SHAHEED ZULFIKAR ALI BHUTTO
INSTITUTE OF SCIENCE AND TECHNOLOGY

Final Year Project Report

Go Car

Project Team:

Nabil Kadiwal 1912277
Abdul Rehman 1912255

1/08/2023

Project Supervisor:

Sir Khawaja Mohiuddin

Submitted in the partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer Science in the

Faculty of Computing and Engineering Sciences

Declaration of Authorship

We, Nabil Kadiwal (1912277) and Abdul Rehman (1912255), declare that this report titled, “Go Car” and the work presented in it are our own. We confirm that:
This work was done wholly or mainly while in candidature for a bachelor’s degree at this University.
Where any part of this report has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated. Where we have consulted the published work of others, this is always clearly attributed.

Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this report is entirely our own work. We have acknowledged all main sources of help.

Where the report is based on work done by ourselves jointly with others, we have made clear exactly what was done by others and what we have contributed ourselves.

Signed:

Date: 1st August 2023

Project Description

As we all know transport service like car and bike are common mode of transport everyone needs on daily bases but not having vehicles or due to highly expensive recourses like petrol and CNG in today's time. It has been very difficult to afford this type of service. As seen after pandemic inflation has increased drastically everywhere due to which petrol price has reached at its peak in regards to it transports service is also charging a lot i.e., Increase the fuel expenditure in every month which is much expensive due to which many of the students and employees are not able to afford this high cost. Also, there are traffic and air pollution problems which are harmful for human beings. So, to overcome these problems we have come up with an application which will be a best platform to help out users to conquer this problem. Our application would not only be helpful for user who wants the carpooling service but also resolve the other problems of our societies.

Acknowledgement

In the name of ALLAH the most beneficent and merciful who gave us the knowledge and courage to work on this research area.

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project.

We would first like to thank our supervisor Khawaja Mohiuddin of the Computer Science faculty at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology. The door to his office was always open whenever we ran into a trouble spot or had a question about our research or writing. He consistently helped, cooperated and motivated us throughout the research.

We would like to thank to our teachers who guided us in the light of their knowledge and experience. We would also like to express our gratitude to our loving parents and family members who helped and gave us encouragement. Furthermore, we would also like to acknowledge with much appreciation the crucial role of the staff of SZABIST, who gave the permission to use all required equipment and the necessary materials to complete the project.

At the end, we would like to thank Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology for providing us with such an inspiring environment. The quality education, the cooperative faculty members and the challenging environment have always motivated and boosted the confidence level of each and every student who has been a part of Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology.

Table of Contents

Proposal:

1. Introduction

This project “Go-Car” is an application, which acts like a carpooling service provider between user/customer and driver. It will provide a platform where user can use our service to book a ride/seat for carpooling for ease of transportation in today’s time. Driver will also register to give carpooling service to the user. To overcome the problem of students, bachelors, teachers and employees we have introduce our app named Go-Car.

2. Objective

To design a real time application for users, this will help them to book a ride (car/bike) for travelling. The app will provide users with different available drivers within that area so that they can avail the carpooling service and making their mean of transport easy in the ride is feasible for user. To solve the problem of heavy travelling expense, and providing best accessible service we had come up with this app **Go-Car**.

3. Problem Description

As we all know transport service like car and bike are common mode of transport everyone needs on daily bases but not having vehicles or due to highly expensive recourses like petrol and CNG in today’s time it has been very difficult to afford this type of service. As seen after pandemic inflation has increased drastically everywhere due to which petrol price has reached at its peak i.e. 230-250rs/liter in regards to it transport service is also charging a lot i.e. Increase of around 6000 to 9000rs from 8500rs to 16000 approx per month which is much expensive due to which many of the students are not able to afford this high cost. So to overcome this problem we have come up with an android application which will be a best platform to help out user to conquer this problem. Our application would not only be helpful for user who wants the carpooling service but also to the users/drivers.

4. Methodology

Our application will act as a solution to the people who are facing problems especially in terms of finances and also to the ones who want transportation service. This application is a service provider to user where a user can book a ride according to his/her requirement like (days he/she wants to travel, timings, price and vehicles) available.
So the solution we will be providing is that the user can book the car according to the seats requirement and wouldn't have to pay for all the seats which will reduce that expense of a whole ride which he/she was giving before. And secondly the user can bargain the fare with driver so that user can travel with a reasonable and fixed price.
The application deployment will be done through flutter dart.
After research we examined some of the methodologies of the system development that assisted us to select this waterfall model that we think would suit best for our project.

Waterfall Model:
It is a standard exemplary which is used in the system development life cycle to build a system with a progressive method (one by one), it is split up into different phases and every single phase consists of tasks and various objectives. In this model, if any changes occur it can't go to the previous phase to handle the changes.
Testing of the software is passed out only once the code is been fully developed. In the waterfall model, requirements must be made clear and separate before going on to the next phase.

Advantages/strengths:
It is easy to use. Requirement is clear before development begins. The quantity of resources required to implement this model are nominal. For each stage, proper documentation is followed for excellence of the development so for better quality it is more suitable.

Drawbacks:
If a user needs the requirement to be update, it will not be executed/ updated in present development process.

Why we choose Waterfall model?
Waterfall methodology is easy as the amount of resources required to implement this model are least. The milestones are easily understood through this method. This method is selected when the requirements are understandable. Each stage of the Waterfall model is finished in a given period of time, before going to a new stage and is completed one at a time.

5. Project Scope

Our project aim is to build an efficient application to give people interactive interface platform where they can avail carpooling service and to solve their expense problem through our service and feature. This project will be built on Flutter-Dart. We are planning to use following libraries and frameworks:

- Flutter for deployment of mobile applications.
- Firebase for backend work

6. Feasibility Study

1. **Risks Involved:** The risk involved is to integrate the Map, AI into the flutter application because any several occasions like Maps working properly, viewing recommended top rider and finding a driver may have error involved because AI doesn't seem to be perfect all areas. So, we make sure to use the application carefully and update our rating and feedback on time to provide best drivers and service to the user.

1. Resource Requirement:

Brand: Laptop/Android Mobile, Processor:
Core i5 – 7th Generation,
RAM: 8 GB, HDD: 1 Terabyte HDD
Operating System: Windows 7 or above

7. Solution Application Areas

Automated system, colleges and universities, Auto and vehicles, travelling.

8. Tools/Technology

Visual studio, Flutter, Dart, Firebase, Google Map API

9. Expertise of the Team Members

The interest of each group member is in the same direction and right up there with enough knowledge regarding this project. Although one group member has more interest and knowledge towards travelling through carpooling services which would be helpful for research purposes and feature designing. Also, he is good at in frontend development. On the other hand, the one has knowledge with backend

development. The project allows us to gain ample knowledge of the most challenging part of technology.

10. Milestones

1. Login/Signup/Forgot Password/OTP
2. Driver registration
3. Map/ location
4. Find pool / offer pool
5. Crud operation
6. Fare calculation (distance matrix).
7. Chat support
8. Push notification
9. Call option
10. Reviews & Rating to driver

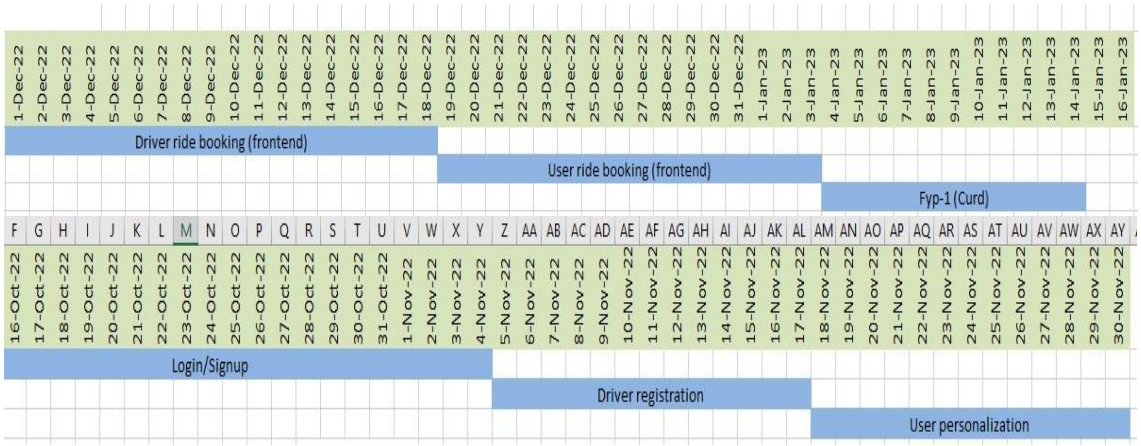
11. Project Schedule

PROJECT NAME	PROJECT DURATION (DAYS)	PROJECT START DATE	PROJECT END DATE
Go-Car	94	13-Oct-2022	14-Jan-2023

FYP – 1

Task ID	Task Description	Task Duration	Start Date	End Date
1	Login/Signup/Forgot/Password/OTP	24	13-Oct	04-Nov
2	Driver registration portal	13	05-Nov	17-Nov

3	User personalization	13	18-Nov	30-Nov
4	Driver Ride Booking (frontend)	18	01-Dec	18-Dec
5	User Ride Booking (frontend)	16	19-Dec	03-Jan
6	Add/update/delete/view	11	04-Jan	14-Jan

