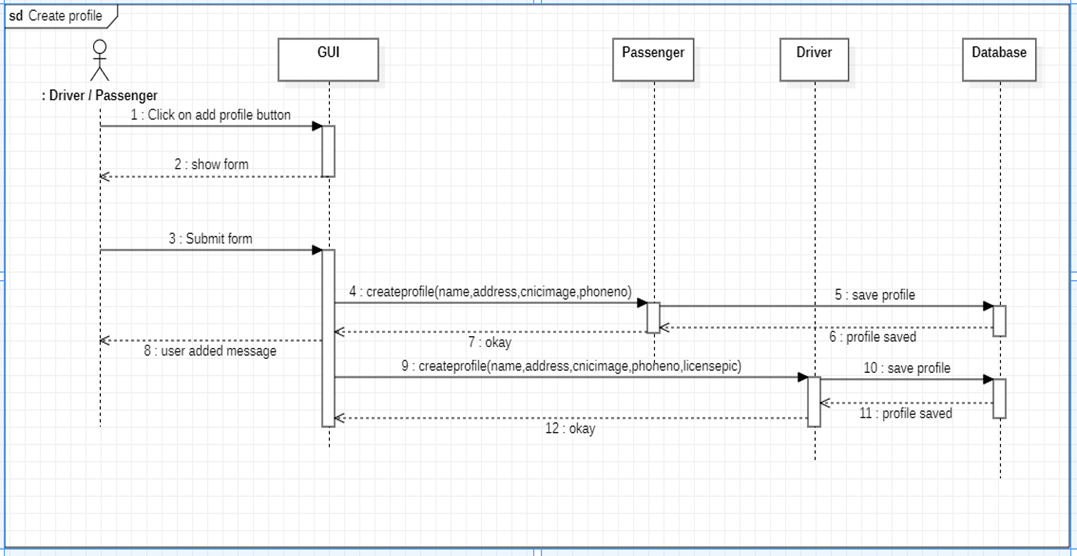
****

**Book Seats:**

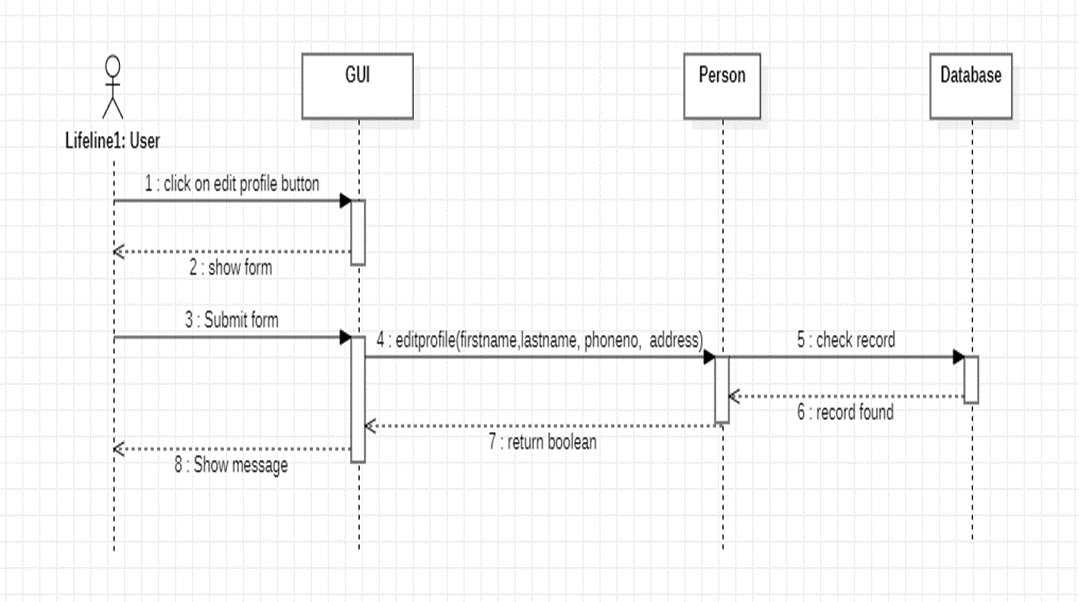
**Add Vehicle :**

****

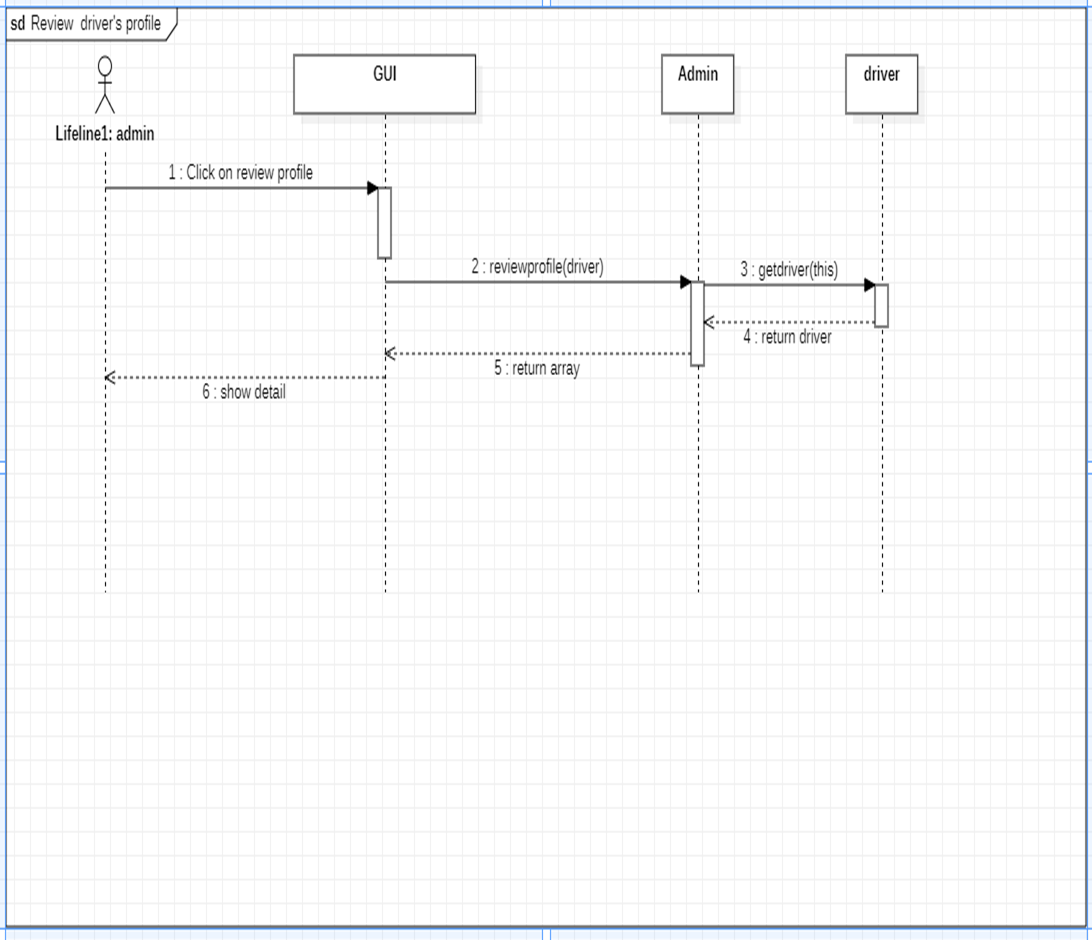
**Create Profile:**

****

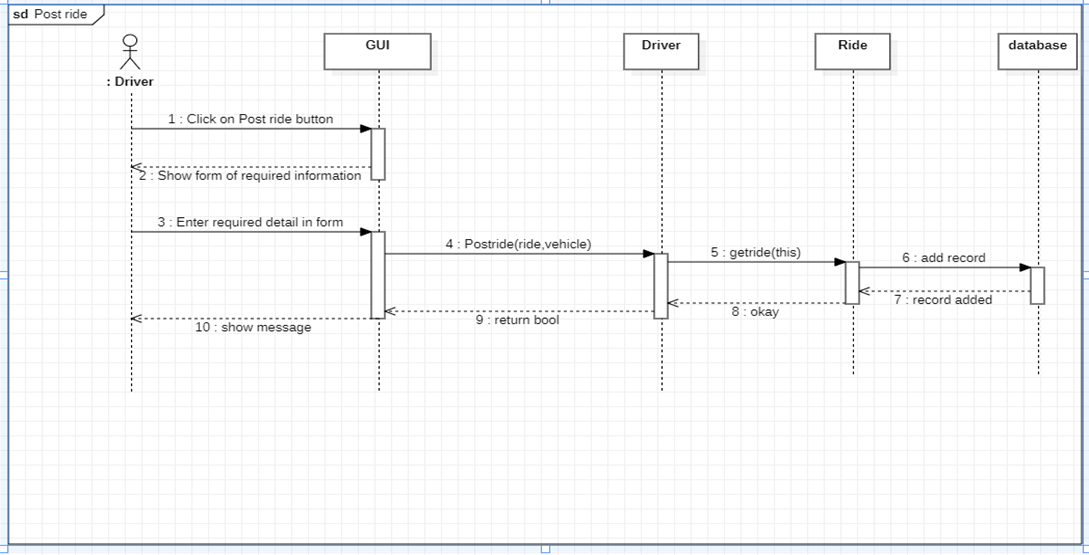
**Edit Profile:**

****

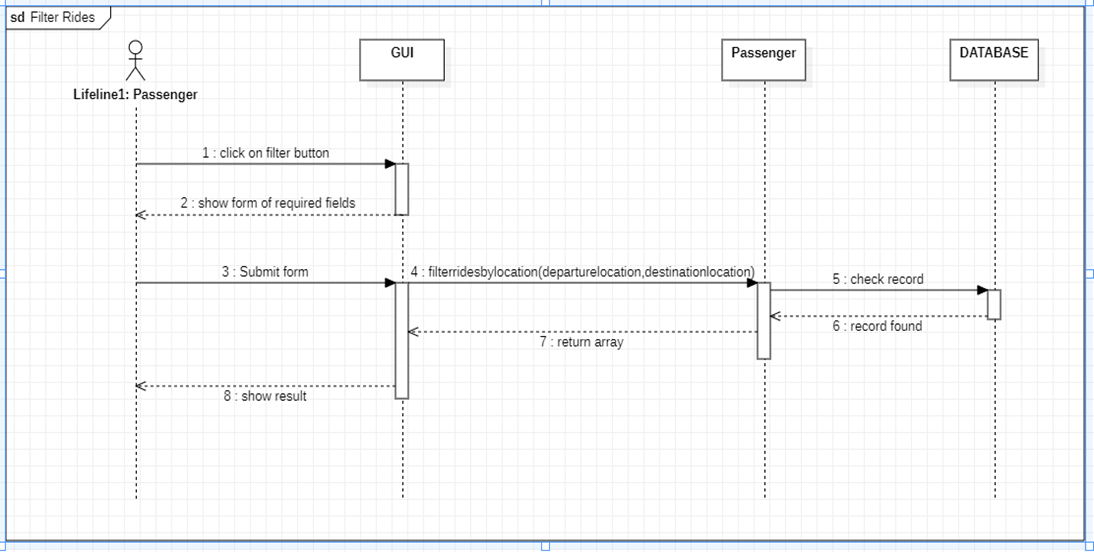
**Review profile:**

****

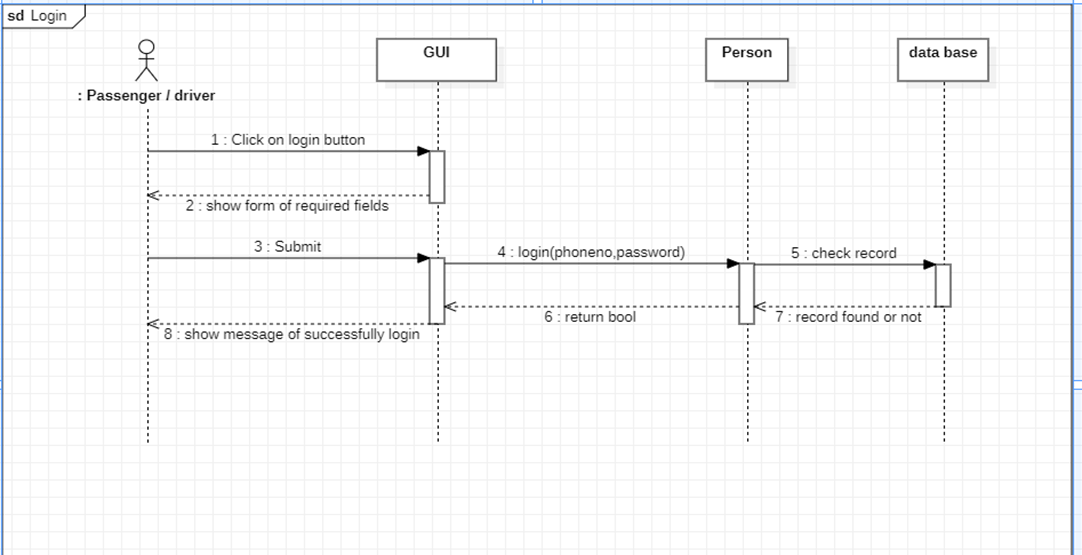
**Post a Ride:**

****

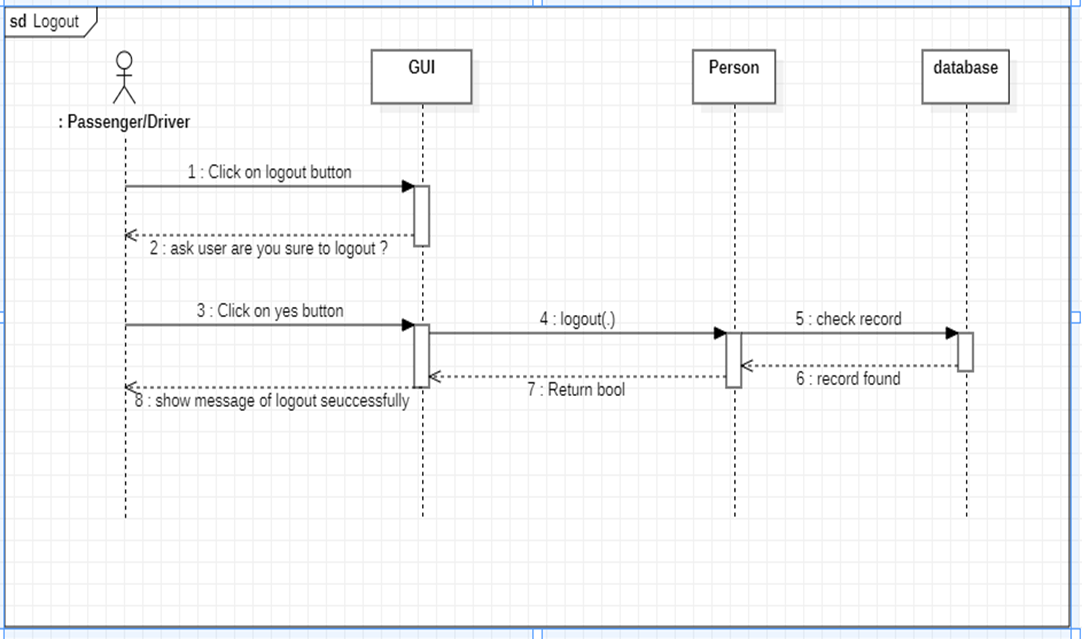
**Filters Ride:**

****

**Login:**

****

**Logout:**

****

**Use Case Scenarios:**

|  |
| --- |
| **4.1--Registration** |

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Scenario** | **Description** |
| **1** | **Priority** | High |
| **2** | **Actor** | Driver, Passenger |
| **3** | **Summary** | This function will allow the user to get registered using his personal details. |
| **4** | **Pre-condition** | Share Ride is installed |
| **5** | **Include** | None |
| **6** | **Extends** | None |
| **7** | **Normal course of Events**  **User Action**  1. User clicks on the login button.    2. Users will fill the registration form. | **Normal course of Events**  **System Response**  1.1-The system will ask the user to continue with the existing account.  1.2-System will check if the user is already registered, if not then it will show the registration panel.  2.1- System will store user details in our  local database and will direct to the  dashboard. |
| **8** | **Post Condition** | The user has entered the system and the system has displayed the dashboard. |
| **9** | **Exception** | Users already exist on our local database. |
| **10** | **Assumption** | User wants to use share ride |
| **11** | **Frequency of Use** | Every time the user will register in the system. |
| **12** | **Alternative Flows** |  |

|  |
| --- |
| **4.2--Login** |

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Scenario** | **Description** |
| **1** | **Priority** | High |
| **2** | **Actor** | Driver, Passenger, Admin |
| **3** | **Summary** | This function will allow the user to login to the system. |
| **4** | **Pre-condition** | * Share Ride is installed. * Users should be registered in the system. |
| **5** | **Include** |  |
| **6** | **Extends** | Logout |
| **7** | **Normal course of Events**  **User Action**  1. User clicks the login button. | **Normal course of Events**  **System Response**  1.1-The system will ask the user to continue with the existing account.  1.2- System will check the address in our local database and will direct to the dashboard. |
