# IT632: SOFTWARE ENGINEERING

## GROUP ID: 7

# PROJECT: A REAL-TIME MOBILE APP FOR SHARING CARS

**GUIDE: SAURABH TIWARI**
**MENTOR: PRIYANKA MISHRA**

### GROUP MEMBERS' DETAILS

| Sr. No. | Member Name | Member ID |
|---|---|---|
| 1 | Devansh Dalmia | 202112002 |
| 2 | Shah Jainam Rajeshbhai | 202112016 |
| 3 | Abhishek Aneja | 202112020 |
| 4 | Solanki Yash Jitendrasinh | 202112028 |
| 5 | Trivedi Kartikkumar Pratikbhai | 202112056 |
| 6 | Dharmil Digantbhai Shah | 202112109 |
| 7 | Desai Darshi Rakeshkumar | 202112113 |
| 8 | Khushi Shah | 202112123 |
| 9 | Shah Nisha Vipulbhai | 202112125 |
| 10 | Acharya Shreya Hiteshkumar | 202112127 |

# INDEX

## (1) Introduction

A real time app for sharing cars is an android application that is made for sharing cars among users. Through the application, any user can register by adding the necessary details and then can easily update that (s)he is available to share the ride. Then on the other side, the user who wants to take a ride can view the users available to share the ride and can request any one of them. This way the users can easily share the ride with another user.

## (2) Overall Description

### (2.1) Scope Of The Project

Carpooling is one of the latest technologies, making travelling convenient and efficient for the commoner. It is also known as car-sharing, in which one can travel to their destination while sharing a vehicle and the expenses incurred. Hence fuel costs, tolls and the stress of driving will be reduced when more people travel together in one vehicle. It also helps to reduce traffic congestion and other poisonous gases in the air. It can help to save a lot of space in the parking lot.

Using the carpooling system is an intelligent decision during high fuel prices and high pollution periods. In our application, we will make an Android-based application that will allow passengers to collaborate with other like-minded people and plan out their journey using the easy UI of the app after signing in to it. Pre-registration ensures security, as only identified people get into the vehicle to establish trust. People will also be able to share expenses and not have to worry about reaching late while making new connections.

## (2.2) Users And Stakeholders

**Users:**

- Carpoolers: Users who are offering rides to start pooling practice.
- Riders: Users who are taking a ride that is offered by the user.
- Admin: Accepts Or Rejects the request sent for adding a car to validate the car details and avoid malfunctioning in the application.

**Stakeholders:**

The term “stakeholder" refers to the people or groups affected by a software development project. Stakeholders exist for both within the organisation and outside of it.

- Government: Governments can also be considered a major stakeholder in a business, as they collect taxes from the company (corporate income taxes), as well as from all the people it employs (payroll taxes) and from other spending the company incurs (sales taxes).
- NGOs: NGOs are generally regarded as important stakeholders, representatives of key stakeholder groups and most trusted institution
- Private Institutions: Independent entity owned by a non-State entity, such as a firm, business enterprise or individual.
- Development Team: A development team is a group of people that work together to create software.
- Project Managers: The project manager is the individual responsible for delivering the project. The individual leads and manages the project team, with authority and responsibility from the project board, to run the project on a day-to-day basis.
- Carpoolers (users offering ride): an arrangement in which a group of people commute together by car.
- Riders (users sharing ride): A rider is someone who rides with the one offering a ride.
- Admin: The one who maintains the application and validates car details.

### (2.3) Possible Features

1. Logging in
2. Personal Profile
3. Add vehicle details
4. Verifying Vehicle details
5. Add Licence details
6. Verifying details
7. Temperature and vaccination status
8. Offer Ride
9. Take ride
10. View active rides
11. Accepting or Reject requests
12. Cancel request ( user requesting can also cancel request until accepted)
13. Cancel ride
14. Modify ride
15. Real-time tracking for mobility updates
16. Getting the quickest and shortest route
17. Navigation
18. Pricing estimation
19. Messaging
20. Notifications
21. Billing section
22. Split-the-bill option
23. Multiple payment channel integrations
24. Feedback section
25. Report
26. Support feature

## (2.4) Requirement Elicitation Technique

### Use Case Approach

Use case approach is being used as a requirement elicitation technique. The use cases describe the ‘what’ of a system and not ‘how’. Hence, they only give a functional view of the system.

The components of the use case design include three crucial things:

**1. Actor:**

It is the external agent that lies outside the system but interacts with it somehow. An actor, maybe a person, a machine etc.

In our project actors are :

1. User offering ride
2. User taking ride
3. Admin

**2. Use Cases:**

They describe the sequence of interactions between actors and the system. They capture who(actors) do what(interaction) with the system.

3. **Use Case Diagram:**

A use case diagram graphically represents what happens when an actor interacts with a system.

## (2.5) Process Model

### Scrum

It is a type of Agile framework. It is a framework within which people can address complex adaptive problems while productivity and creativity of delivering product is at highest possible values. Scrum uses an Iterative process.

This process includes a Scrum Master who sets up the scrum team, arranges the meeting and ensures the process happens smoothly. The scrum team manages its work and organises it to complete the sprint. It is a combination of incremental and iterative methodology. The process involves breaking down each project into prioritised requirements, and delivering each individually within an iterative cycle. An iteration is the routine of developing small sections of a project at a time. Each iteration is reviewed and assessed by the development team and client.