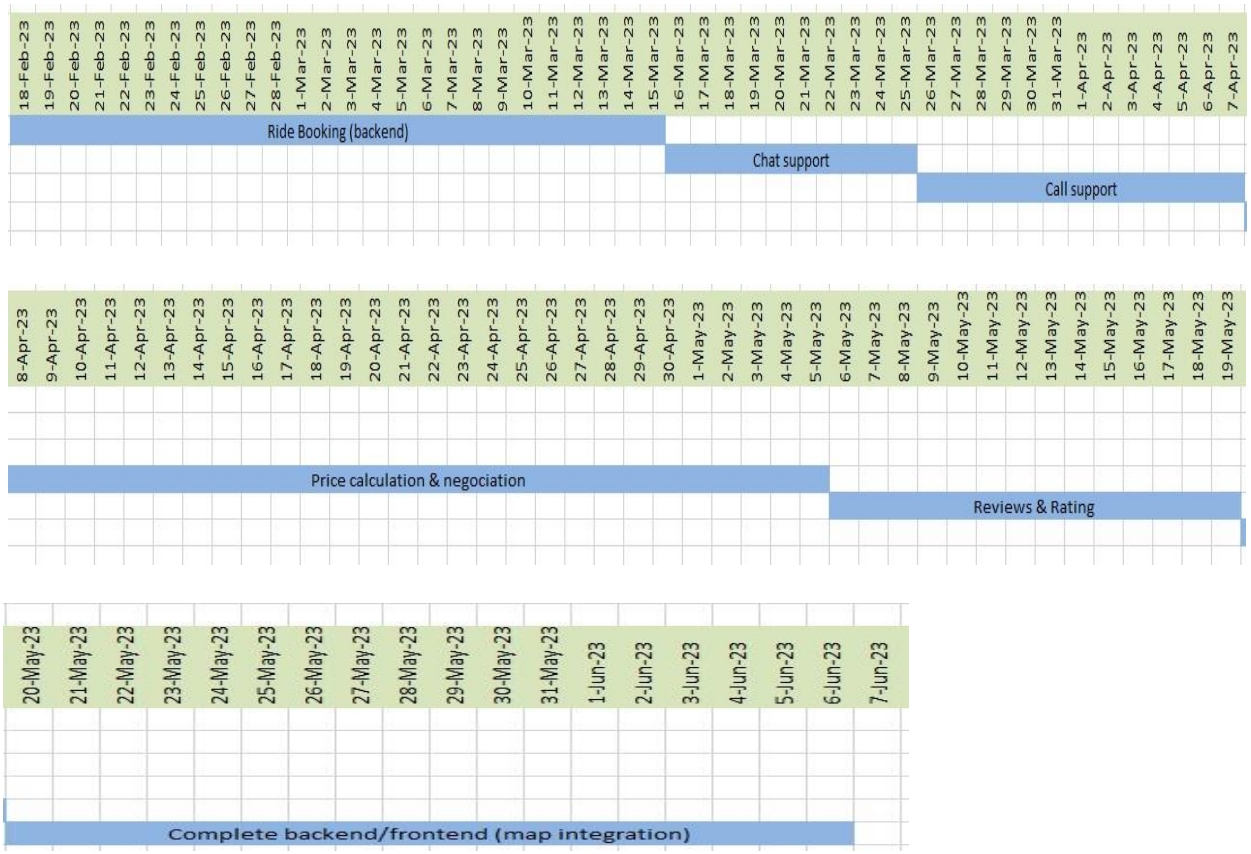


PROJECT NAME	PROJECT DURATION (DAYS)	PROJECT START DATE	PROJECT END DATE
Go-Car	109	18-Feb	6-June

FYP - 2

Task ID	Task Description	Task Duration	Start Date	End Date
1	Ride Booking (backend)	26	18-feb	15-Mar
2	Chat support	10	16-Mar	25-Mar
3	Push notification	10	16-Mar	25-Mar
4	Call option	13	26-Mar	07-Apr
5	Fare calculation	28	08-Apr	05-May
6	Reviews & Rating	14	06-May	19-May

7	Complete backend/frontend (with map integration)	18	20-May	6-June



12. Work Breakdown Structure

FYP – 1 (Work Breakdown Structure)

Project	**Go-Car**

Phase	**Feature**	**Design & Development**	**Code**	**Testing**	**Deploy**
Tasks	Login/Signup/Forgot Password/OTP	Create UML	Write Code	Conduct System Testing	Install System
	Driver registration	Create UML	Write code	Conduct system testing	Install system
	Add/ update/delete/view user info	Create Database	Test Code	Conduct Acceptance Testing	Train User
	User personalization	create Database	Test Code	Conduct Acceptance Testing	Train User
	Find pool/ offer pool (frontend)	Create UML	Write code	Conduct Acceptance Testing	install system

FYP – 2 (Work Breakdown Structure)

Phase	**Feature**	**Design & Development**	**Code**	**Testing**	**Deploy**
Tasks	Find pool/ offer pool (backend)	Database	write code	Conduct Acceptance Testing	Train User
	chat support	Create UML	write Code	Conduct Acceptance Testing	Train User
	Push notification	Create prototype	Integrate code	Conduct Acceptance testing	Install system
	call option	create prototype	integrate code	–	–
	fares calculation	database	write Code	Conduct Acceptance Testing	Train User
	Ratings and review	Testing the model	write code	conduct system testing	install system
	Map integration	Create prototype	Integrate code	–	–

13. References

https://www.sciencedirect.com/science/article/pii/S2772424722000257

https://www.ijrar.org/papers/IJRAR2001653.pdf

https://uxdesign.cc/ux-case-study-redesigning-a-carpooling-app-815581b3ea24

Software requirement specification (SRS):

1. Introduction

1.1 Purpose

As we all know transport service like car and bike are common mode of transport everyone needs on daily bases but not having vehicles or due to highly expensive recourses like petrol and CNG in today's time it has been very difficult to afford this type of service. So, we came up with a solution providing carpooling service by our application "GO-CAR" which help people/students to travel in low fares. Basically, a user pays for a seat not for all seats after completion of his/her ride.

1.2 Document Conventions

The document was created using the 'Arial' font style in Microsoft Word 2019. This document has been written in a fixed font size of 11 points.

1.3 Intended Audience and Reading Suggestions

The proposed audience of this document would be driver, user. A Driver offers a pool and user request a pool. The SRS document can be utilized in every situation involving the project's needs and selected plan of action.

1.4 Product Scope

Our project aim is to build an efficient application to give people interactive interface platform where they can avail carpooling service and to solve their expense problem through our service and feature. This project will be built on Flutter-Dart. We are planning to use following libraries and frameworks:

- Flutter for deployment of mobile applications.
- Dart
- Firebase for backend work **1.5 References**

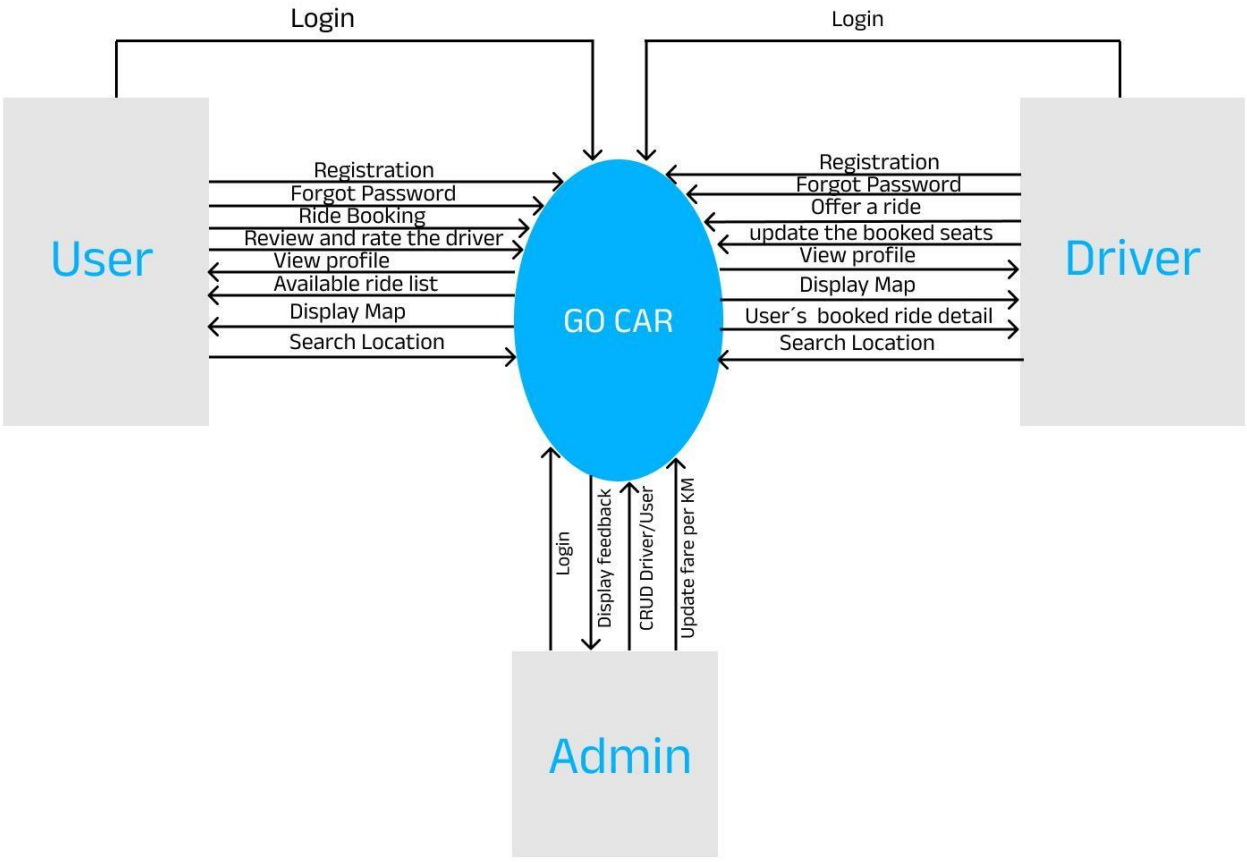
https://www.ijrar.org/papers/IJRAR2001653.pdf

https://uxdesign.cc/ux-case-study-redesigning-a-carpooling-app-815581b3ea24

2. Overall Description

2.1 Product Perspective

Following picture shows the product perspective.



2.2 Product Functions

Following are some main functionalities of our product;

- Login & Registration for USER and DRIVER
- User can book a ride according to available rides (pool)
- User can rate the driver
- User can check the ride details
- Driver can Offer a ride (pool)

- Admin can do operation like CRUD of driver or user.
- Admin can update fare price per KM.

2.3 User Classes and Characteristics

Following are the stakeholder who will be using the application:

Admin:

- Admin can add, delete, update, and view Driver and User Also can update fare calculation pricing method.

User:

- User will use all the functionality of the application like booking ride or cancel ride.
  User can rate driver.

Driver:

- Driver can offer a ride or pool to user and update the seat availability.

2.4 Operating Environment

Hardware:

- Android phone
- Processor Core i5-6th Generation
- Laptop (i5 8GB RAM) Software:
- Android studio
- Vs code
- Firebase
- Figma (Designing)

2.5 Design and Implementation Constraints

Following are the constraint for the application in order to work for the system properly in Environment:

1. Mobile phone should have minimum memory of 2GB+.
2. Mobile phone should have enough hardware space for application.
3. System should support Android Studio.

2.6 User Documentation

In this project the SRS will give a clear picture to the user in co-operating with the system. The Document will be in an essential comprehensible language. Moreover, the documentation will Include ERD diagram, context diagram, System sequence diagram, and use cases.

2.7 Assumptions and Dependencies

We assume that user must have few knowledges about applications and internet. Those users who are using laptop should have Windows to use the system. The RAM should be at least of 2GB. User must have Android Studio in their personal computers.