| **8** | **Post Condition** | The user has entered the system and the system has displayed the dashboard. |
| **9** | **Exception** | * User address does not exist on our local database. * Connection to database failed. |
| **10** | **Assumption** | Users want to use the system. |
| **11** | **Frequency of Use** | Every time the user will login the system. |
| **12** | **Alternative Flows** | Registration |

|  |
| --- |
| **4.3--Post Ride** |

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Scenario** | **Description** |
| **1** | **Priority** | High |
| **2** | **Actor** | Driver, Passenger |
| **3** | **Summary** | This function will allow the user to post a ride. |
| **4** | **Pre-condition** | * Share Ride is installed. * Users should be registered in the system. |
| **5** | **Include** | * Add pick up location. * Add destination. |
| **6** | **Extends** | * Add payment method * View Passengers profile * Cancel ride |
| **7** | **Normal course of Events**  **User Action**  1. User clicks the ride button.  2. User clicks the check profile button.  3. The user will click the Payment button.  4. User Clicks the pick-up location.  5. User clicks the Destination button.  6. User clicks the cancel button to cancel the ride. | **Normal course of Events**  **System Response**  1.1-The system will ask the user to post a ride.  2.1- The system will show the user profile  3.1-System will show the available payment methods.  4.1-System will show the current location using GPS  5.1-System will ask the user to enter the destination location.  6.1-The system will cancel the ride |
| **8** | **Post Condition** | The user has posted a ride in the system |
| **9** | **Exception** | ● Connectivity error  ● Connection to database failed. |
| **10** | **Assumption** | Users want to post a ride |
| **11** | **Frequency of Use** | Every time the user posts a ride the system. |
| **12** | **Alternative Flows** |  |

|  |
| --- |
| **4.4--View History** |

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Scenario** | **Description** |
| **1** | **Priority** | High |
| **2** | **Actor** | Driver, Passenger, Admin |
| **3** | **Summary** | This function will allow the user to view History. |
| **4** | **Pre-condition** | * Share Ride is installed. * User is logged in. |
| **5** | **Include** |  |
| **6** | **Extends** |  |