### Reasons for choosing Scrum

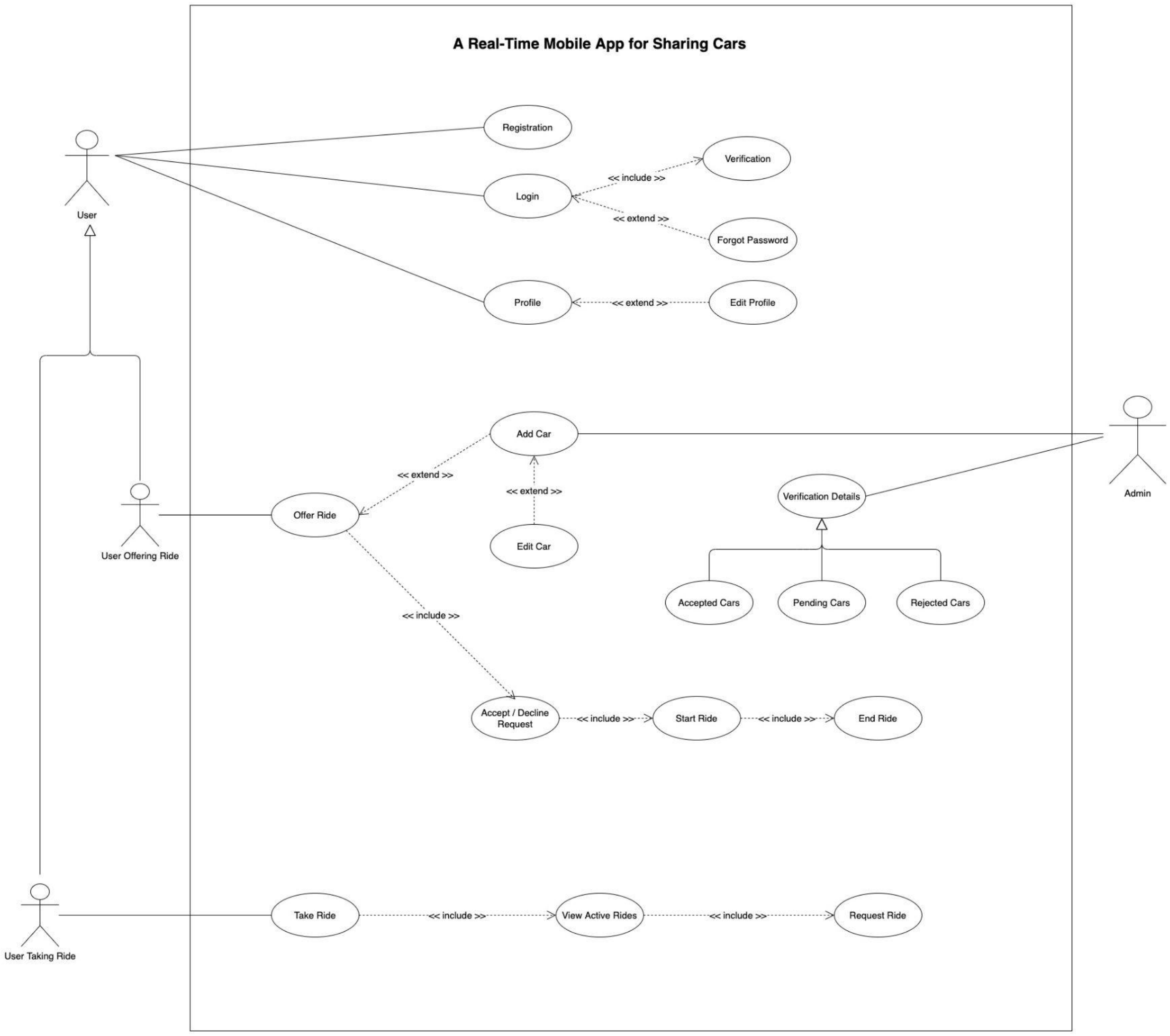
1. In this model, the development team does not attempt to develop all features at once. Instead, we divide the car-pooling project into multiple iterations to focus on the quality.
2. Every iteration includes cross functional teams working on different areas such as planning, requirements, analysis, coding and testing.
3. After every iteration, the project will be reviewed by different users of the application.
4. Adaptability is one of the key reasons as in this model teams can easily make changes in the plan.
5. The iterations in it are shorter in duration ranging from 2 weeks to 2 months.
6. It reduces the risk as the application will be developed in a sprint allowing us to find out whether we are on track or not.

# (3) Functional Requirements

## (3.1) User Stories

- As a user, I want to register myself, so that I can log in to the application.
- As a user, I want to log in, so that I can use the functionalities of the application.
- As a user, I want the application to verify users' login credentials, so that no user can access the application without registration.
- As a user, I want to reset the password, so that I can log in again in case I forgot the password.
- As a user, I want to take a tour of the application, so that I don't face any difficulty while using it.
- As a user, I want to edit my profile details, so that I can update the details whenever required.
- As a user offering a ride, I want to add the car, so that I can offer the ride.
- As a user offering a ride, I want to edit the car details, so that I can update the information whenever required.
- As an admin, I want to accept or reject the cars that the user adds so that I can authenticate that the information is genuine.
- As an admin, I want to view all accepted cars, so that I can view that information in future.
- As an admin, I want to view all rejected cars, so that I can view that information in future.
- As a user offering a ride, I want to offer a ride by filling in the necessary details, so that anybody who wants to share the ride can request me.
- As a user offering a ride, I want to accept or reject users' taking a ride, so that I can choose according to my choice.
- As a user offering a ride, I want to start the ride, so that I can keep the record of rides (when it started).
- As a user offering a ride, I want to end the ride, so that I can keep the record of rides (when it ended).
- As a user taking a ride, I want to view the offered rides, so that I can select the ride according to my requirements.
- As a user taking a ride, I want to request a ride to a user available, so that I can take a ride with that user.

## (3.2) USE CASE DIAGRAM



### (3.3) Use Case Description

| Sr. No | Use Case | Description |
|---|---|---|
| 1 | Registration | Users have to register by providing necessary details. |
| 2 | Login | After registration, users need to log in to access the functionalities of the application. |
| 3 | Verification | The application will verify the details of the user with the database before letting him / her login. |
| 4 | Profile | The user will be able to access his profile details whenever required. |
| 5 | Offer Ride | The user will select this feature to offer the ride to another user. |
| 6 | Add Car | The user offering a ride will be required to add the car and licence details which further will be verified by the admin.. |
| 7 | Edit Car | The user will be able to edit the car details whenever required. |
| 8 | Accept / Decline Request | The user can accept or decline the request of the users who are requesting to share the ride. |
| 9 | Take Ride | In order to take a ride, the user needs to click on the take a ride button and provide the necessary details. |
| 10 | View Rides | The user will be able to view the ride(s) available according to the details filled by him/her. |
| 11 | Request Ride | In order to take a ride, the user needs to request another available user. |
| 12 | Verification Details | The admin will be able to view the verification details log where he can see all accepted, pending and rejected car and licence details of the users. |

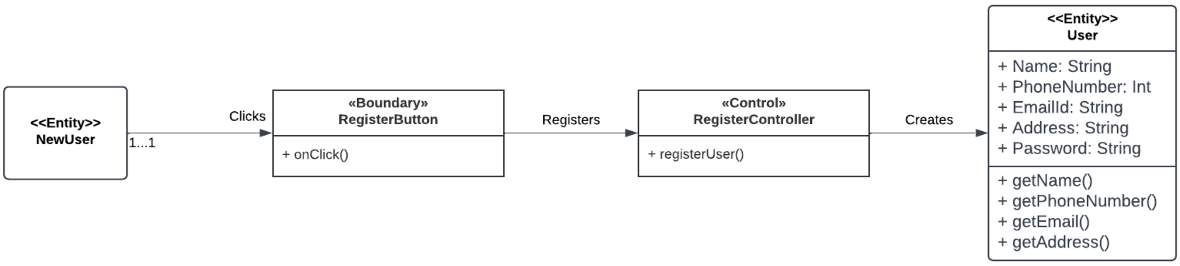
### (4) Non-Functional Requirements

- As a user, I want to be able to run the application on all Android versions so that I can use it on any android device.
- As a user, I want the application to be available 99.99% of the time so that I can access it whenever required.
- As a user, I want the application to respond quickly so that I can have a pleasant and seamless experience.
- As a user, I also want to use the application on iOS devices so that I can use the application irrespective of the type of mobile.
- As a user, I want the application not to share any of my personal details with any third party without my permission so that I can trust and use the application without any fear.
- As an application user, I want the number of users to be scalable in the database so that the application works smoothly.
- As a user, I want an easy user interface, so that it is easy to navigate the application.