3. External Interface Requirements

3.1 User Interfaces

3.2 Hardware Interfaces

The hardware Interfaces for our project is android based smart phones with the connectivity of Internet and minimum 1 GB Ram. The phone users will generally use smart touch to interact with system while the laptop users will use keyboard to interact with the system.

3.3 Software Interfaces

GO-CAR is an application and it runs on all android based smart phones of
Version 5.0 or higher.
Technologies used are:

- Dart
- Flutter
- Figma for designing

Libraries:

- Firebase (Firestore)

3.4 Communications Interfaces

We will use HTTP communication standards to communicate with the internet as it is very common and secure.

4. System Features

These following features will be implemented in our application in order to interact with the system.

4.1 System Feature 1

Easy registration process and OTP verification:
In this process, the users will be able to register themselves in our app by providing a few basic details, then the user will be able to login into our GO-CAR App. Our App will send one-time password code to the user after enrolling themselves into our app. This technique will verify the user's identity **(High priority).**

4.2 System Feature 2

Push Notification:
Our App will send one-time notification to the drivers whenever there is any new booking done any user to the ride that driver has offers. **(High priority)**

4.3 System Feature 3

Find pool or ride:
User can find a pool from driver/available list and driver can offer a pool to users currently searching for a ride. **(High priority)**

4.4 System Feature 5

Book pool:
User can find a pool for available list and can book the ride according to his need. User can enter details of his/her source and destination and then perform booking process.
(High priority)

4.5 System Feature 6

Chat Support/feedback for USER:
User can ask queries from admin if he/she has problem regarding driver.
(Medium priority)

4.6 System Feature 7

Call Option for Driver and USER:
User can call to driver to find where the driver is waiting for him. Also driver can call user to meet at a place and start their ride. **(Medium priority)**

4.7 System Feature 8

Review and Ratings for Driver:
User can rate and give review to driver after completion of the ride, which help other to find top reviewed driver. **(Medium priority)**

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Performance requirements define acceptable response times for system functionality. Although the system is developed suiting for the least system performances, the Performance of the system will highly depend on the performance of hardware and Software components of the installing computer. When consider about the timing Relationships of the system the load time for user interface screens take no longer than Five seconds do. It makes fast access to system functions. The log in information shall be Verified within 8-10 seconds causes' efficiency of the system.

5.2 Safety Requirements

Our system would be maintaining backup of every data that is stored in the database in Excel sheet so we can restore original file if anything to original data goes wrong.

5.3 Security Requirements

We provide different validations (logins) for different type of users. We are responsible for Our security and validations. However, it is team responsibility to create secure logins and not share personal information with anyone he/she could not trust.

5.4 Software Quality Attributes

- Usability:

Our app is user friendly; we have design user friendly UI and the buttons to switch on next Screen which can be easily understand by users.

- Reliability:

For OTP Code, the code will send to the authenticate email address which is been provided by user.

- Flexibility:

Our App will be able to add new features and handle them easily.

- Maintainability:

Users review matters here after their feedback on annual average we will check performance if performance will be poor then we will try to improve it.

- Availability:

Application must be connected to internet and GPS when it is in use.

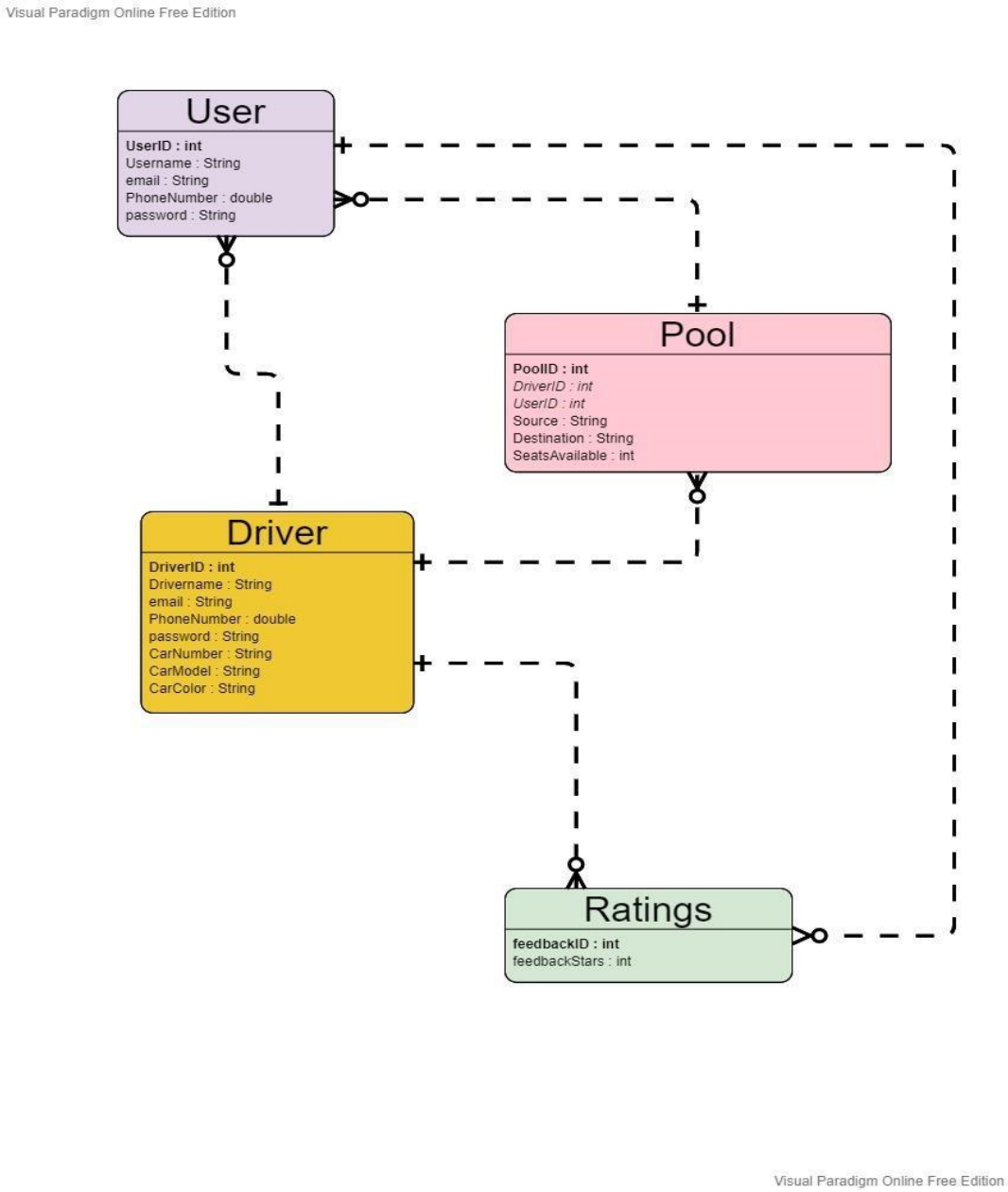
5.5 Business Rules

Admin will be given the rights of the application. Only admin can have full access of the System, like to add, update, and delete user and driver, updating fare (km) price, also he can view rating and reviews.

6. Other Requirements

Appendix: Analysis Models 6.1

6.1 ERD:



6.2 Use Case:

Task No. 1

Use case Name	Registration (sign up)		
Summary	To register an account		
Primary actor	User and Driver		
Precondition	User/Driver not registered yet		
Feature		**System Response**	**Actor Response**
		System will display Registration interface.	
			User/Driver enters his/her detail to register themselves.
		System will verify detail and check email address is valid or not	
			User/driver will press the sign up button
		System will store details in the database	
		System will show the message of successful registration	
Alternative Path	none		
Post condition	User/ Driver now registered and can login with their username and password.		

Task No. 2

Use case Name	Login		
Summary	To login into application		
Primary actor	User		
Precondition	Must have an account		
Feature		**System Response**	**Actor Response**
		System will display login interface.	
			Enters details associated with his/her account
			Press login button
		System will verify details	
		Successfully login and home page show if email address and password are in correct form.	
Alternative Path	If email address and password are incorrect: Show error message “Invalid email address or password” and redirects to login page		
Post condition	User/Driver can access the application.		

Task No. 3

Use case Name	Forgot password		
Summary	To reset the password		
Primary actor	Users /Driver		
Precondition	User/driver must have an account		
Feature		**System Response**	**Admin Response**
		Enter email address	
			Enter the required field and press reset password button.
		Send the link to reset password on the user given email	
			Open the link and set a new password
		Password changed successfully	
Alternative Path	If user/driver enters the email which is not registered so there will be error message.		
Post condition	New password updated.		

Task No. 4

Use case Name	OTP		
Summary	To check the user identity.		
Primary actor	User /Driver		
Precondition	Must register an account first.		
Feature		**System Response**	**Actor Response**
		System will display OTP interface.	
			Enters your mobile number.
			Press confirm button.
		System will send an email with OTP code to your Email.	
			User/driver will enter provided OTP code in OTP text field and press submit button.
		If OTP is correct, user/driver will be redirected to their login screen.	
Alternative Path	If OTP is incorrect user will be given an error. and redirected to OTP verification		
Post condition	Now user/driver can use the applications. Now driver can offer a pool. And user can accept or reject a pool.		

Task No. 5

Use case Name	Offer pool		
Summary	To generate a request for a ride by driver.		
Primary actor	Driver		
Precondition	driver must be logged in with an account.		
Feature		**System Response**	**Actor Response**
		System shows the map to driver and ride offering interface.	
			Driver enter details like source and destination, time, etc.
		System generate a ride request for user to join.	
			Driver wait for the user to join.
Alternative Path	None.		
Post condition	User(passenger) join the ride according to his need.		

Task No. 6

Use case Name	User Book ride		
Summary	To book a ride by driver according to his requirements and feasibility.		
Primary actor	User		
Precondition	User must be logged in with an account.		
Feature		**System Response**	**Actor Response**

		System show the available pools/rides to user.	
			User select the pool and confirm his seats.
		System generate a ride show navigation to driver.	
			Driver complete the ride.
		System generate fare for the ride.	
Alternative Path	User reject the ride.		
Post condition	User rate the driver.		

Task No. 7

Use case Name	Review and rating		
Summary	To rate the driver performance.		
Primary actor	User		
Precondition	Must complete a ride first.		
Feature		**System Response**	**Actor Response**
		System show the feedback box to give stars as an input.	
			User select value and input feedback and review about driver.
			Press confirm button.