| **7** | **Normal course of Events**  **User Action**   1. User clicks the ride history button**.** | **Normal course of Events**  **System Response**  1.1-The system will show the ride history. |
| **8** | **Post Condition** | The user has logged in the system |
| **9** | **Exception** | * User does not have a previous ride. * Connection to database failed. |
| **10** | **Assumption** | Users want to view history of |
| **11** | **Frequency of Use** | Every time the user presses the History Button. |
| **12** | **Alternative Flows** |  |

|  |
| --- |
| **4.5--Seat Availability** |

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Scenario** | **Description** |
| **1** | **Priority** | High |
| **2** | **Actor** | Driver, Passenger |
| **3** | **Summary** | This function will allow the user to view seat availability. |
| **4** | **Pre-condition** | * User is logged in. * User has picked a particular ride. |
| **5** | **Include** |  |
| **6** | **Extends** |  |
| **7** | **Normal course of Events**  **User Action**   1. User clicks the seat availability button. | **Normal course of Events**  **System Response**  1.1- The system will show the vacant seats in the system. |
| **8** | **Post Condition** | The user will view the number of seats. |
| **9** | **Exception** | * Connectivity error. * Connection to database failed. |
| **10** | **Assumption** | Users want to view the vacant seats. |
| **11** | **Frequency of Use** | Every time the user views the availability of seats. |

|  |
| --- |
| **4.6– Edit Profile** |

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Scenario** | **Description** |
| **1** | **Priority** | High |
| **2** | **Actor** | Driver, Passenger |
| **3** | **Summary** | This function will allow the user to edit profiles when they give wrong information. |
| **4** | **Pre-condition** | * User is logged in. * User has picked a particular ride |
| **5** | **Include** |  |
| **6** | **Extends** |  |
| **7** | **Normal course of Events**  **User Action**   1. User clicks the edit profile button. | **Normal course of Events**  **System Response**  1.1- The system will show the edit option to change. |
| **8** | **Post Condition** | The user will change the profile. |
| **9** | **Exception** | * Connectivity error. * Connection to database failed. |
| **10** | **Assumption** | Users want to view the profile. |
| **11** | **Frequency of Use** | Every time the user edits a profile. |
| **12** | **Alternative Flows** |  |