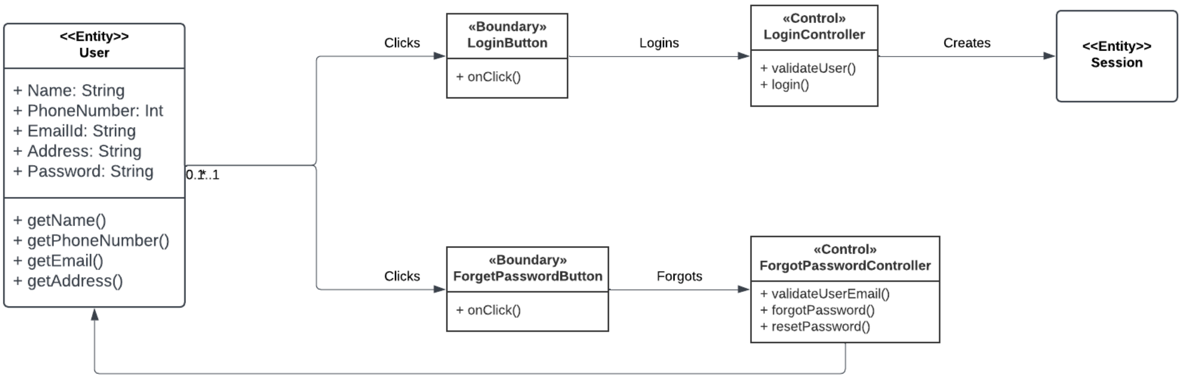
# (5) Analysis Design Documents

## (5.1) Analysis Class Diagram

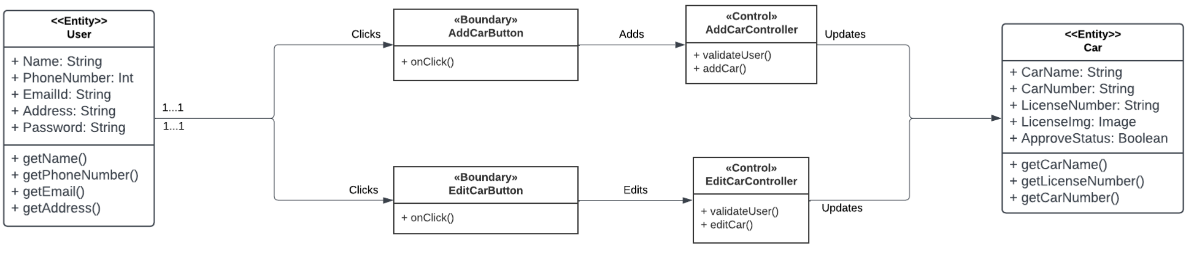
- ### REGISTRATION CLASS DIAGRAM



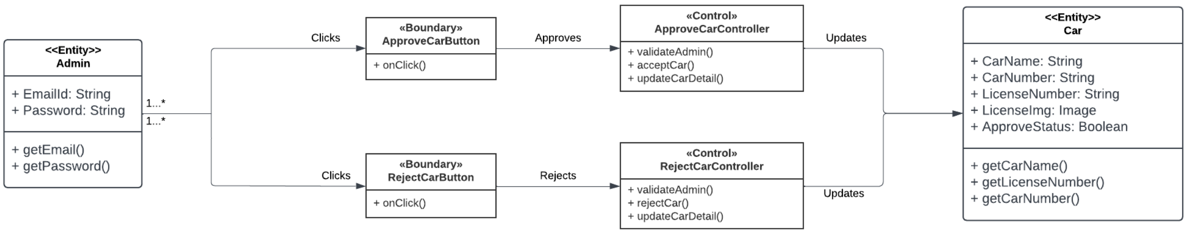
- ### LOGIN FORGOT PASSWORD



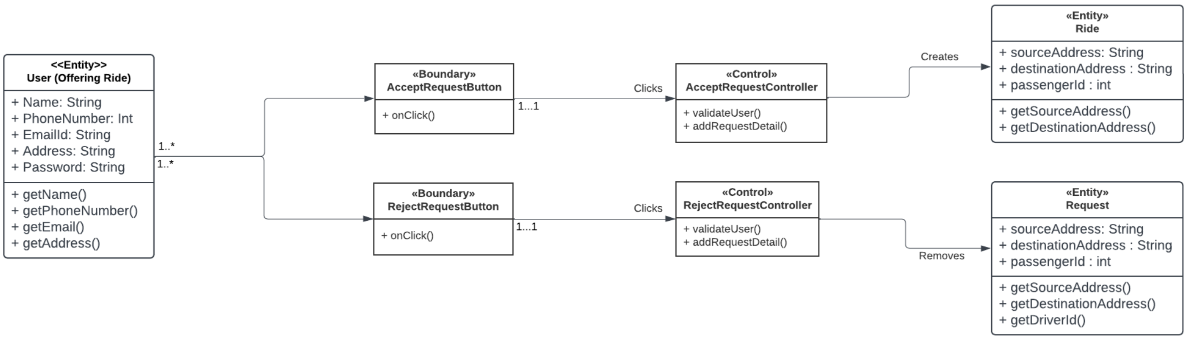
- ### ADD CAR CLASS DIAGRAM



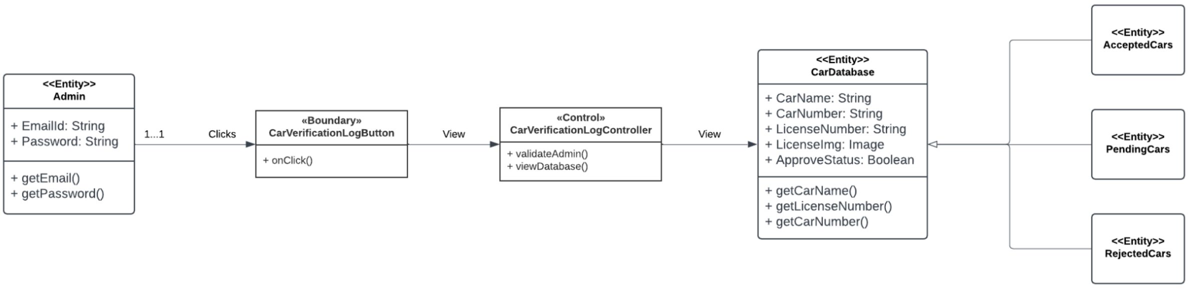
- ### ADMIN VERIFYING CAR DETAILS CLASS DIAGRAM



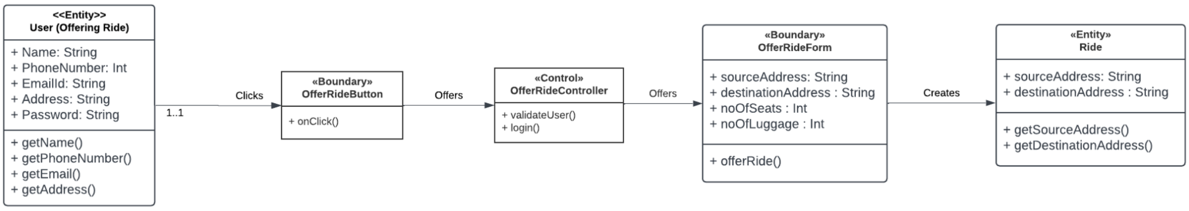
- **RESPOND TO RIDE REQUEST CLASS DIAGRAM**