	System will post/save the review.	
		Admin can view the rating and review of a driver.
Alternative Path	If user don't want to review he can close the screen.	
Post condition	Redirect to home screen. (ride booking screen)	

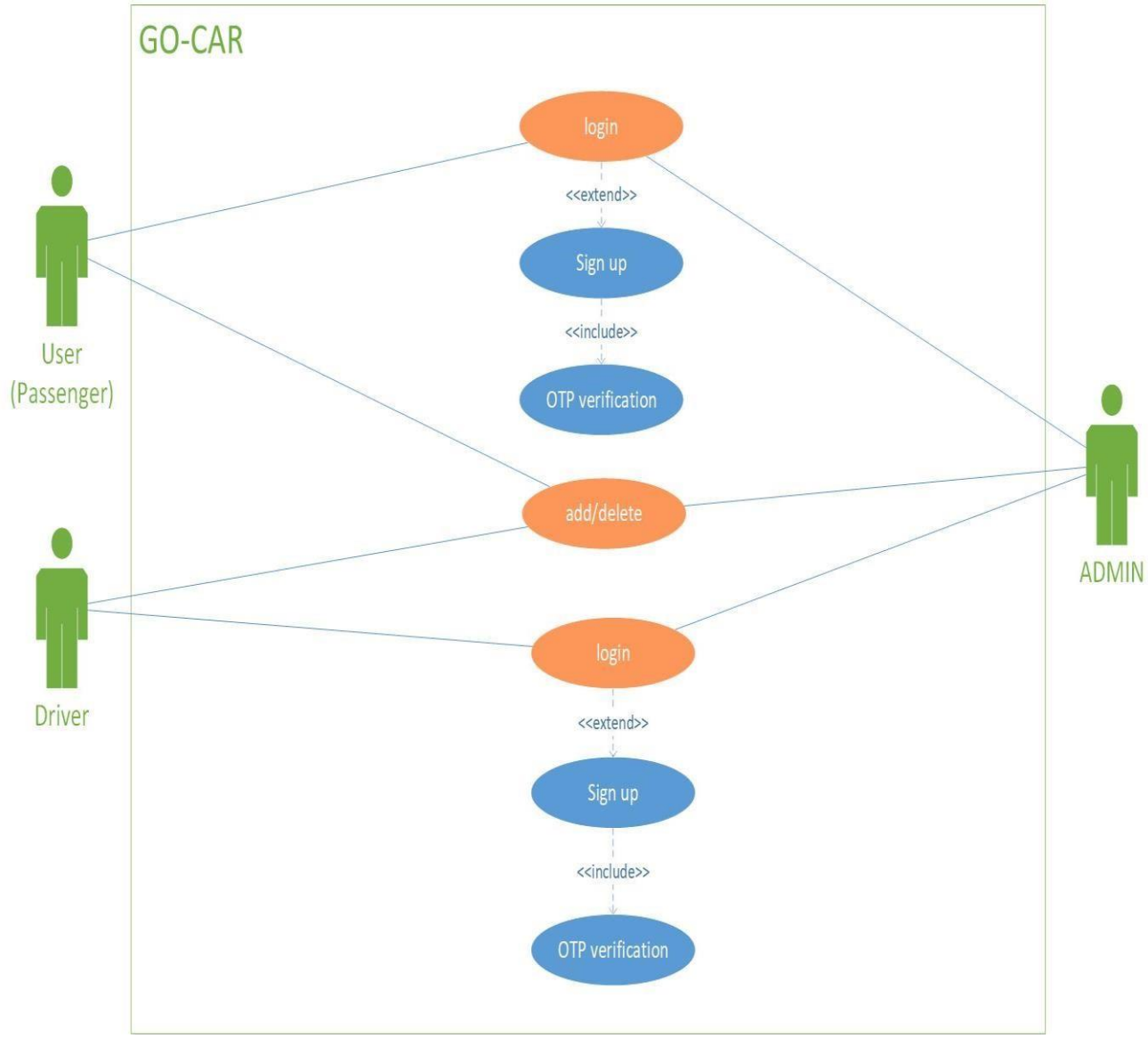
Task No. 8

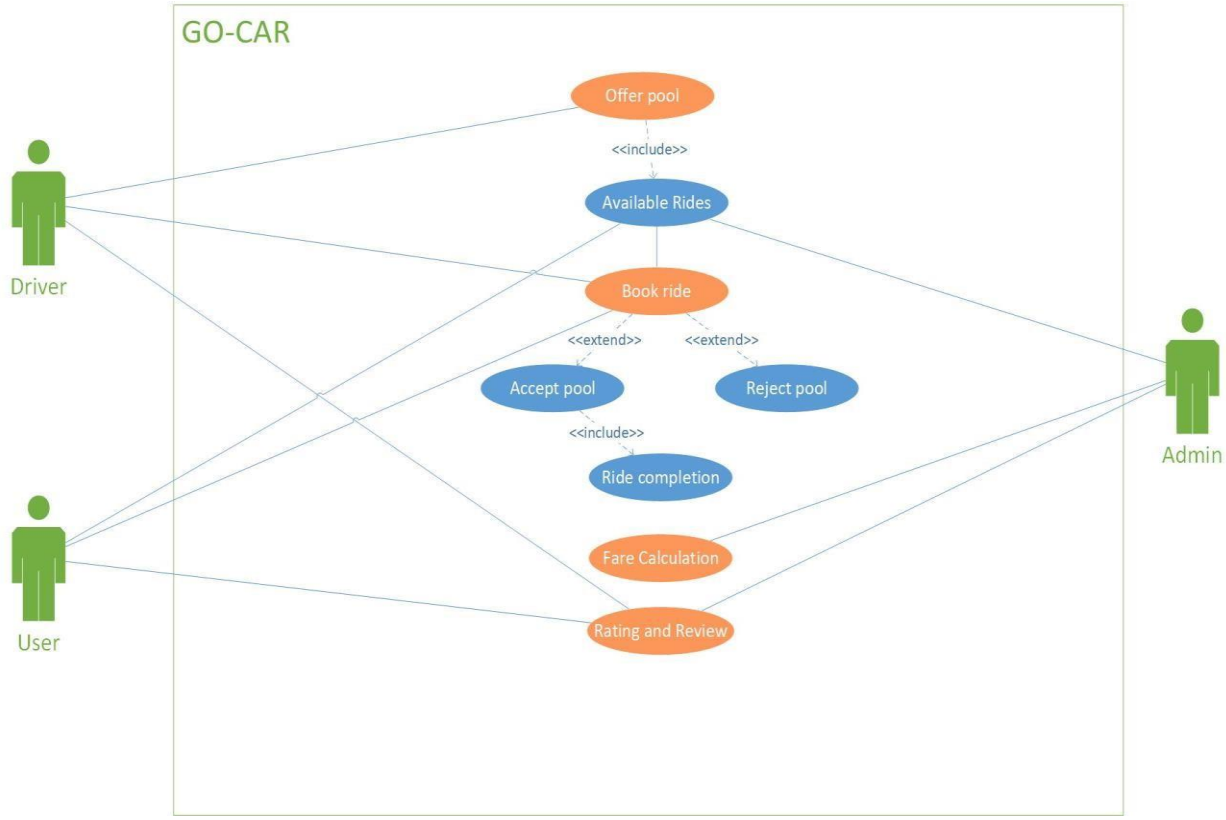
Use case Name	Push notification	
Summary	To get notification related to ride been booked	
Primary actor	Users	
Precondition	Must be login	
Feature	**System Response**	**Actor Response**
	System will send notification to the driver if any new user has book a ride.	
		User(Driver) will receive a notification regarding users which had book the ride.
		User(Driver) will open the notification.
Alternative Path	None	
Post condition	None	

Task No. 9

Use case Name	Fare Calculation	
Summary	To get the price/fare of the ride	
Primary actor	Users	
Precondition	Must be login & have booked any ride from available rides list.	
Feature	**System Response**	**Actor Response**
	System will send ride fare amount to the user if any user has book a ride.	
		User can read the amount the user had to pay for that ride.
Alternative Path	None	
Post condition	Redirect to home screen.	

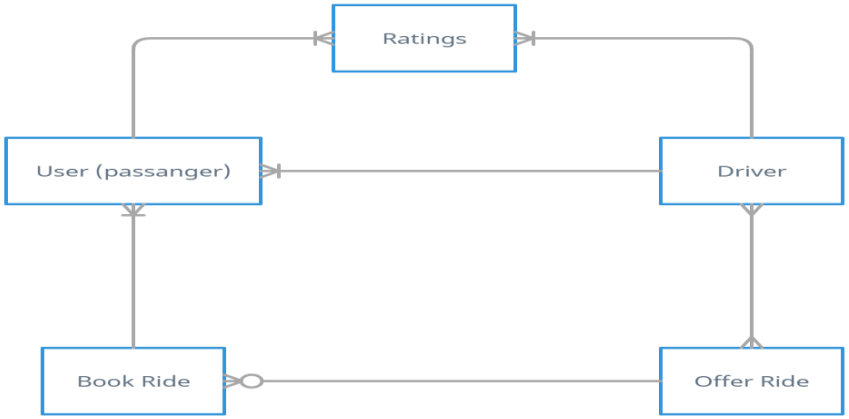
6.3 Use case (Uml):