|  |
| --- |
| **4.7--View Profile** |

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Scenario** | **Description** |
| **1** | **Priority** | High |
| **2** | **Actor** | Driver, Passenger |
| **3** | **Summary** | This function will allow the user to view profiles. |
| **4** | **Pre-condition** | * User is logged in. * User has viewed the profile. |
| **5** | **Include** |  |
| **6** | **Extends** |  |
| **7** | **Normal course of Events**  **User Action**   1. User clicks the view profile button. | **Normal course of Events**  **System Response**  1.1- The system will show the profile in the system. |
| **8** | **Post Condition** | The user will view the profile. |
| **9** | **Exception** | * Connectivity error. * Connection to database failed. |
| **10** | **Assumption** | Users want to view the profile. |
| **11** | **Frequency of Use** | Every time the user views the profile. |
| **12** | **Alternative Flows** |  |

|  |
| --- |
| **4.8--Add vehicle** |

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Scenario** | **Description** |
| **1** | **Priority** | High |
| **2** | **Actor** | Driver, Admin |
| **3** | **Summary** | This function will allow the user to enter their vehicle details for the add in system. |
| **4** | **Pre-condition** | * Users are registered. * User is logged in. |
| **5** | **Include** |  |
| **6** | **Extends** |  |
| **7** | **Normal course of Events**  **User Action**   1. User clicks add vehicle button. | **Normal course of Events**  **System Response**  1.1-The system will show the form to add the vehicle details. |
| **8** | **Post Condition** | The user has added the vehicle |
| **9** | **Exception** | * User does not fully fulfill the requirement. * Connection to database failed. |
| **10** | **Assumption** | Users want to add the vehicle . |
| **11** | **Frequency of Use** | Every time the user presses the add vehicle button |
| **12** | **Alternative Flows** |  |

|  |
| --- |
| **4.9--View Booking requests** |

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Scenario** | **Description** |
| **1** | **Priority** | High |
| **2** | **Actor** | Driver |
| **3** | **Summary** | This function will allow the user to view the number of bookings. |
| **4** | **Pre-condition** | · User is logged in. |
| **5** | **Include** |  |
| **6** | **Extends** | View documents, Chat |
| **7** | **Normal course of Events**  **User Action**   1. User clicks the view bookings button. | **Normal course of Events**  **System Response**  1.1-The system will show the ride bookings with the options to verify the details of the person. |
| **8** | **Post Condition** | The user has logged in the system |
| **9** | **Exception** | Connection to database failed. |
| **10** | **Assumption** | Users want to view bookings. |
| **11** | **Frequency of Use** | Every time the user presses the view booking request button. |
| **12** | **Alternative Flows** |  |

|  |
| --- |
| **5.0--View History** |

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Scenario** | **Description** |
| **1** | **Priority** | High |
| **2** | **Actor** | Driver, Passenger, Admin |
| **3** | **Summary** | This function will allow the user to view History. |
| **4** | **Pre-condition** | * Share Ride is installed. * User is logged in. |
| **5** | **Include** |  |
| **6** | **Extends** |  |
| **7** | **Normal course of Events**  **User Action**   1. User clicks the ride history button. | **Normal course of Events**  **System Response**  1.1- The system will show the ride history. |
| **8** | **Post Condition** | The user has logged in the system. |
| **9** | **Exception** | * User does not have a previous ride. * Connection to database failed. |
| **10** | **Assumption** | Users want to view history of |
| **11** | **Frequency of Use** | Every time the user presses the History Button. |
| **12** | **Alternative Flows** |  |

**Test Plan Document**

**Document Information**

|  |  |
| --- | --- |
| **Category** | **Information** |
| Customer | GIFT University |
| Project | <Share ride> |
| Document | Test Plan |
| Document Version | 1.0 |
| Identifier | TEST-PLAN-<Share ride> |
| Status | Final |
| Author(s) | Abubaker Siddique, Rizwan ashraf, Muhammad Ansar, Muhammad umar |
| Approver(s) | Project advisor : <Sumaira Farid> |
| Issue Date | <26-6-2022> |
| Document Location | GIFT University Library |
| Distribution | 1. Advisor  2. PM  3. Project Office |

**1. Introduction**

Share ride is a transport idea based on mobile application that lets you comfortably travel from one city to another city as well as within the city.A per son who has to travel from one city to another city has to pay amount of a complete ride (vehicle). In this application we are giving a solution where the passenger will book the seats that he needs and pay only for these seats. If passenger has need only one seat then he has no need to pay for whole vehicle. In this application if the driver has to move from one place to another for his personal work he has free seats in his vehicle, he can post to our application the time of departure and the number of seats available. Registered passengers can book seats accordingly to their need.

**1.1. Purpose of Document**

This document helps in keeping the track of the functioning of the entire system and also improves on the quality of the software product. All members of the testing team utilize this document to check that the system is working as planned and to verify the overall system's functionality. This document's goal is to ensure that the proposed system is error-free.

**1.2. Project Overview**

When the system is developed, it will facilitate the customers and admin of the “Share ride App”.

**2. Scope of Testing**

The following features will be tested:

∙ Register

∙ Login

∙ View Profile

∙ Update Profile

∙ edit profile

∙ post ride

∙ cancel ride

∙ view history

∙ approve driver

∙ Block driver

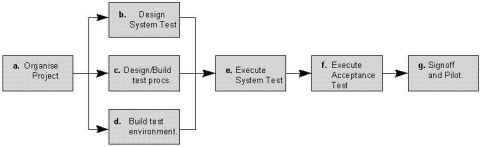
∙ seats availability

∙ View booking

**3. Test Plan Strategy**

*This test plan's goal is to guarantee that everyone who reads it has a shared knowledge of the test technique, as well as the amount of resources, timelines, etc.*

*The test plan procedure is shown in the image below.*

**

**Unit Testing**

**Definition:**

Unit Test consists of testing individual programs or subroutines as they are written instead of testing the entire system after it has been written. The testing of the smaller building blocks is done first and then these blocks are combined and tested. Unit testing means testing each function independently to verify correct processing in a stand-alone environment.

Unit testing is ideally developed by following the simple steps.

● Writing required test cases with inputs and correct outputs.

● Arrange: setup testing modules and prepare the system for testing.

● Act: performing test

● Asserts: Verify the result pass or fail.

UNIT TESTING unit testing is the testing of an individual software module. It is usually completed by the programmer. Detailed knowledge of external program design and code is required. i.e. they can never depend on any external factors.

**Participants***:*

Abubaker Siddique

Rizwan ashraf

Muhammad Ansar

Muhammad umar

**Methodology:**

a. The initial unit test will be performed simultaneously with development.

b. The test phase will begin when the completed modules are developed, using Unit Tests on these modules.

c. The consistency of each individual module with its original goal will be evaluated.

**Integration Testing**

**Definition:**

*The stage of software testing when individual software modules are integrated and tested as a group is known as integration testing (sometimes known as integration and testing, abbreviated "I&T"). Before system testing and following unit testing, it happens. The goal of integration testing is to create an integrated system that is suitable for system testing by taking as input modules that have undergone unit testing, grouping them into bigger aggregates, and applying the tests outlined in an integration test plan to those aggregates.*

*It is said that integration tests are longer to execute because they often involve reading and writing to a database. The function of a class login that are dependent on a local database*

**Participants:**

Abubaker Siddique

Rizwan ashraf

Muhammad Ansar

Muhammad umar

**Methodology:**

a. All the components are integrated at this phase.

b. After that, various inputs are applied to the integrated modules, and their output is considered carefully.

c. The modules are revised if there are any inconsistencies.