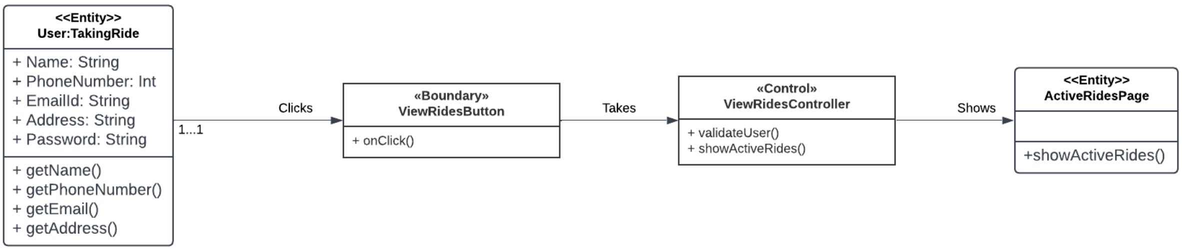
- **CAR VERIFICATION LOG CLASS DIAGRAM**



- **OFFER RIDE CLASS DIAGRAM**

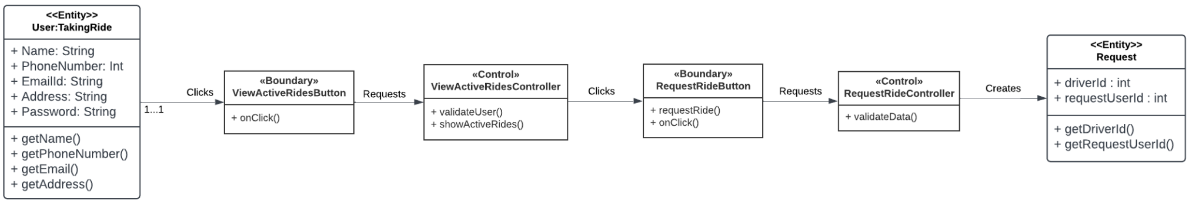


- **VIEWING ACTIVE RIDE**

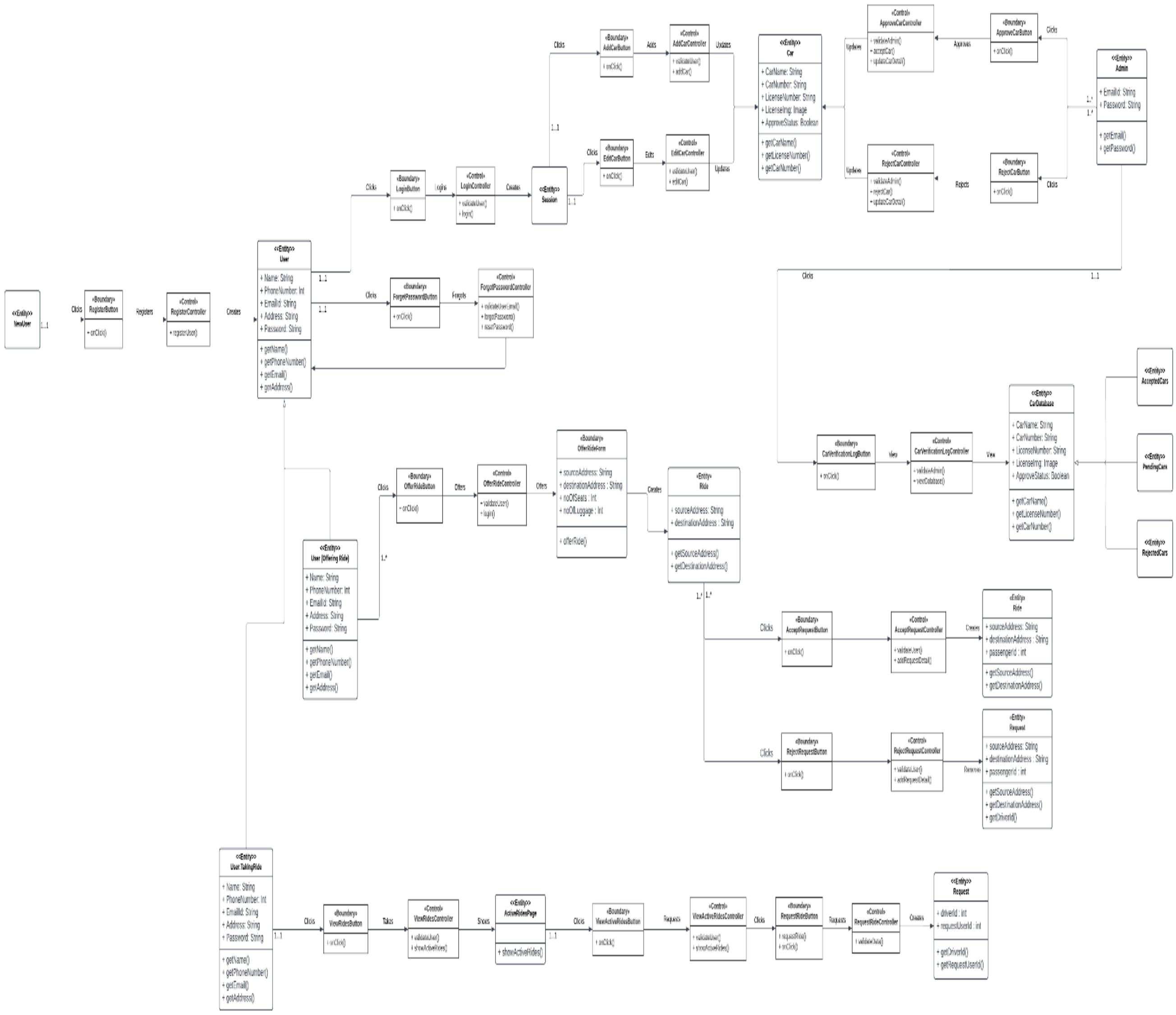


-

- **REQUESTING RIDE**

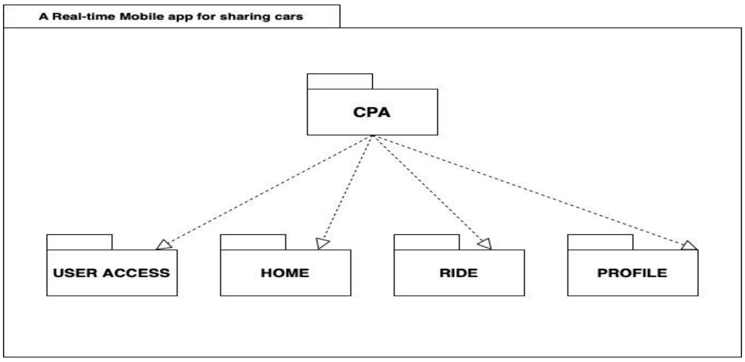


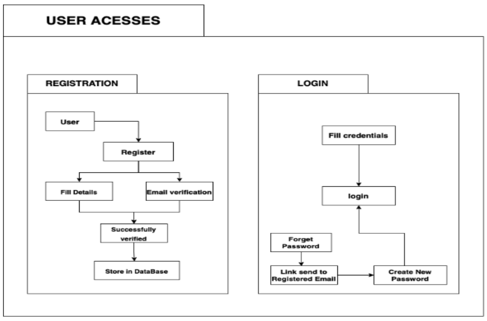
## (5.2) Complete Analysis Class Diagram