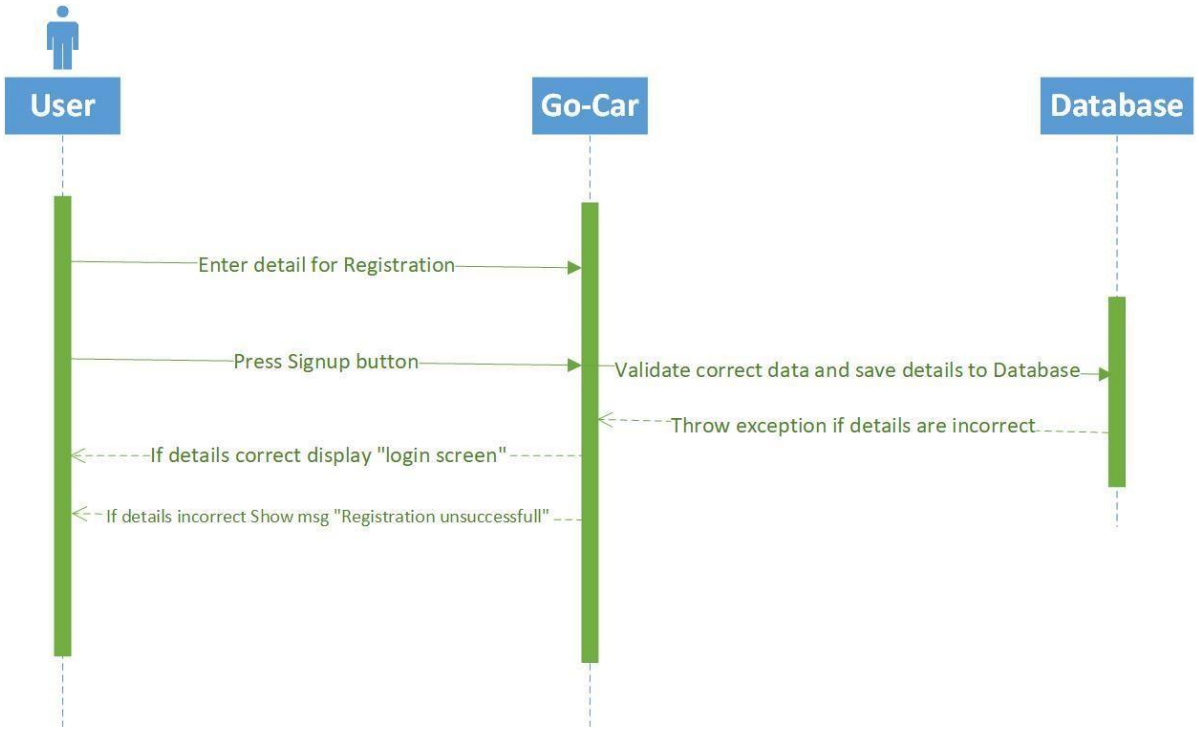


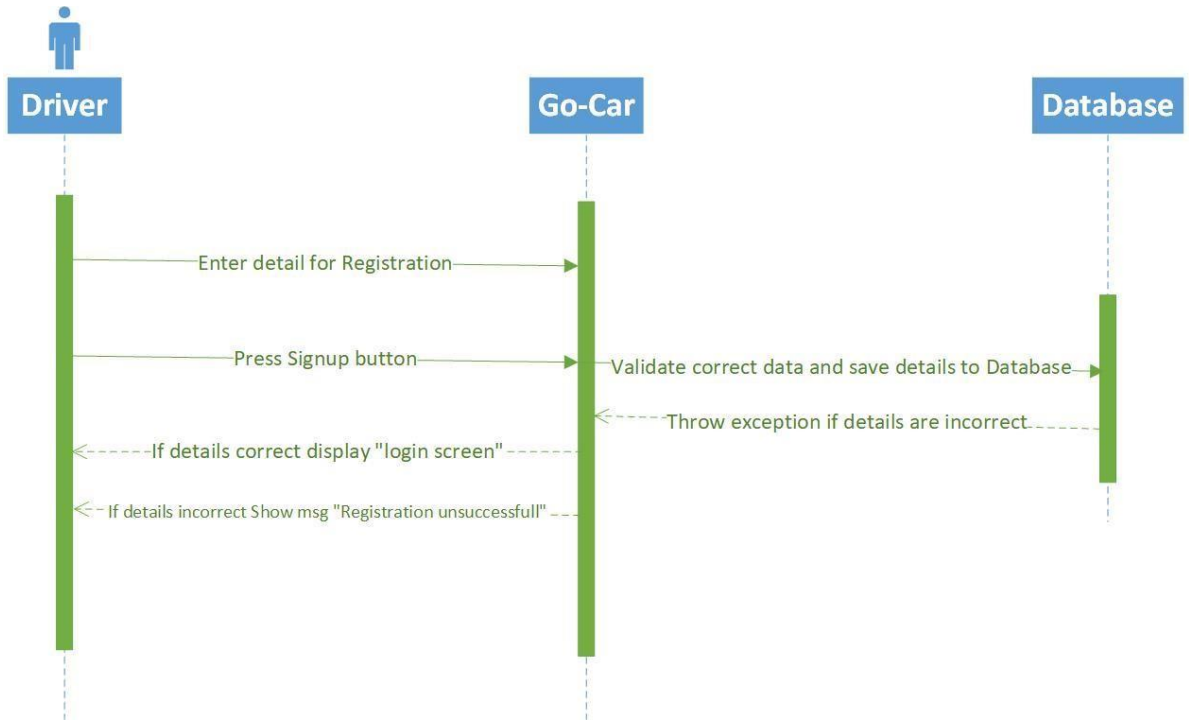
S

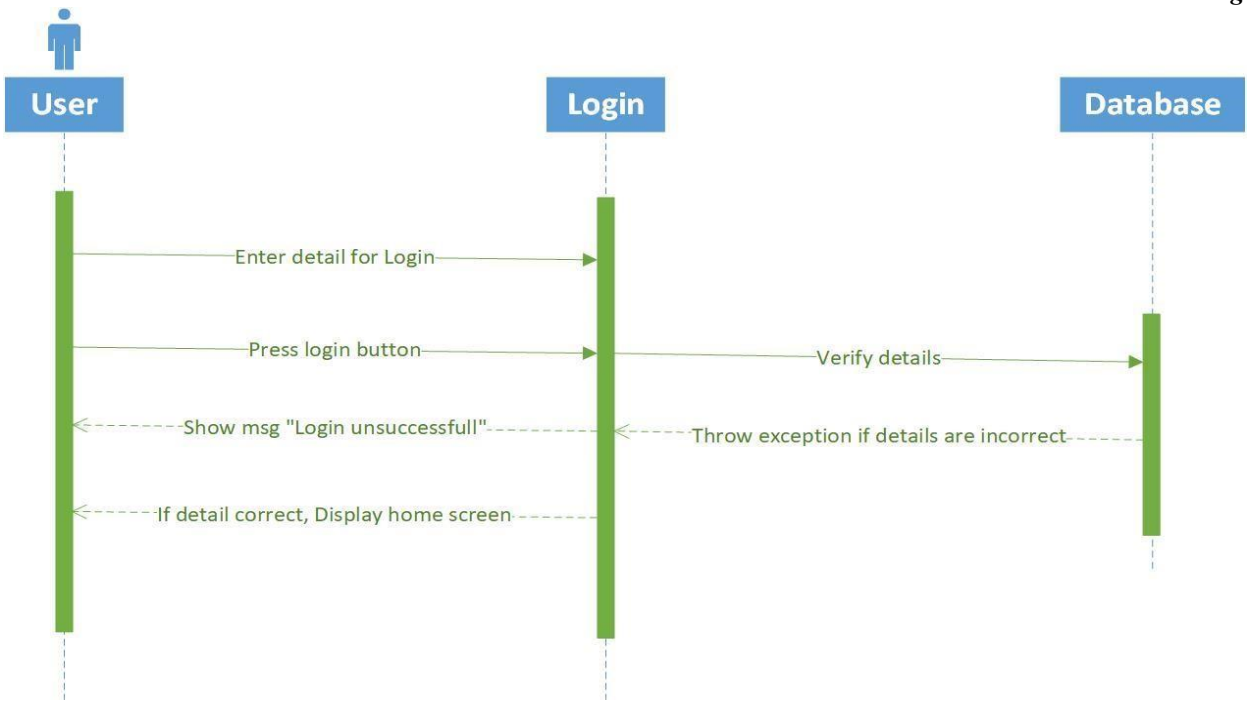
6.4 Domain Model:

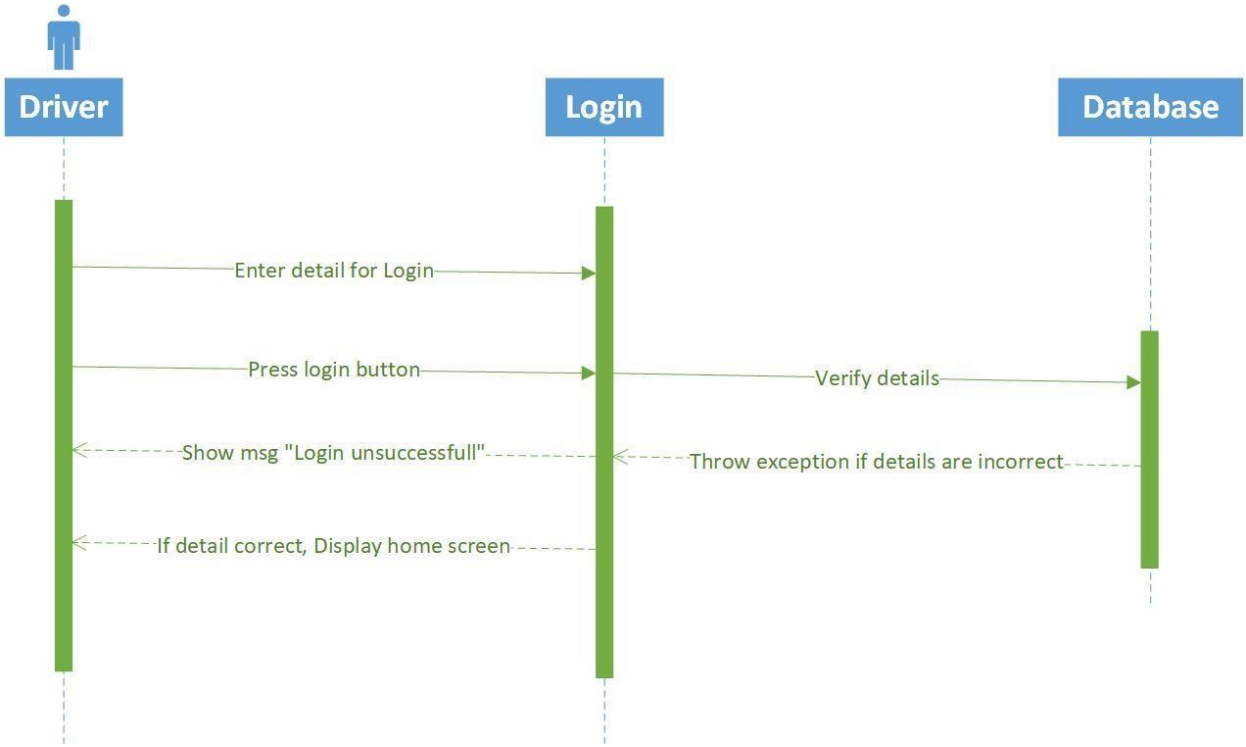


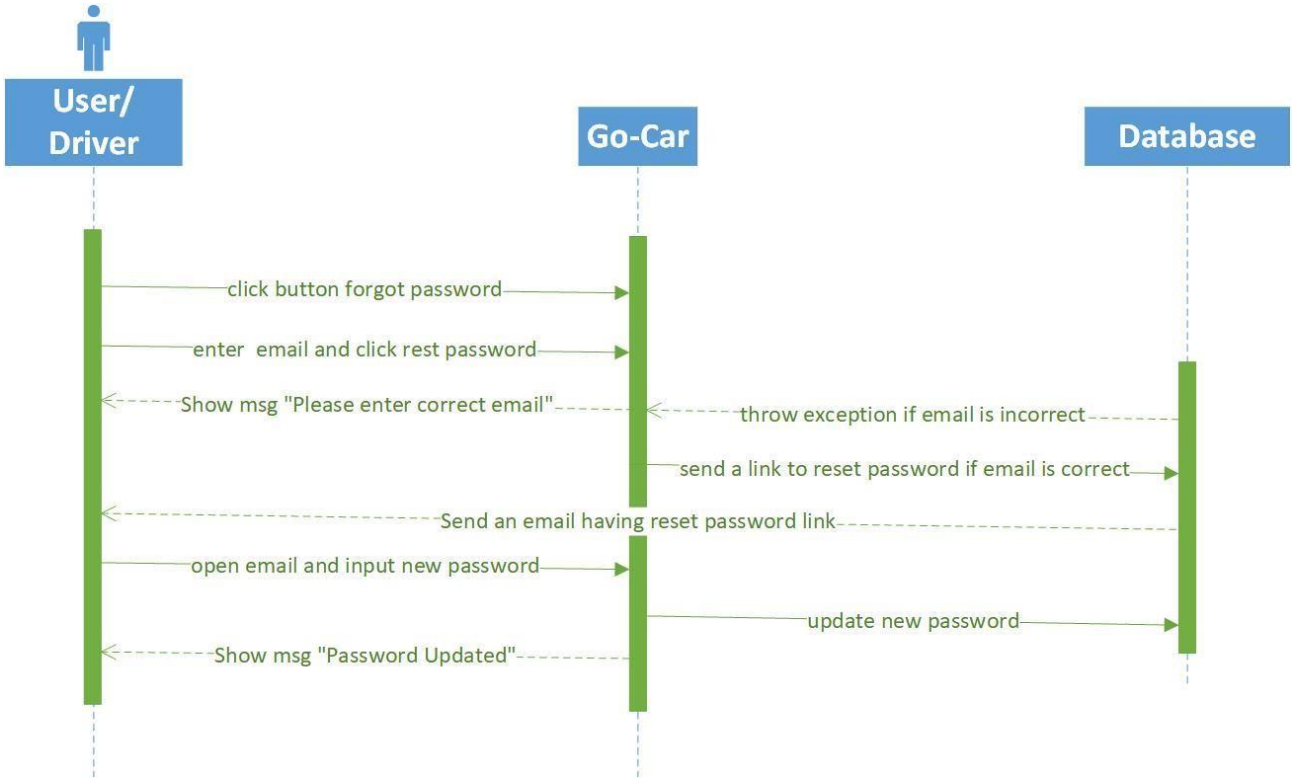
6.5 Sequence diagram:

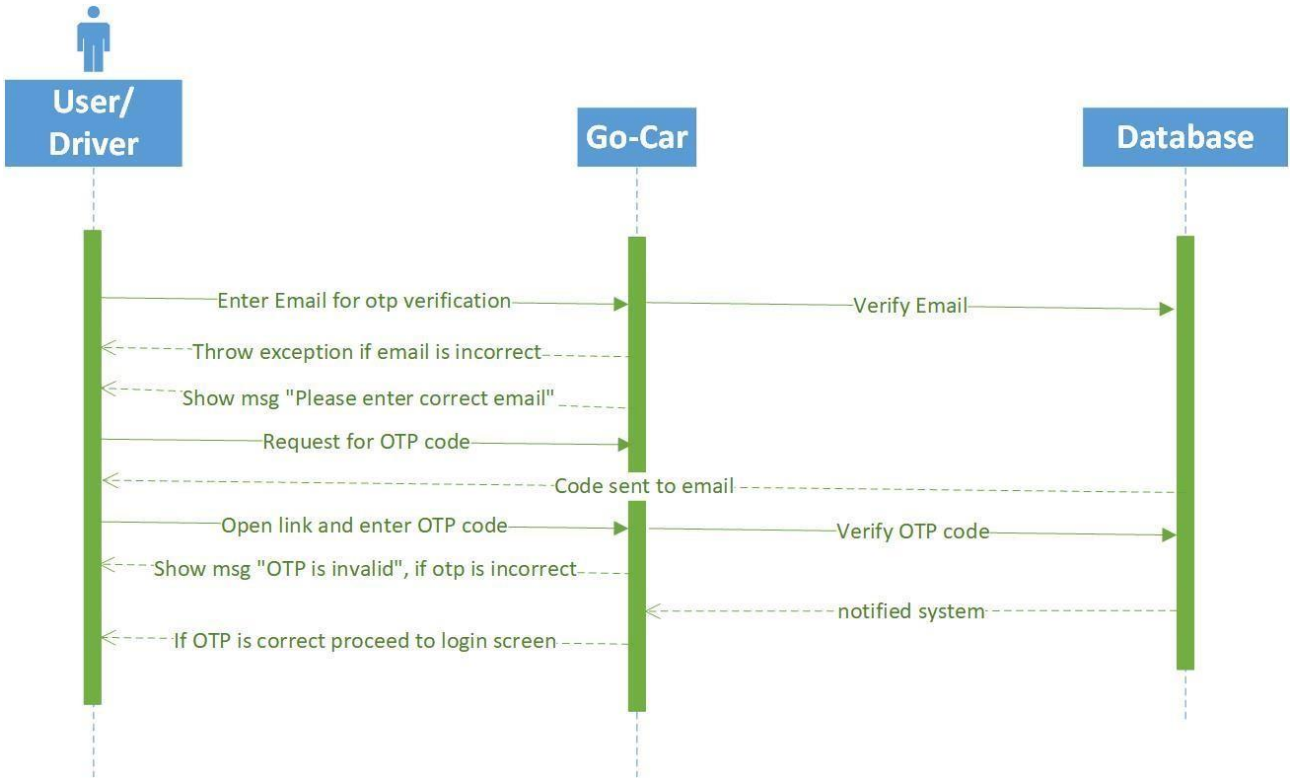


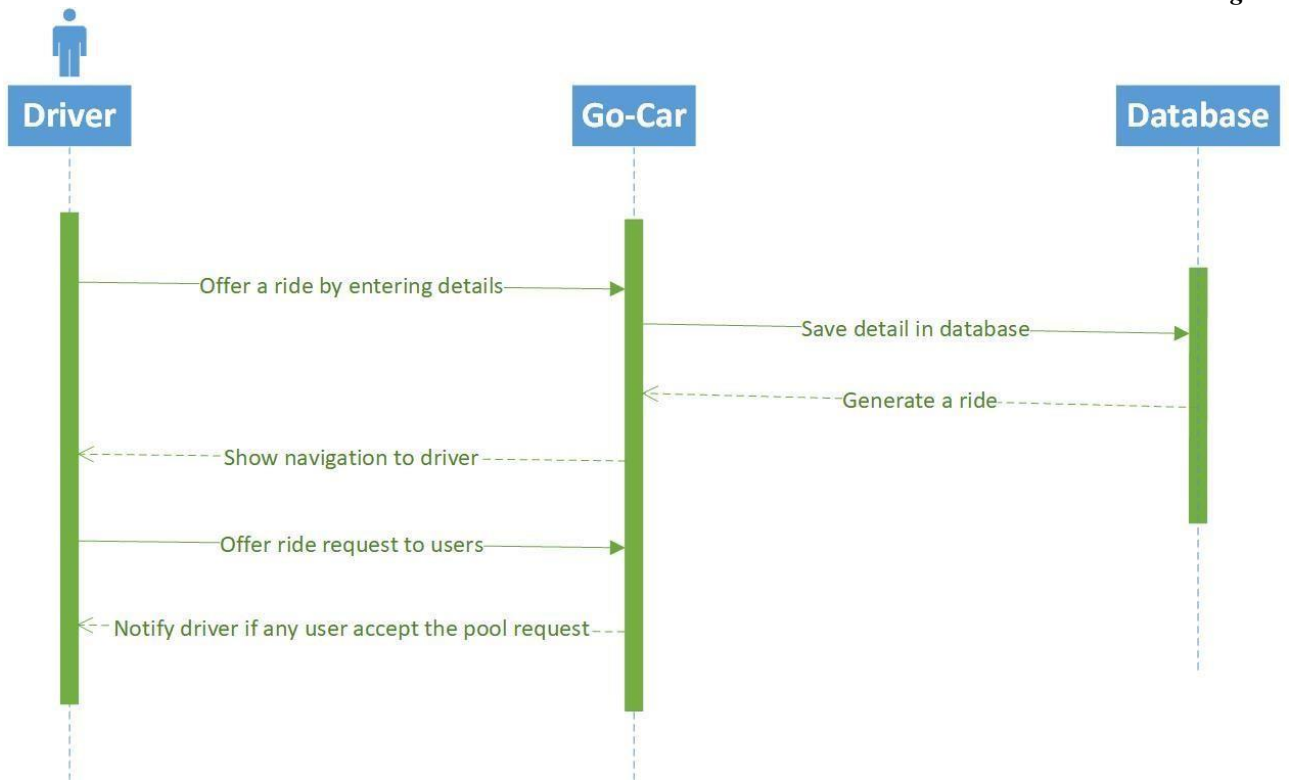


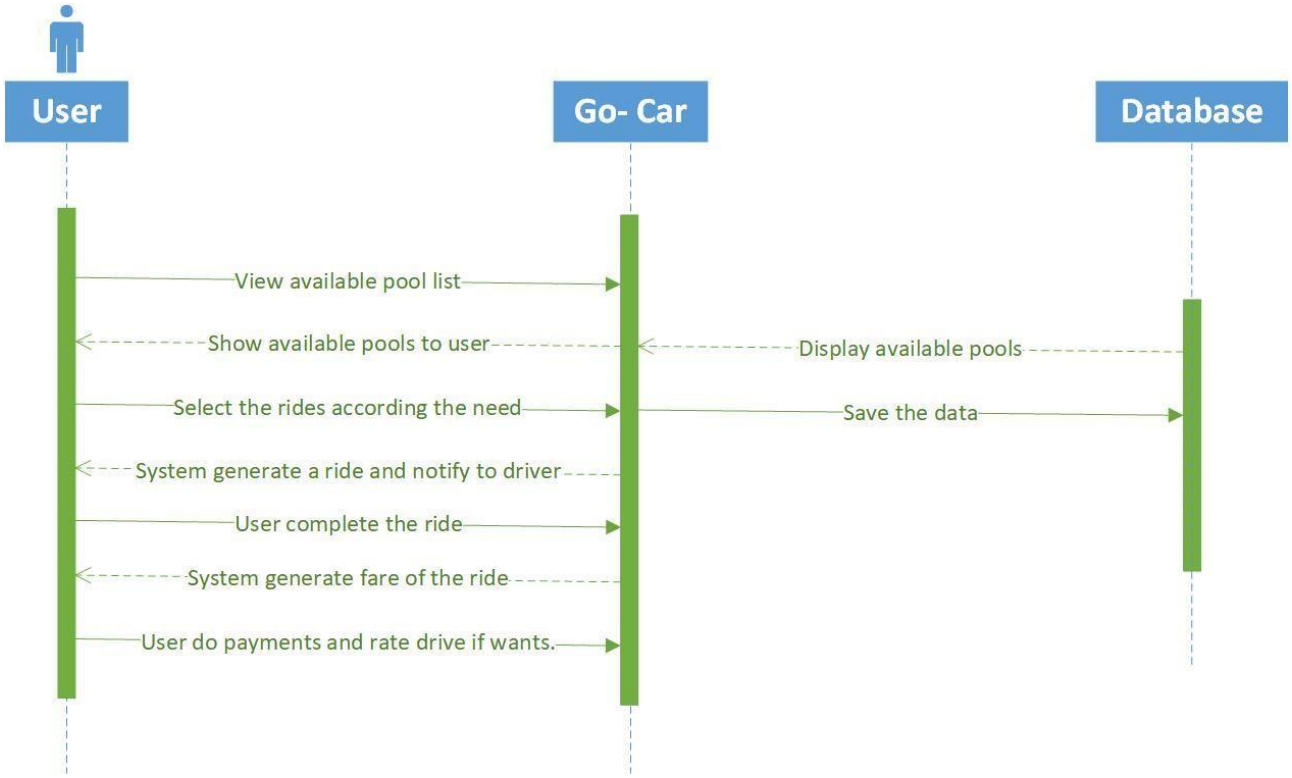

User
Login
Database
Enter detail for Login
Press login button
Verify details
Show msg "Login unsuccessfull"
Throw exception if details are incorrect
If detail correct, Display home screen

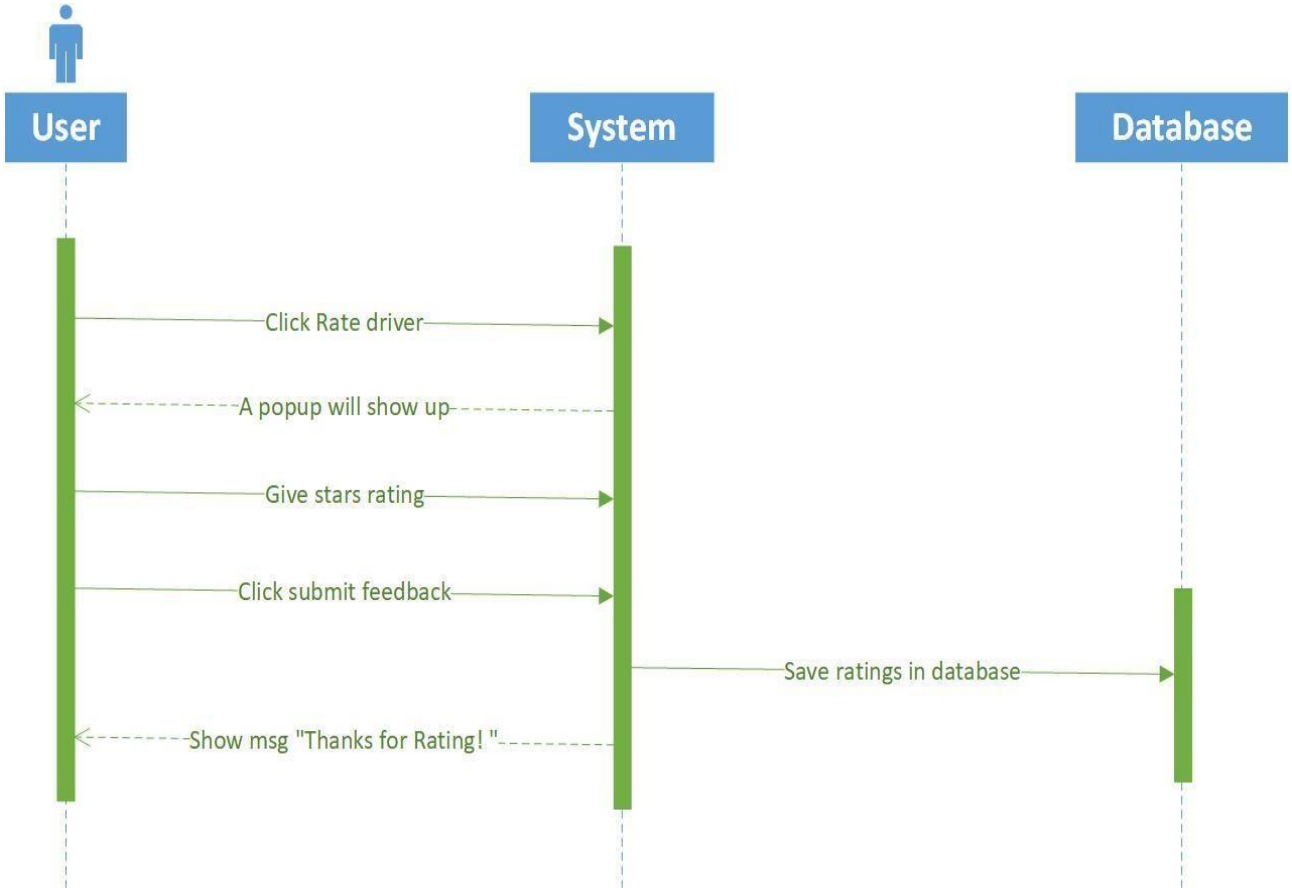

Driver
Login
Database
Enter detail for Login
Press login button
Verify details
Show msg "Login unsuccessfull"
Throw exception if details are incorrect
If detail correct, Display home screen

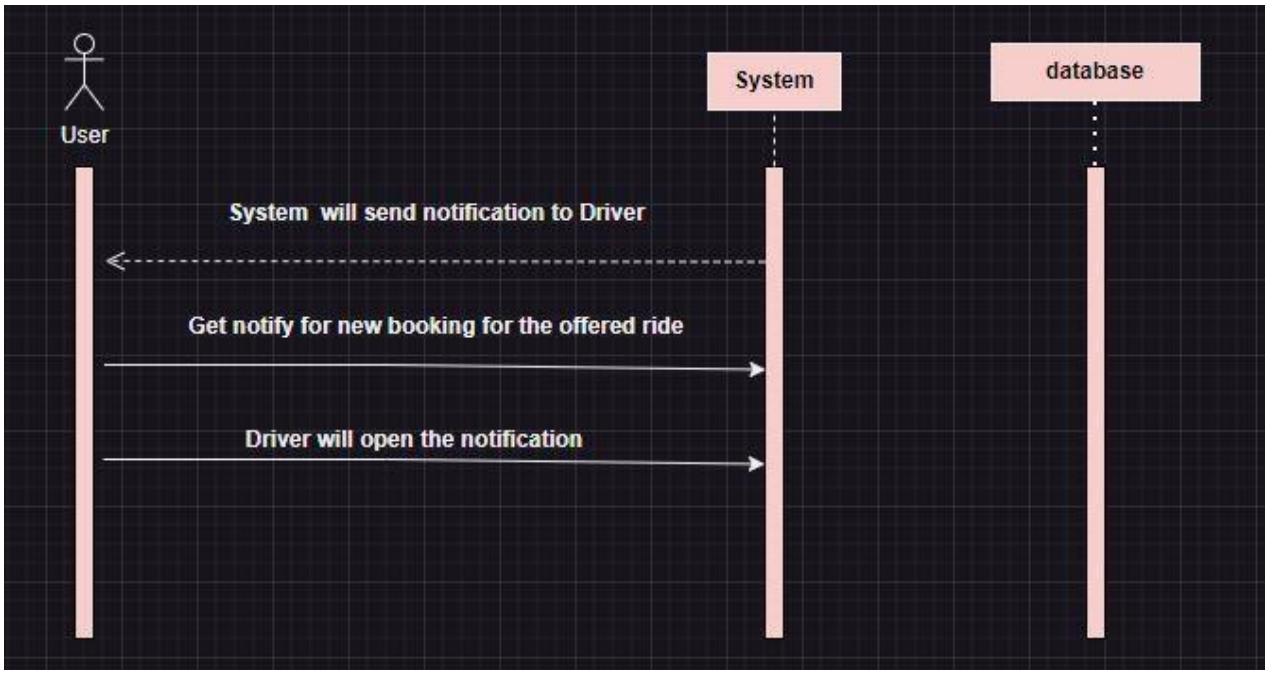

User/ Driver
Go-Car
Database
click button forgot password
enter email and click rest password
Show msg "Please enter correct email"
throw exception if email is incorrect
send a link to reset password if email is correct
Send an email having reset password link
open email and input new password
update new password
Show msg "Password Updated"


User/ Driver
Go-Car
Database
Enter Email for otp verification
Verify Email
Throw exception if email is incorrect
Show msg "Please enter correct email"
Request for OTP code
Code sent to email
Open link and enter OTP code
Verify OTP code
Show msg "OTP is invalid", if otp is incorrect
notified system
If OTP is correct proceed to login screen


Driver
Go-Car
Database
Offer a ride by entering details
Save detail in database
Generate a ride
Show navigation to driver
Offer ride request to users
Notify driver if any user accept the pool request


User
Go- Car
Database
View available pool list
Show available pools to user
Display available pools
Select the rides according the need
Save the data
System generate a ride and notify to driver
User complete the ride
System generate fare of the ride
User do payments and rate drive if wants.


User
System
Database
Click Rate driver
A popup will show up
Give stars rating
Click submit feedback
Save ratings in database
Show msg "Thanks for Rating! "


User
System
database
System will send notification to Driver
Get notify for new booking for the offered ride
Driver will open the notification

6.6 Testing Documentation:

Test Cases Only (20-50%) done;

Test Case 1 Registration (Sign up)

Test Case ID	Test Case Name	Test Case Summary	Test Case Steps	Expected Results	Actual Results
TC001	Registration	The aim of this test is to confirm that the registration process of adding new user or driver is functioning properly.	1.User/Diver enters his/her details to register themselves. 2.User/Driver then press the sign-up button.	User/Driver will be registered and created his/her details to register themselves.	Everything proceeded correct as expected.