**System Testing**

**Definition:**

*Software system testing is testing done on a complete, integrated system to see whether the system complies with the requirements. Black box testing includes system testing, which should not call for any understanding of the logic or code's internal structure. System testing often uses the software system itself combined with any suitable hardware system as well as any of the "integrated" software components that have passed integration testing as its input. The goal of integration testing is to find any discrepancies between any of the assemblages—groups of integrated software units—or any of the assemblages and the hardware. System testing is a more constrained sort of testing that looks for flaws both inside and outside the system.*

**Participants:**

Abubaker Siddique

Rizwan ashraf

Muhammad Ansar

Muhammad umar

**Methodology:**

a) In this phase, all integrated modules are joined into a single system.

b) Additionally, the system as a whole is provided various inputs and its relevant outputs are taken into account.

c) The system is examined in the context of its interconnected modules in the event that it produces any undesirable outputs.

**4. Test Environment**

*Users must have a stable internet connection. No hardware is needed during project testing other than an iOS and Android device. The testing team needs AR supported Android and iOS servers to execute the system in order to set up the testing environment.*

**5. Schedule**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Testing Activities Begin End Person Responsible** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Designing Test Cases 15-09-2022 21-06-2022 Abubaker Siddique, Rizwan ashraf Executing Test Cases 10-79-2022 21-07-2022 Muhamad ansar, Muhammad umar Unit Testing 11-07-2022 21-07-2022 Abubaker siddique Integration Testing 20-07-2022 21-07-2022 Rizwan ashraf System Testing 21-07-2022 21-07-2022 Muhammad ansar* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6. Control Activities**

09-06-22 Forms are re-designed and their validity is discussed

10-06-22 Project schedule and Design Test Cases are discussed in meeting 11-06-22 ERD is re-designed and project work is divided among TM

15-06-22 Database of the whole system is re-designed in university

17-06-22 Divided forms among TM are integrated in Project

19-06-22 Database diagram discussed in meeting and test case execution started 20-06-22 Forms are shown and discussed in Project meeting

21-06-22 Test Plan is submitted

*.*

**7. Resources**

*This section contains the information regarding resources in terms of human, hardware and software.*

***7.1) Human***

*All team members fully participated in the whole work of the project, including designing, implementing, integrating and testing.*

***7.2) Hardware***

*No hardware was used in entire project testing.*

***7.3) Software***

*No software was used in entire project testing.*

**8. Test Case Design and Description**

**1.Register**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | *10* | | | | | |
| **QA Test Engineer** | | [Abu Baker Siddique](mailto:181370091@gift.edu.pk) | | **Reviewed By** | | *Dr. Sumaira Farid* | |
| **Test case Version:** | | *1.0* | |  | | | |
| **Test Execution Date:** | | *16-06-2022* | | | | | |
| **Use Case Reference(s)** | | *UC-01* | | | | | |
| **GUI Reference(s)** | | *.* | | | | | |
| **Objective** | | *To successfully Register* | | | | | |
| **Product / Version/ Module** | | *Version 1.0* | | | | | |
| **Environment:** | | *The system should be properly connected to the internet.* | | | | | |
| **Assumptions:** | | *The Web server and database server should be alive.* | | | | | |
| **Prerequisite:** | | *Students should be logged into the GCMS and the Student will click submit assignment.* | | | | | |
| **Test Case Description** | | This test case will contain all the scenarios against which a user should be registered. | | | | | |
| **Input Parameters** | **Expected Output** | | **Actual Output** | | **Test Conformance Status** | | **Possible Reason(s) in case of failure** |
| User Enters name, valid email address, phone number, dob, password and confirms  password. | User forgets to enter a required field.  . | | The System generates an error “Please fill  out all fields” | | Passed | | The case is Passed. |
| User Enters name, valid email address, phone number, dob, password and confirms  password. | User gets  registered in the  system. | | User gets  registered in the  system. | | Passed | | The case is Passed. |

**2.Login**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | *02* | | | | | |
| **QA Test Engineer** | | *Rizwan Ashraf* | | **Reviewed By** | | *Dr. Sumaira Farid* | |
| **Test case Version:** | | *1.0* | |  | | | |
| **Test Execution Date:** | | *16-06-2022* | | | | | |
| **Use Case Reference(s)** | | *UC-02* | | | | | |
| **GUI Reference(s)** | | *.* | | | | | |
| **Objective** | | *To successfully Login into the system* | | | | | |
| **Product / Version/ Module** | | *Version 1.0* | | | | | |
| **Environment:** | | *The system should be properly connected to the internet.* | | | | | |
| **Assumptions:** | | *The Web server and database server should be alive.* | | | | | |
| **Prerequisite:** | | *Students should be logged into the GCMS and the Student will click submit assignment.* | | | | | |
| **Test Case Description** | | This test case will contain all the scenarios against which a user should be logged in. | | | | | |
| **Input Parameters** | **Expected Output** | | **Actual Output** | | **Test Conformance Status** | | **Possible Reason(s) in case of failure** |
| “John” is entered into the USERNAME field and  “guess1234” is entered into the PASSWORD field. | System logs the user John into the system.  . | | The System successfully logs the user john into the system. | | Passed | | The case is Passed. |
| “Nick” is entered into the USERNAME field and  “guess1234” is entered into the PASSWORD field. | System should not log the user into the system | | System displays an error message indicating that the USERNAME or PASSWORD field is incorrect. | | Passed | | The case is Passed. |
| USERNAME and  PASSWORD fields are left blank. | System should not log the user into the system | | System displays an error message indicating that the USERNAME or PASSWORD field is incorrect. | | Passed | | The case is Passed |

**3. View Profile**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | *03* | | | | | |
| **QA Test Engineer** | | *Muhammad Umer* | | **Reviewed By** | | *Dr. Sumaira Farid* | |
| **Test case Version:** | | *1.0* | |  | | | |
| **Test Execution Date:** | | *16-06-2022* | | | | | |
| **Use Case Reference(s)** | | *UC-05* | | | | | |
| **GUI Reference(s)** | | *.* | | | | | |
| **Objective** | | *To successfully allow the user to view his/her profile.* | | | | | |
| **Product / Version/ Module** | | *Version 1.0* | | | | | |
| **Environment:** | | *The system should be properly connected to the internet and the user must be logged in.* | | | | | |
| **Assumptions:** | | *The Web server and database server should be alive.* | | | | | |
| **Prerequisite:** | | *The user is logged in into the system.* | | | | | |
| **Test Case Description** | | This test case will contain all the scenarios against which a user view his/her profile.  . | | | | | |
| **Input Parameters** | **Expected Output** | | **Actual Output** | | **Test Conformance Status** | | **Possible Reason(s) in case of failure** |
| User clicks on the View Profile button. | Users shall be navigated to the profile page.  . | | User is  navigated to the profile page | | Passed | | Poor Internet connection. The server is down. |