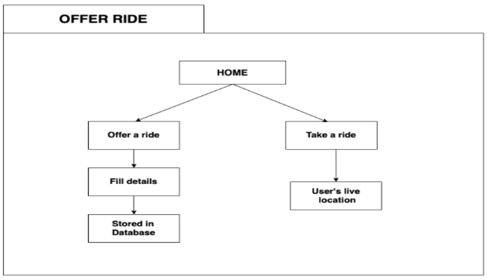


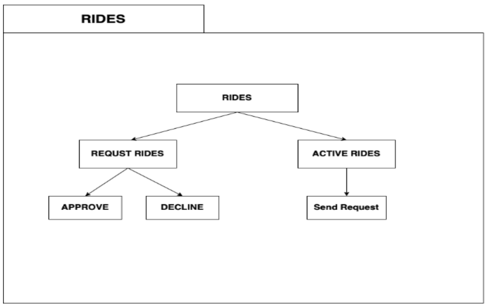
# (6) System Design

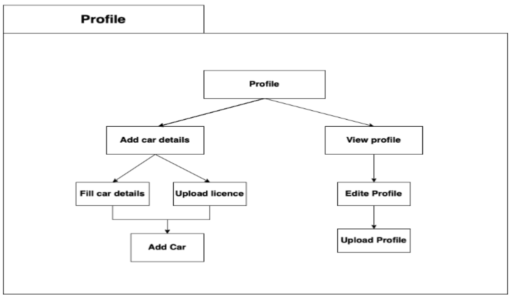
## (6.1) Sub System Design



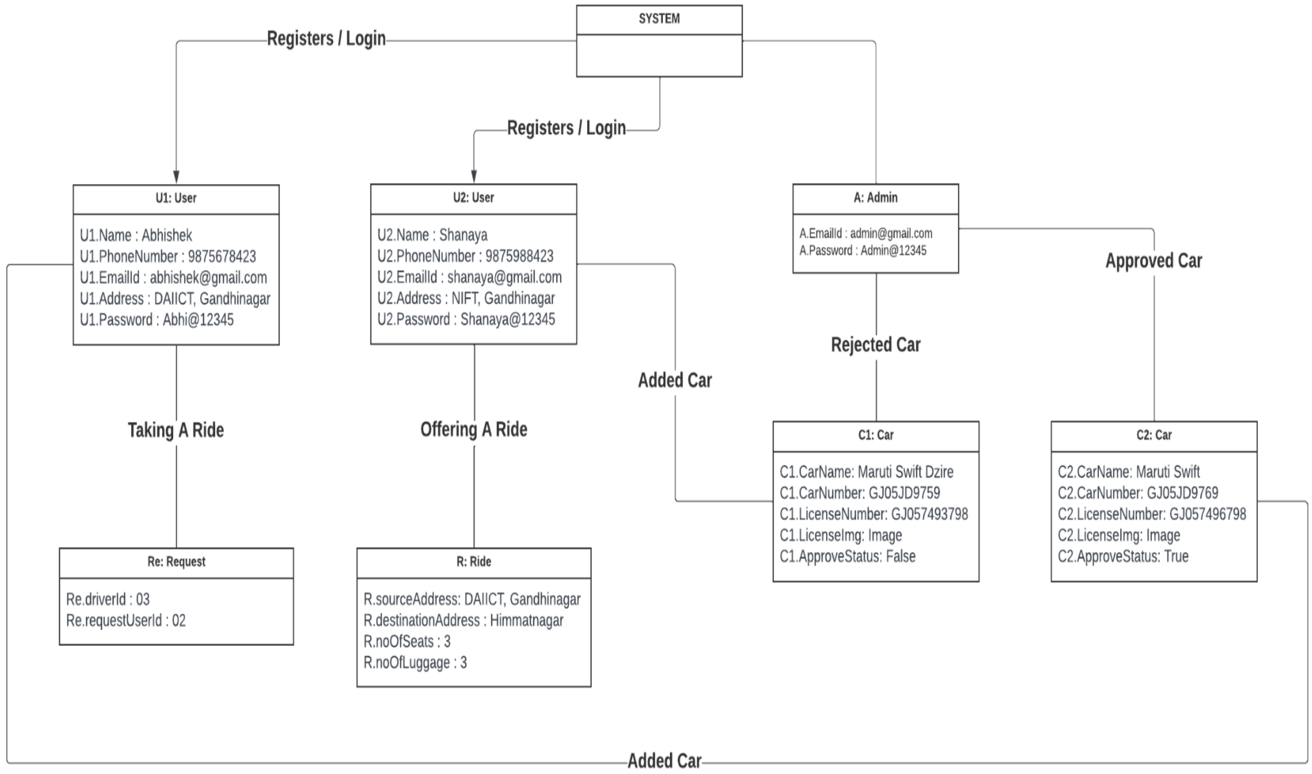








## (6.2) Object Design



## (7) Testing plan

There are two ways to test an Android application :

**Manual Testing**

Manual testing is performed manually by user or beta testers. Thus, it is very time consuming. 20% testing of applications is done using this technique.

We have implemented testing with the help of testing techniques like documentation testing, functional testing, usability testing, compatibility testing, ui testing and many more.
Testers get navigational charts, screen layouts, and other requirements invisible on the design. These requirements are analysed for completeness and inconsistency. Contradictions in the requirements must be resolved before the start of development.

Artefacts like Requirements (Specification, PRD), Test Plan, Test Cases, Traceability Matrix are created and analysed on this stage.The installation of the application should take place without significant errors, if the device meets the system requirements.Verify the application automatically starts correctly.Ensure the user manual is available.Ensure the application's operation during startup/exit meets the basic requirements.

**Automated Testing**

Automated Testing saves time and increases efficiency. Used for 80% test cases. This is mainly used when manual testing gets tedious.

In automated testing we have used the **appium** tool and adb installer to test the application. Different test cases are being generated and tested using emulators and some other physical devices.

## (8) Testing Strategies and Framework

### Installing and running the application

The installation of the application should take place without significant errors, if the device meets the system requirements.
Verify the application automatically starts correctly.
Ensure the user manual is available.
Ensure the application's operation during startup/exit meets the basic requirements.

### Fields testing

Verify the required fields work correctly.
Make sure that mandatory and optional fields are displayed in different ways.

### Interruptions testing

Incoming and outgoing calls, SMS, and MMS.
Battery discharge/removal.
Disconnecting and connecting the network/Wi-Fi.
Disconnecting and connecting the SD-card.
Charging the device.

### Constant users feedback testing

Downloading content messages
Progress bar.
The appropriate reaction of the buttons on pressing.
Network access error messages.
Attempt to delete important information messages.
Availability and synchronisation of sound, vibration, and visual notifications.
The appearance of a screen (message) at the end of the process ( game).