**4.Update Profile**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | *04* | | | | | |
| **QA Test Engineer** | | [Abu Baker Siddique](mailto:181370091@gift.edu.pk) | | **Reviewed By** | | *Dr. Sumaira Farid* | |
| **Test case Version:** | | *1.0* | |  | | | |
| **Test Execution Date:** | | *16-06-2022* | | | | | |
| **Use Case Reference(s)** | | *UC-06* | | | | | |
| **GUI Reference(s)** | | *.* | | | | | |
| **Objective** | | *To successfully allow a user to update his/her profile.* | | | | | |
| **Product / Version/ Module** | | *Version 1.0* | | | | | |
| **Environment:** | | *The system should be properly connected to the internet and the user must be logged in.* | | | | | |
| **Assumptions:** | | *The Web server and database server should be alive.* | | | | | |
| **Prerequisite:** | | *The user is already logged in into the system and on the update profile page.* | | | | | |
| **Test Case Description** | | This test case will contain all the scenarios against which a user updates the profile. | | | | | |
| **Input Parameters** | **Expected Output** | | **Actual Output** | | **Test Conformance Status** | | **Possible Reason(s) in case of failure** |
| User has  enter new data values and click on save changes button. | The new data will be updated in database and system will  display a  confirmation message.  . | | The new data is updated in  database and system will  display a  confirmation message | | Passed | | The case is Passed. |
| User  enter the same old data values and click on save changes button. | System should display an error message “Please Enter the new data”. | | The new data will be updated in database and system will  display a  confirmation message | | Passed | | The case is Passed. |

**5.Post a Ride**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | *05* | | | | | |
| **QA Test Engineer** | | [Abu Baker Siddique](mailto:181370091@gift.edu.pk) | | **Reviewed By** | | *Dr. Sumaira Farid* | |
| **Test case Version:** | | *1.0* | |  | | | |
| **Test Execution Date:** | | *16-06-2022* | | | | | |
| **Use Case Reference(s)** | | *UC-07* | | | | | |
| **GUI Reference(s)** | | *.* | | | | | |
| **Objective** | | *To successfully allow a user to post a ride.* | | | | | |
| **Product / Version/ Module** | | *Version 1.0* | | | | | |
| **Environment:** | | *The system should be properly connected to the internet and the user must be logged in.* | | | | | |
| **Assumptions:** | | *The Web server and database server should be alive.* | | | | | |
| **Prerequisite:** | | *The user is already logged in into the system and clicked the post ride button.* | | | | | |
| **Test Case Description** | | This test case will contain all the scenarios against which a user can post a ride.. | | | | | |
| **Input Parameters** | **Expected Output** | | **Actual Output** | | **Test Conformance Status** | | **Possible Reason(s) in case of failure** |
| User will clicked the post ride button  And enter the ride’s details | The System will display a message “Ride Posted Successfully”.  . | | The System will display a message “Ride Posted Successfully”. | | Passed | | The case is Passed. |

**6.View Ride History**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | *06* | | | | | |
| **QA Test Engineer** | | [Abu Baker Siddique](mailto:181370091@gift.edu.pk) | | **Reviewed By** | | *Dr. Sumaira Farid* | |
| **Test case Version:** | | *1.0* | |  | | | |
| **Test Execution Date:** | | *16-06-2022* | | | | | |
| **Use Case Reference(s)** | | *UC-08* | | | | | |
| **GUI Reference(s)** | | *.* | | | | | |
| **Objective** | | *To successfully allow a user to view history.* | | | | | |
| **Product / Version/ Module** | | *Version 1.0* | | | | | |
| **Environment:** | | *The system should be properly connected to the internet and the user must be logged in.* | | | | | |
| **Assumptions:** | | *The Web server and database server should be alive.* | | | | | |
| **Prerequisite:** | | *The user is already logged in into the system and clicked the view history button.* | | | | | |
| **Test Case Description** | | This test case will contain all the scenarios against which a user can view history. | | | | | |
| **Input Parameters** | **Expected Output** | | **Actual Output** | | **Test Conformance Status** | | **Possible Reason(s) in case of failure** |
| User has clicked the view history button. | System will display a  Complete details of the ride's history.  . | | System will display a  Complete details of the ride's history. | | Passed | | The case is Passed. |

**7.Chat**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | *07* | | | | | |
| **QA Test Engineer** | | *Rizwan Ashraf* | | **Reviewed By** | | *Dr. Sumaira Farid* | |
| **Test case Version:** | | *1.0* | |  | | | |
| **Test Execution Date:** | | *16-06-2022* | | | | | |
| **Use Case Reference(s)** | | *UC-09* | | | | | |
| **GUI Reference(s)** | | *.* | | | | | |
| **Objective** | | *To successfully allow a user to send a message to another user.* | | | | | |
| **Product / Version/ Module** | | *Version 1.0* | | | | | |
| **Environment:** | | *The system should be properly connected to the internet and the user must be logged in.* | | | | | |
| **Assumptions:** | | *The Web server and database server should be alive.* | | | | | |
| **Prerequisite:** | | *The user is already logged in into the system.* | | | | | |
| **Test Case Description** | | This test case will contain all the scenarios against which a user can send a message to another user. | | | | | |
| **Input Parameters** | **Expected Output** | | **Actual Output** | | **Test Conformance Status** | | **Possible Reason(s) in case of failure** |
| User clicks on the chat Button. | The system shall display a keypad and allow a user to send a message. | | The system shall display a keypad and allow a user to send a message. | | Passed | | The case is Passed. |
| User clicks on the chat Button. | The system shall display a keypad and allow a user to send a message. | | The system shall display a keypad and allow a user to send a message. | | Passed | | Poor Internet connection. The server is down. |

**8. Approve User**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | *07* | | | | | |
| **QA Test Engineer** | | *Rizwan Ashraf* | | **Reviewed By** | | *Dr. Sumaira Farid* | |
| **Test case Version:** | | *1.0* | |  | | | |
| **Test Execution Date:** | | *16-06-2022* | | | | | |
| **Use Case Reference(s)** | | *UC-10* | | | | | |
| **GUI Reference(s)** | | *.* | | | | | |
| **Objective** | | *To successfully allow an admin to approve the profile of new user.* | | | | | |
| **Product / Version/ Module** | | *Version 1.0* | | | | | |
| **Environment:** | | *The system should be properly connected to the internet and the user must be logged in..* | | | | | |
| **Assumptions:** | | *The Web server and database server should be alive.* | | | | | |
| **Prerequisite:** | | *The user is already logged in into the system as an admin.* | | | | | |
| **Test Case Description** | | This test case will contain all the scenarios against which an admin can approve the profile of another new user. | | | | | |
| **Input Parameters** | **Expected Output** | | **Actual Output** | | **Test Conformance Status** | | **Possible Reason(s) in case of failure** |
| Admin click on the approve button. | The system shall add the user. | | The system shall add the user. | | Passed | | The case is Passed. |

**9. Block user**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID:** | | *09* | | | | | |
| **QA Test Engineer** | | [Abu Baker Siddique](mailto:181370091@gift.edu.pk) | | **Reviewed By** | | *Dr. Sumaira Farid* | |
| **Test case Version:** | | *1.0* | |  | | | |
| **Test Execution Date:** | | *16-06-2022* | | | | | |
| **Use Case Reference(s)** | | *UC-11* | | | | | |
| **GUI Reference(s)** | | *.* | | | | | |
| **Objective** | | *To successfully allow an admin to block users.* | | | | | |
| **Product / Version/ Module** | | *Version 1.0* | | | | | |
| **Environment:** | | *The system should be properly connected to the internet and the user must be logged in.* | | | | | |
| **Assumptions:** | | *The Web server and database server should be alive.* | | | | | |
| **Prerequisite:** | | *The user is already logged in into the system and added multiple users in the system.* | | | | | |
| **Test Case Description** | | This test case will contain all the scenarios against which an admin can block the user.  . | | | | | |
| **Input Parameters** | **Expected Output** | | **Actual Output** | | **Test Conformance Status** | | **Possible Reason(s) in case of failure** |
| Admin click on the remove or block button. | The System should block the user and remove all data from the database. | | The System should block the user and remove all data from the database. | | Passed | | The case is Passed. |

**Test Case Scenario (TCS)**

|  |  |  |
| --- | --- | --- |
| **ID** | **Description** | **Responsible** |
| TCS-01 | User registration | Abubaker Siddique |
| TCS-02 | User login. | Abubaker Siddique |
| TCS-03 | Save Password | Muhammad Ansar |
| TCS-04 | Update Profile | Abubaker Siddique |
| TCS-05 | View Profile | Abubaker siddique |
| TCS-06 | Post Ride | Muhammad Umer |
| TCS-07 | View history | Rizwan Ashraf |
| TCS-08 | Send Message | Rizwan Ashraf |
| TCS-09 | approve User | Rizwan Ashraf |
| TCS-10 | Block User | Muhammad ansar |
| TCS-11 | Seat Availability | Muhammad ansar |
| TCS-12 | View Bookings | Muhammad ansar |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User registration (Positive)** | | | | |
| **Test Case scenario:**TCS-01 | | | | |
| **Test Case ID:** | TC-01 | | | |
| **Description:** | Users shall be able to register as new users by giving his google account details.  Additional information such as name, email, password and confirm password. | | | |
| **Pre-condition:** | User is already registered with google | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Sta Remarks** |
| 1. Click on the register button  2.Enter google account  3. Input name, email, password and  confirm password. | Name=Ansar  email=ansar12@gmail.com  Password: Password123#  Confirm Password=Password123# | Users should be registered. | User  registered  as a new  user. | Pass |
| **Post Condition** | User is successfully registered. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User registration(Negative**) | | | | |
| **Test Case scenario** TCS-02 | | | | |
| **Test Case ID:** | TC-01 | | | |
| **Description:** | Users will not be able to register on invalid Google account. | | | |
| **Pre-condition:** | User is already registered with google.. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Sta Remarks** |
| 1. Click on the register button  2.Enter google account  3. Input name, email, password and  confirm password. | Name= Ansar  email=ansar12@gmail.com  Password: Password123#  Confirm Password=Password123# | Users should be registered. | User is not registered as a new user. | Fail |
| **Post Condition** | User does not log in successfully. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User registration(Negative)** | | | | |
| **Test Case scenario TCS-03** | | | | |
| **Test Case ID:** | **TC-01** | | | |
| **Description:** | **Users will not be able to register on empty text fields.** | | | |
| **Pre-condition:** | **User is already registered with google.** | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the register button  2.Enter google account  3. Input name, email, password and  confirm password. | Name=  email=  Password:  Confirm Password= | Users should be registered. | User is not registered as a new user. | Fail |
| **Post Condition** | **User do not register successfully.** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User registration(Negative)** | | | | |
| **Test Case scenario** TCS-04 | | | | |
| **Test Case ID:** | TC-01 | | | |
| **Description:** | Users will not be able to register on invalid email format. | | | |
| **Pre-condition:** | User is already registered with Google. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the register button  2.Enter google account  3. Input name, email, password and  confirm password. | Name= Ansar  email=ansar12@gmail.com.pk  Password: Password123#  Confirm Password=Password123# | Users should be registered. | User is not registered as a new user. | Fail |
| **Post Condition** | User is not successfully registered. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User registration(Negative)** | | | | |
| **Test Case scenario** TCS-05 | | | | |
| **Test Case ID:** | TC-01 | | | |
| **Description:** | Users will not be able to register if the password does not match. | | | |
| **Pre-condition:** | User is already registered with Google. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the register button  2.Enter google account  3. Input name, email, password and  confirm password. | Name= Ansar  email=ansar12@gmail,com Password: Password123#  Confirm Password=password1343# | Users should be registered. | User is not registered as a new user. | Fail |
| **Post Condition** | User is not successfully registered. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User registration(Negative)** | | | | |
| **Test Case scenario TCS-07** | | | | |
| **Test Case ID:** | **TC-01** | | | |
| **Description:** | Users will not be able to register if password length is less than 8 digits and if the password does not include a special character and some numbers. | | | |
| **Pre-condition:** | User is already registered with Google. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the register button  2.Enter google account  3. Input name, email, password and  confirm password. | Name= Qaisar Javaid  email=ansar12@gmail.com  Password: abc123  Confirm Password=abc123 | Users should be registered. | User is not  registered  as a new  user. | Fail |
| **Post Condition** | **User is not successfully registered.** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Login(Positive)** | | | | |
| **Test Case scenario** TCS-01 | | | | |
| **Test Case ID:** | TC-02 | | | |
| **Description:** | Users will be able to login with credentials. | | | |
| **Pre-condition:** | User is already registered with Google. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on username field 2. Enter username and password | Username: ansar12@gmail.com  Password : Password123# | Users should be logged in. | User is logged in.. | Pass |
| **Post Condition** | User is successfully Logged in. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User login(Negative)** | | | | |
| **Test Case scenario** TCS-02 | | | | |
| **Test Case ID:** | TC-02 | | | |
| **Description:** | Users will not be able to log in on invalid username and password | | | |
| **Pre-condition:** | User is already registered with Google. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the username and password field 2. Enter username and password | Username: ansarali  Password : Password | Users should be logged in. | User is not Logged in. | Pass |
| **Post Condition** | User is not successfully logged in. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User login(Negative) | | | | |
| **Test Case scenario** TCS-03 | | | | |
| **Test Case ID:** | TC-02 | | | |
| **Description:** | Users will not be able to login on empty text fields. | | | |
| **Pre-condition:** | User is already registered with Google.. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the username and password field 2. Enter username and password | Username=  Password= | Users should be logged in. | User is not  Logged in. | Pass |
| **Post Condition** | User is not successfully logged in. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User login(Negative)** | | | | |
| **Test Case scenario** TCS-04 | | | | |
| **Test Case ID:** | TC-02 | | | |
| **Description:** | Password field is either visible or an asterisk or bullet sign. | | | |
| **Pre-condition:** | User is already registered with Google. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the username and password field 2. Enter username and password | Username: ansar12@gmail.com  Password : Password123# | Password field is visible or asterisk or bullet sign | Password is a bullet sign. | Pass |
| **Post Condition** | User is not successfully logged in. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Save Password(Positive)** | | | | |
| **Test Case scenario** TCS-01 | | | | |
| **Test Case ID:** | TC-03 | | | |
| **Description:** | Users will be able to save passwords for future use. | | | |
| **Pre-condition:** | User is already registered with Google. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the username and password field 2. Enter username and password | Username: ansar12@gmail.com  Password : Password123#  Save Password | Password should be save on google account | Password is successfully saved | Pass |
| **Post Condition** | Password is successfully saved | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ForgotPassword(Positive)** | | | | |
| **Test Case scenario** TCS-04 | | | | |
| **Test Case ID:** | TC-10 | | | |
| **Description:** | Users will be able to use forgot password functionality. | | | |
| **Pre-condition:** | User is already registered with SmarterChat. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the forgot password button. 2. Enter a registered email. | Username: ansar12@gmail.com | System should generate a reset password email to the user. | Reset password email is sent | Pass |
| **Post Condition** | Password is successfully changed. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Update Profile(Positive)** | | | | |
| **Test Case scenario** TCS-04 | | | | |
| **Test Case ID:** | TC-01 | | | |
| **Description:** | Users will be able to edit his profile information. | | | |