### Update testing

All user data is saved after updates.
Ensure the update progress is displayed properly.
Make sure updates are supported by older operating systems.
Testing  various ways of installing updates (Wi-Fi, Bluetooth, USB)

### Device resources testing

Lack of space to install or run the application.
Memory leaks. Pay attention to windows, with a lot of information, and tasks with long workflow.
Installing/replacing the app on the SD-card.
The absence of some functions supported by the application (3G, SD-card, etc.).
Ensure the installed application does not interfere with the normal operation of other apps and does not consume their memory.

**Some other verifications:**

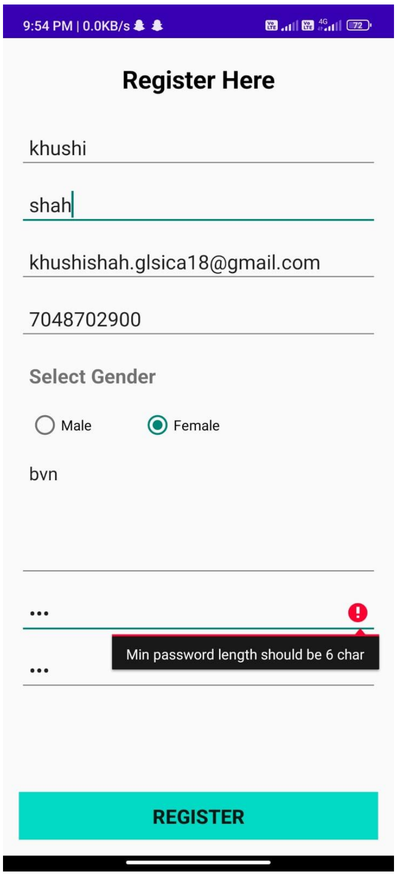
Make sure the information error messages are correct on time and appropriate.
Verify connection to analytical tools like Google Analytics.
Testing the power consumption.
Verify the necessary options to correct work with social networks – Share, Publish, Navigation.

## Black Box Testing

## Registration Flow -

| SR | Class | Valid/Invalid |
|---|---|---|
| C1 | password.length > 6 | Invalid |
| C2 | password.length = 6 | Valid |
| C3 | password.length < 6 | Invalid |

| Test Action and Input Data | Expected Outcome | Class |
|---|---|---|
| Equivalence Class | | |
| password = tus736 | Successfully Registered | C2 |
| password = bsg2 | Register Unsuccessful | C1 |
| password = bsk28 | Register Unsuccessful | C3 |
| Boundary Value Analysis | | |
| password = jei83 | Register Successful | C3 |
| password = sjd2871 | Register Successful | C1 |



Register Here

khushi

shah

khushishah.glsica18@gmail.com

7048702900

Select Gender

○ Male ● Female

bvn

•••

Min password length should be 6 char

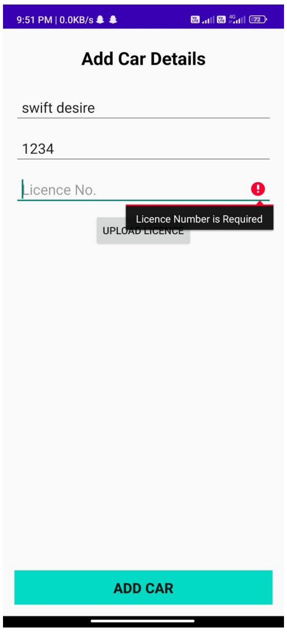
•••

## Add Car Flow -

| SR | Class | Valid/Invalid |
|---|---|---|
| C1 | LicenseNumber.length > 13 | Invalid |
| C2 | LicenseNumber.length = 13 | Valid |
| C3 | LicenseNumber.length < 13 | Invalid |

| Test Action and Input Data | Expected Outcome | Class |
|---|---|---|
| Equivalence Class | | |
| LicenseNumber = GJ1234567890123 | Successfully Registered | C2 |
| LicenseNumber = GJ01234567 | Register Unsuccessful | C1 |
| LicenseNumber = GJ01234 | Register Unsuccessful | C3 |
| Boundary Value Analysis | | |
| LicenseNumber = GJ0123456123456 | Register Unsuccessful | C3 |

| | | |
|---|---|---|
| LicenseNumber = GJ012345678123 | Register Unsuccessful | C1 |



Add Car Details

swift desire

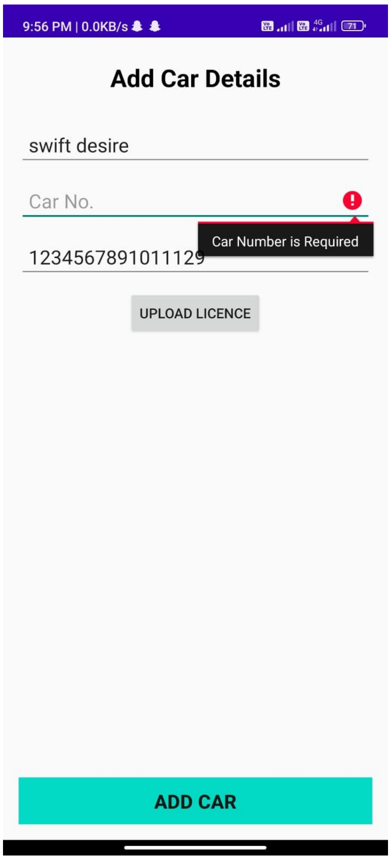
1234

Licence No.

Licence Number is Required

| SR | Class | Valid/Invalid |
|---|---|---|
| C1 | CarNumber.length > 10 | Invalid |
| C2 | CarNumber.length = 10 | Valid |
| C3 | CarNumber.length < 10 | Invalid |

| Test Action and Input Data | Expected Outcome | Class |
|---|---|---|
| Equivalence Class | | |
| CarNumber = GJ51AJ2231 | Successfully Registered | C2 |
| CarNumber = GJ51AJ22 | Register Unsuccessful | C1 |
| CarNumber = GJ51A | Register Unsuccessful | C3 |
| Boundary Value Analysis | | |
| CarNumber = GJ51AJ2239 | Register Unsuccessful | C3 |
| CarNumber = GJ51a5534 | Register Unsuccessful | C1 |



Add Car Details

swift desire

Car No.