| **Pre-condition:** | User is logged in. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the edit profile button.  2.Input name, image,email, password and  confirm password. | Username: ansar12@gmail.com  Password : Password123# | Users should be able to edit profile. | User has changed his profile information. | Pass |
| **Post Condition** | User has changed his profile information. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Update Profile(Negative)** | | | | |
| **Test Case scenario** TCS-02 | | | | |
| **Test Case ID:** | TC-04 | | | |
| **Description:** | Users will not be able to edit his profile information if he left empty text fields. | | | |
| **Pre-condition:** | User is logged in. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the edit profile button.  2.Input name, image,email, password and  confirm password. | Name:  email:  Password:  Confirm Password: | User should be able to edit profile. | User has not changed his profile information. | Fail |
| **Post Condition** | User has not changed his profile information. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logout(Positive)** | | | | |
| **Test Case scenario** TCS-01 | | | | |
| **Test Case ID:** | TC-05 | | | |
| **Description:** | Users will be able to logout | | | |
| **Pre-condition:** | User is logged in. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Status Remarks** |
| 1. Click on the logout button. |  | User will be able to logout. | User has logged out successfully | Pass |
| **Post Condition** | User has logged out successfully | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logout(Negative)** | | | | |
| **Test Case scenario** TCS-02 | | | | |
| **Test Case ID:** | TC-05 | | | |
| **Description:** | Users will not be able to logout | | | |
| **Pre-condition:** | User is logged in. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Status Remarks** |
| 1. Click on the logout button. |  | User will be able to logout. | User hasn't logged out successfully | Fail |
| **Post Condition** | User has not logged out successfully | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Post Ride(Positive)** | | | | |
| **Test Case scenario** TCS-01 | | | | |
| **Test Case ID:** | TC-06 | | | |
| **Description:** | Users will be able to post ride | | | |
| **Pre-condition:** | User is logged in. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the post ride button. | Enter the post | User will be able to post ride | User has posted ride successfully | Pass |
| **Post Condition** | User has posted ride successfully | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Post Ride(Negative)** | | | | |
| **Test Case scenario** TCS-02 | | | | |
| **Test Case ID:** | TC-06 | | | |
| **Description:** | Users will not be able to post ride | | | |
| **Pre-condition:** | User is logged in. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the post ride button. | Enter the post | User will be able to post ride | User has not posted ride successfully | Fail |
| **Post Condition** | User has not posted ride. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ViewHistory(Positive)** | | | | |
| **Test Case scenario** TCS-01 | | | | |
| **Test Case ID:** | TC-07 | | | |
| **Description:** | Users will be able to view history of ride | | | |
| **Pre-condition:** | User is logged in. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the view history UI |  | User will be able to view history | User has view the view history | Pass |
| **Post Condition** | User has viewed history successfully | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ViewHistory(Negative)** | | | | |
| **Test Case scenario** TCS-02 | | | | |
| **Test Case ID:** | TC-07 | | | |
| **Description:** | Users will not be able to view history of ride | | | |
| **Pre-condition:** | User is logged in. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the history UI |  | User will be able to view history | User has not view the history | Fail |
| **Post Condition** | User has not viewed history . | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ViewHistory(Positive)** | | | | |
| **Test Case scenario** TCS-03 | | | | |
| **Test Case ID:** | TC-07 | | | |
| **Description:** | Users will be able to view history of ride | | | |
| **Pre-condition:** | User is logged in. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the history UI |  | User will be able to view history | User has view the history | Pass |
| **Post Condition** | User has viewed history successfully | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ViewHistory(Negative)** | | | | | |
| **Test Case scenario** TCS-04 | | | | | |
| **Test Case ID:** |  | TC-07 | | | |
| **Description:** |  | Users will not be able to view history. | | | |
| **Pre-condition:** |  | User is logged in. | | | |
| **Test Step** |  | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the history UI |  |  | User will be able to view history | User has not view the history | Fail |
| **Post Condition** |  | User has not viewed history successfully | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Send Message(Positive)** | | | | |
| **Test Case scenario** TCS-01 | | | | |
| **Test Case ID:** | TC-08 | | | |
| **Description:** | Users will be able to send messages in the chat box. | | | |
| **Pre-condition:** | User is logged in.  There must be at least two participants in the meeting. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the chat button. | Enter the message to send. | User will be able to send the message. | User has send the message successfully | Pass |
| **Post Condition** | User has send the message successfully | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Send Message(Negative)** | | | | |
| **Test Case scenario** TCS-02 | | | | |
| **Test Case ID:** | TC-08 | | | |
| **Description:** | Users will not be able to send messages in the chat box. | | | |
| **Pre-condition:** | User is logged in.  There must be at least two participants in the meeting. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the chat button. | Enter the message to send. | Users will be able to send the message. | User has not send the message successfully | Fail |
| **Post Condition** | User has not send the message successfully | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Send Message(Positive)** | | | | |
| **Test Case scenario** TCS-03 | | | | |
| **Test Case ID:** | TC-08 | | | |
| **Description:** | Users will not be able to send spaces in the chat box. | | | |
| **Pre-condition:** | User is logged in.  There must be at least two participants in the meeting. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the chat button. | Enter the space button multiple times to send. | Users will be able to send the message. | User has not send the message successfully | Fail |
| **Post Condition** | User has not send the message successfully | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Accept user(Positive)** | | | | |
| **Test Case scenario** TCS-01 | | | | |
| **Test Case ID:** | TC-09 | | | |
| **Description:** | Admin will be able to accept new user’s profile. | | | |
| **Pre-condition:** | User is logged in as a Host. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the approve button. | Enter the users of the participants. | Users will be able to accept new user’s profiles. | User’s profile has been approved successfully | Pass |
| **Post Condition** | Host has added the user successfully | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Accept user(Negative)** | | | | |
| **Test Case scenario** TCS-02 | | | | |
| **Test Case ID:** | TC-09 | | | |
| **Description:** | Users will not be able to accept new users. | | | |
| **Pre-condition:** | User is logged in as an admin. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the approve button. | Enter the users of the participants. | Users will not be able to accept new users. | User has not been added successfully | Fail |
| **Post Condition** | admin has not added the new user successfully | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Block user(Positive)** | | | | |
| **Test Case scenario** TCS-01 | | | | |
| **Test Case ID:** | TC-10 | | | |
| **Description:** | Admin will be able to block the user's profile. | | | |
| **Pre-condition:** | User is logged in as a Host. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the block profile button. | Enter the user id of the participants. | Admin will be able to block the user. | User has blocked successfully | Pass |
| **Post Condition** | Host has blocked the user. | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Block user(Negative)** | | | | |
| **Test Case scenario** TCS-02 | | | | |
| **Test Case ID:** | TC-10 | | | |
| **Description:** | Admin will not be able to block the user. | | | |
| **Pre-condition:** | User is logged in as an admin. | | | |
| **Test Step** | **Test data** | **Expected**  **Result** | **Actual**  **Result** | **Stat Remarks** |
| 1. Click on the block button. | Enter the user id of the participants. | Admin will be able to block the user. | User has not been blocked. | Fail |
| **Post Condition** | User has not blocked successfully | | | |