Car Number is Required

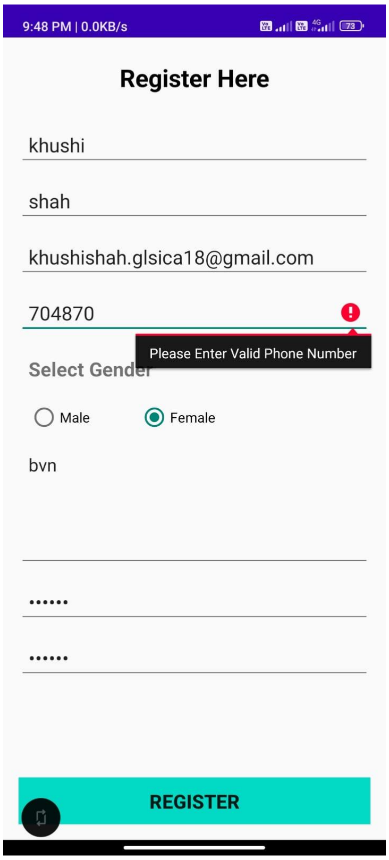
1234567891011129

## Add User -

| SR | Class | Valid/Invalid |
|---|---|---|
| C1 | phoneNumber.length > 10 | Invalid |
| C2 | phoneNumber.length = 10 | Valid |
| C3 | phoneNumber.length < 10 | Invalid |

| Test Action and Input Data | Expected Outcome | Class |
|---|---|---|
| Equivalence Class | | |
| phoneNumber = 9875678989 | Successfully Registered | C2 |
| phoneNumber = 98756789 | Register Unsuccessful | C1 |
| phoneNumber = 987567 | Register Unsuccessful | C3 |
| Boundary Value Analysis | | |
| phoneNumber = 98756789 | Register Unsuccessful | C3 |

| | | |
|---|---|---|
| phoneNumber = 98756 | Register Unsuccessful | C1 |



9:48 PM | 0.0KB/s

Register Here

khushi

shah

khushishah.glsica18@gmail.com

704870

Please Enter Valid Phone Number

Select Gender

○ Male ● Female

bvn

••••••

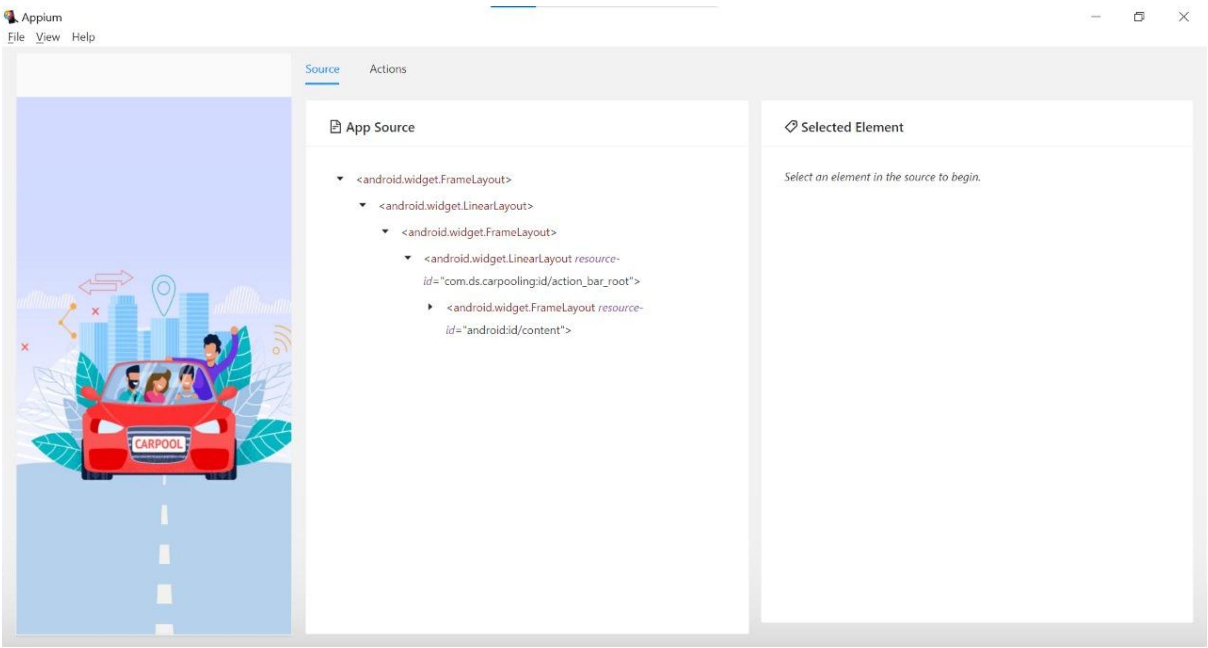
••••••

**General Manual Testing**

| Test Id | Test case name | Test case Description | Input Description | Expected output | Observed | Status |
|---|---|---|---|---|---|---|
| 1 | Login | Username/ password is wrong. | Username/ password is wrong. | Enter correct username and password | Enter correct username and password | Fail |
| 2 | Login | Username/ Password is right | Entering registered username and password | Successfully logged in | Successfully logged in | Success |
| 3 | Registration | Registering with non-valid ID | abcgmail.com | Please Provide Email like abc@gmail.com | Please Provide valid Email | Fail |
| 4 | Offering ride | Offering ride without entering destination address | Enter Source address | Destination Address is also Required | Destination Address is also Required | Fail |
| 5 | Offering ride | Offer Ride without entering source destination | Enter Destination address | Source destination address is also required | Source destination address is also required | Fail |
| 6 | Taking ride | Take Ride without allowing GPS | Gps not enabled | Enable gps for user's current location | Enable gps for user's current location | Fail |
| 7 | Offering Ride | Offering ride with entering same source and destination address | Exactly same source and destination address | Enter different source and destination address | Not able to offer ride as address of source and destination is same | Fail |
| 8 | Profile | Add car detail | Add car name =Verna,<br>Car no=Gj095141,<br>Licence no=12C036KK532, | Car details Successfully Updated. | You have to fill all details,like Car model name,Car No,licence No & Upload licence also | Success |

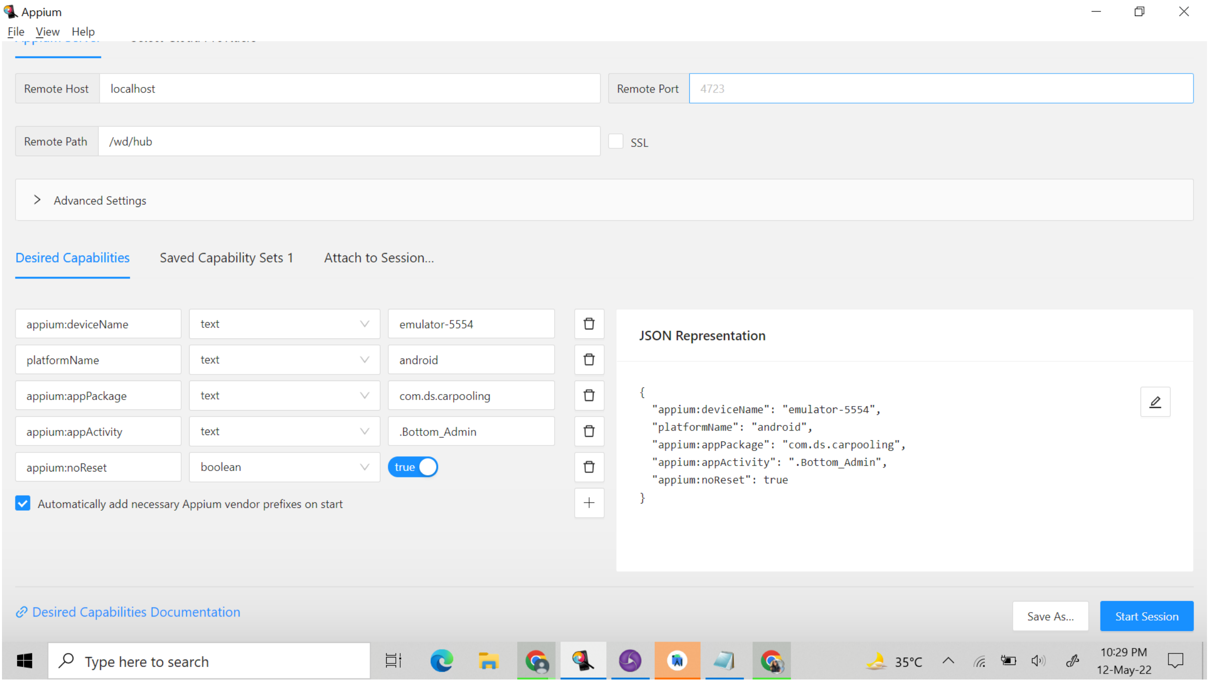
| | | | | | | |
|---|---|---|---|---|---|---|
| | | | upload licence= image.png | | | |
| 9 | Profile | Add car detail | After clicking on add a car again it should show previous car detail which was add before | Car detail Not showing previous added cars | Not showing the previous data | Fail |
| 9 | Profile | View Profile/Details | After clicking on view profile the detail which was entered at the time of registration | Successfully showing all the details in view profile | You can see the detail which was already registered | Success |
| 10 | Profile Update details | Detail updation in update profile | By editing the details on the already provided details it should be updated | Successfully updated | You have to modify previous detail | Success |
| 11 | Rides | Offer a Rides | Destination and other required details input | After clicking on offer a ride compulsory user need to fill all the required details in order to request | Successfully giving error message | Success |
| 12 | Rides | Active a ride | Asking for request by clicking on Request button | After clicking Request button it should show the the detail on request ride | Successfully showing the output | Success |

| | | | | | | |
|---|---|---|---|---|---|---|
| 13 | Rides | Request a ride(Approve ) | Asking for request Approve | After clicking on accept button request should be approved | Successfully Showing the output | Success |
| 14 | Rides | Request a ride(Decline) | Asking for Decline | After clicking on Decline button request should be Decline | Successfully showing the output | Success |



```
<android.widget.FrameLayout>
  <android.widget.LinearLayout>
    <android.widget.FrameLayout>
      <android.widget.LinearLayout resource-
      id="com.ds.carpooling:id/action_bar_root">
        <android.widget.FrameLayout resource-
        id="android:id/content">
```

*Select an element in the source to begin.*



Remote Host: localhost

Remote Port: 4723

Remote Path: /wd/hub

□ SSL

> Advanced Settings

| | | |
|---|---|---|
| appium:deviceName | text | emulator-5554 |
| platformName | text | android |
| appium:appPackage | text | com.ds.carpooling |
| appium:appActivity | text | .Bottom_Admin |
| appium:noReset | boolean | true |

☑ Automatically add necessary Appium vendor prefixes on start

JSON Representation

```
{
  "appium:deviceName": "emulator-5554",
  "platformName": "android",
  "appium:appPackage": "com.ds.carpooling",
  "appium:appActivity": ".Bottom_Admin",
  "appium:noReset": true
}
```

## Session Capablities:

```
      {
  "platform": "LINUX",
  "webStorageEnabled": false,
  "takesScreenshot": true,
  "javascriptEnabled": true,
  "databaseEnabled": false,
  "networkConnectionEnabled": true,
  "locationContextEnabled": false,
  "warnings": {},
  "desired": {
    "platformName": "android",
    "app": "C:\\Users\\darsh\\Downloads\\app-debug.apk",
    "deviceName": "emulator-5554",
    "ensureWebviewsHavePages": true,
    "nativeWebScreenshot": true,
    "newCommandTimeout": 3600,
    "connectHardwareKeyboard": true
  },
  "platformName": "android",
  "app": "C:\\Users\\darsh\\Downloads\\app-debug.apk",
  "deviceName": "emulator-5554",
  "ensureWebviewsHavePages": true,
  "nativeWebScreenshot": true,
  "newCommandTimeout": 3600,
  "connectHardwareKeyboard": true,
  "deviceUDID": "emulator-5554",
  "appPackage": "com.ds.carpooling",
  "deviceApiLevel": 30,
  "platformVersion": "11",
  "deviceScreenSize": "1080x2280",
  "deviceScreenDensity": 440,
  "deviceModel": "sdk_gphone_x86",
  "deviceManufacturer": "Google",
  "pixelRatio": 2.75,
  "statBarHeight": 135,
  "viewportRect": {
    "left": 0,
    "top": 135,
    "width": 1080,
    "height": 1842
  },
  "lastScrollData": null
}
```

## (9) Challenges We Faced

### (9.1) Open Issues

We faced many issues while working on this android application. While developing, there were some issues like fragment layout, integrating maps, routing different routes and many more. Challenges were easily solved as it was a team of ten members.

We can also add on the feature of payment after the completion of ride, splitting the bill and also cancelling the ride if it is accepted by 2 users offering the ride. We can improve the verification process in our application.

## (10) Lessons Learnt

By using software engineering concepts at various stages of software development, we were able to gain a better understanding of them.

We also learned about other software engineering tools, such as appium, adb.

We learned how to organise and coordinate among ourselves both offline and online because our group was just ten students.

It is not only important to develop the architecture beforehand, but it is also significant to think of how you will handle bugs and other problems. For all systems that are bigger than a few lines of code, it is necessary to develop a roadmap so you will know when and where to look for the problem and what exactly happens inside the program.
Most of the development tasks require changing existing lines of code in one way or another. Even new features need to fit into the original functionality. First, developers need to examine and understand the pros and cons of the current solution and only then build something new.

**CONTRIBUTION**

| ID | NAME | WORK |
|---|---|---|
| 202112002 | Devansh Dalmia | Documentation, Routing, Registration / Login Module, Presentation, Splash Screen |
| 202112016 | Jainam Shah | Use Case Diagram, Routing, View Profile, Edit Profile, Presentation, Designing, Active Ride Fragment |
| 202112020 | Abhishek Aneja | Documentation, Recycler View, Designing, Add car, Backend Integration, Testing |
| 202112028 | Yash Solanki | Sub System Diagram, Testing, Add Profile Module, Usecase description, Analysis class diagram |
| 202112056 | Kartik Trivedi | Recycler View, Designing, Ride Fragment, View Profile, Presentation |
| 202112109 | Dharmil Shah | Firebase, Add a car, Request to admin, Admin side (Pending, Rejected, Approved) , Map Integration, Add a car, Rides Fragment |
| 202112113 | Darshi Desai | Admin Side Panel, Testing, Map Integration, Offer a ride |
| 202112123 | Khushi Shah | Firebase integration, Documentation, Routing, Forgot Password, Email Verification, Backend, Map Integration |
| 202112125 | Nisha Shah | Admin Side Panel, Testing, Map Integration, Take a ride |
| 202112127 | Shreya Acharya | Testing,  Analysis Class Diagram |