Test case 2 (Login)

Test Case ID	Test Case Name	Test Case Summary	Test Case Steps	Expected Results	Actual Results
TC002	Login	To verify that by clicking login, after entering a valid email and password.	1.Enter details associated with his/her account. 2.Press login button.	Successfully login if: The email and password are correct.	If email and password are correct, the result is expected.

Test case 3 Forgot password

Test Case ID	Test Case Name	Test Case Summary	Test Case Steps	Expected Results	Actual Results
TC003	Forgot password	To reset the password through email.	1.Enter the required field and press forget password. 2.Open the link and set the new password.	Password changed successfully .	If email is correct, so the result is expected.

Test Case 4 (OTP Verification)

Test Case ID	Test Case Name	Test Case Summary	Test Case Steps	Expected Results	Actual Results
TC004	Email OTP Verificatio n	We are verifying that user are registered with fake or real email address.	1.User/Diver enters his/her Email Address and press OTP Button. 2.User/Driver will enter correct OTP Code and press verify button.	Send OTP code on Email address. Home page should be display on the screen.	Everything proceeded correct send OTP Code on Email. Home page would be displaying on the screen for User/Driver.

Test Case 5 (view profile)

Test Case ID	Test Case Name	Test Case Summary	Test Case Steps	Expected Results	Actual Results

TC005	View Profile	The User can view their account details by this feature view profile.	1. User/Diver will have to register and login our App. 2. User/Driver will have to click on profile button form bottom navigation.	User will able to view the Bottom navigation bar in which there will be option as my profile. The User will be navigate to next page which will show the user account details.	If everything is correct then profile would been displayed. Display the profile details.

Test Case 6 (Offer Ride)

Test Case ID	Test Case Name	Test Case Summary	Test Case Steps	Expected Results	Actual Results
TC006	Offer Ride	The driver can offer a request/ride to other user offering his/her service.	1. Diver will have to first register him as driver and login our App. 2. User/Driver will have to click on “offer ride” and then provide the details and offer the ride.	User will able to offer the ride to other user so that driver can generate the offer ride. The User will be navigate to home page after user completed the offer ride process.	If everything is correct and complete the ride would be offered. Display the ride in available rides.

Test Case 7 (Book Ride)

Test Case ID	Test Case Name	Test Case Summary	Test Case Steps	Expected Results	Actual Results
TC007	Book Ride	The user can book a ride to avail the service according to his/her preference.	1. User will have to first register him as driver and login our App. 2. User will have to click on “Book Ride” and then select the ride which suits his/her requirement and then book that ride.	User will able to book the ride which ever ride suits his/her requirement. The User will be navigate to home page after user completed the book ride process.	If everything is correct and complete the ride would be booked.

Test Case 8 (Ratings)

Test Case ID	Test Case Name	Test Case Summary	Test Case Steps	Expected Results	Actual Results
TC008	Ratings	To get overall idea of user experience.	1. Click on rating option and Select the number of star. 2. User will have to click on “Submit button”.	Upon successful submission it will store ratings in the database.	If everything is correct so the result is expected and then user will redirect to home page.

Test Case 9 (Push Notification)

Test Case ID	Test Case Name	Test Case Summary	Test Case Steps	Expected Results	Actual Results
TC009	Push Notification	To get notification to driver when any user try to book a ride he/she has offered.	1. Driver will received notification for ride been booked 2. User will see the notification 3. User will have to click on the notification to open app.	Upon successful driver will get notification for ride been booked by any one	If everything is correct so the result is expected and then notification received.

6.7 Iteration plan:

S.No.	Features	FYP-I Iterations			
		Monthly Iteration-I	**Monthly Iteration-II**	**Monthly Iteration-III**	**Monthly Iteration-IV**
F1, F2	User Registration/login Driver Registration/login	Requirements (70%)	Requirements (100%)		
		Design (100%)	Design(100%)		
		Implementation (70%)	Implementation(100%)		
		Testing	Testing		
F3	OTP verfication		Requirements		
			Design		
			Implementation		
			Testing		
F4, F5,F6,F7	Driver(Offer Ride), user request ride, available rides fare calculation formula only		Requirements	Requirements	
			Design	design	
			Implementation	Implementation	
			Testing	Testing	
F7,F8,F9	Fare Calculation, map integration, driver ratings				Requirements
					Design
					Implementation
					Testing
Output Features		F1, F2	F2, F3,F4,F5	F4, F5,F6,F7	F7,F8,F9

S.No.	Features	FYP-I & FYP-II Iterations			
		Monthly Iteration-I	Monthly Iteration-II	Monthly Iteration-III	Monthly Iteration-IV
F1	User Driver Registration/Login	Requirements			
		Design			
		Implementation			
			Testing		
F2,F3,F4	OTP Verification, some parts of UI, Map integration	Requirements (50%)	Requirements (100%)		
			Design		
			Implementation(100%)		
F4,F5,F6,F7,F8	Driver(Offer ride), User (Book ride), availble ride, Serching places		Requirements		
			Design		
			Implementation	Implementation(100%)	
F9,F10, F11, F12,F13	Fare calculation, Push notification, ride Information, Reviews & rating, call		Requirements(100%)	Requirements(80%)	Requirements(100%)
			Design(50%)	Design(100%)	Design(100%)
				Implementation	Implementation(100%)
Output Features		**F1**	**F2, F3,F4**	**F3, F4,F5,F6,F7,F8**	**F9,F10,F11,F12,F13**

Software Design Specification (SDS):

7. Introduction:

7.1 Purpose of this document

The basis of this document is to give the details about the system working and its assumptions which are made while manufacturing the system etc. the targeted audience for this document will get to know about the system in detail and also will be able to understand about its architecture and work flow for specific task which is described in detail for both designers and stakeholders.

7.2 Scope of the development project

Our project primarily focuses on providing a platform for user (student/staff) & driver (student/staff) to resolve this problem in effectively manner and satisfying both needs. To overcome the problem of transportation and high expense of traveling we are providing a platform to reduce their problem through which they don't have to face problem in travelling.

- OTP Verification: In this feature it will verify the Email Address of user by sending them an OTP Verification code when user is doing registration.
- Easy registration process: In this process, the users & Drivers will be able to register themselves in our app by providing a few basic details, then the user will be able to login into our GO CAR App.
- Offer Ride: Driver will offer a ride with some details like (source, destination, date & time, etc.) to the users.
- Book Ride: Users will be able to book the ride according to their requirement matching at that time of booking from the available rides.
- Available Rides: At the time of booking users will able to view available rides at that time to book the pool.
- Chat Support: User can ask queries from admin if he/she has problem regarding driver.
- Call Option: User can call to driver to find where the driver is waiting for him. Also driver can call user to meet at a place and start their ride.
- Review & Rating: User can rate and give review to driver after completion of the ride,
- Embedded Map: Users and Drivers can see the location and nearby places through the map which is implanted.
- Fare Calculation: It will give the final fare/cost of ride to the user.
- Push Notification: It will notify driver whenever any user book his/her ride.

7.3 Definitions, acronyms, and abbreviations

- **Flutter**: It is an open source development kit for building a software which is designed by Google.
- **Firebase**: This platform is also created by Google for building the web applications and mobile applications.

7.4 References

- https://docs.flutter.dev/
- https://www.ijrar.org/papers/IJRAR2001653.pdf
- https://uxdesign.cc/ux-case-study-redesigning-a-carpooling-app-815581b3ea24

7.5 Overview of document

There are total 8 sections and each section consist of a details about the system.

Section 1.0: This section is an introductory part which describes the purpose of the document, its scope,
References and the brief overview of the document.

Section 2.0: This section defines about the system architecture which includes database and the program data. It's also describe the relationship between the different components.

Section 3.0: This section consist of the detailed description about the each software component.

Section 4.0: This section describe about the user interface design.

Section 5.0: In this include the subsections which deals with how reuse is playing a role in our product
Design, how it is playing role in product implementation.

Section 6.0: This sections help readers and users understand the reasoning for these decisions

Section 7.0: This section involves the pseudo code implementation for the different components used
In the project.

Section 8.0: This section present diagrams such as class diagram, object diagram, state diagram, activity diagram, sequence diagram, use cases and much more.

8. System architecture description:

This section provides an overview and rationale for the program's data and architectural design decisions.

8.1 Section Overview

In this section we will mention few things about general constraints, Data Design, program structure, and the Alternative considered.

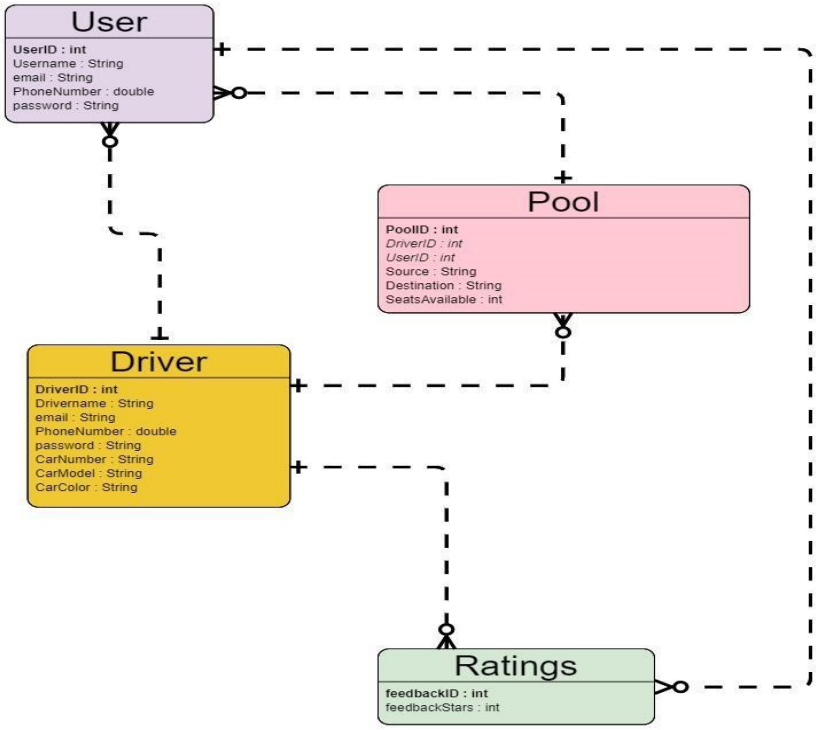
8.2 Section 2.2

Which is general constraints in this we will discuss about our both hardware and software environments and the limitations of the system.
In **section 2.3** we will include the database diagram mentioning the entities and attributes values based on constraints and relationships.
In third **section 2.4** we have brought about the architectural model and its different components ascertaining the entire project's visual interface.
In **section 2.5** the alternative of the architectural model for our system

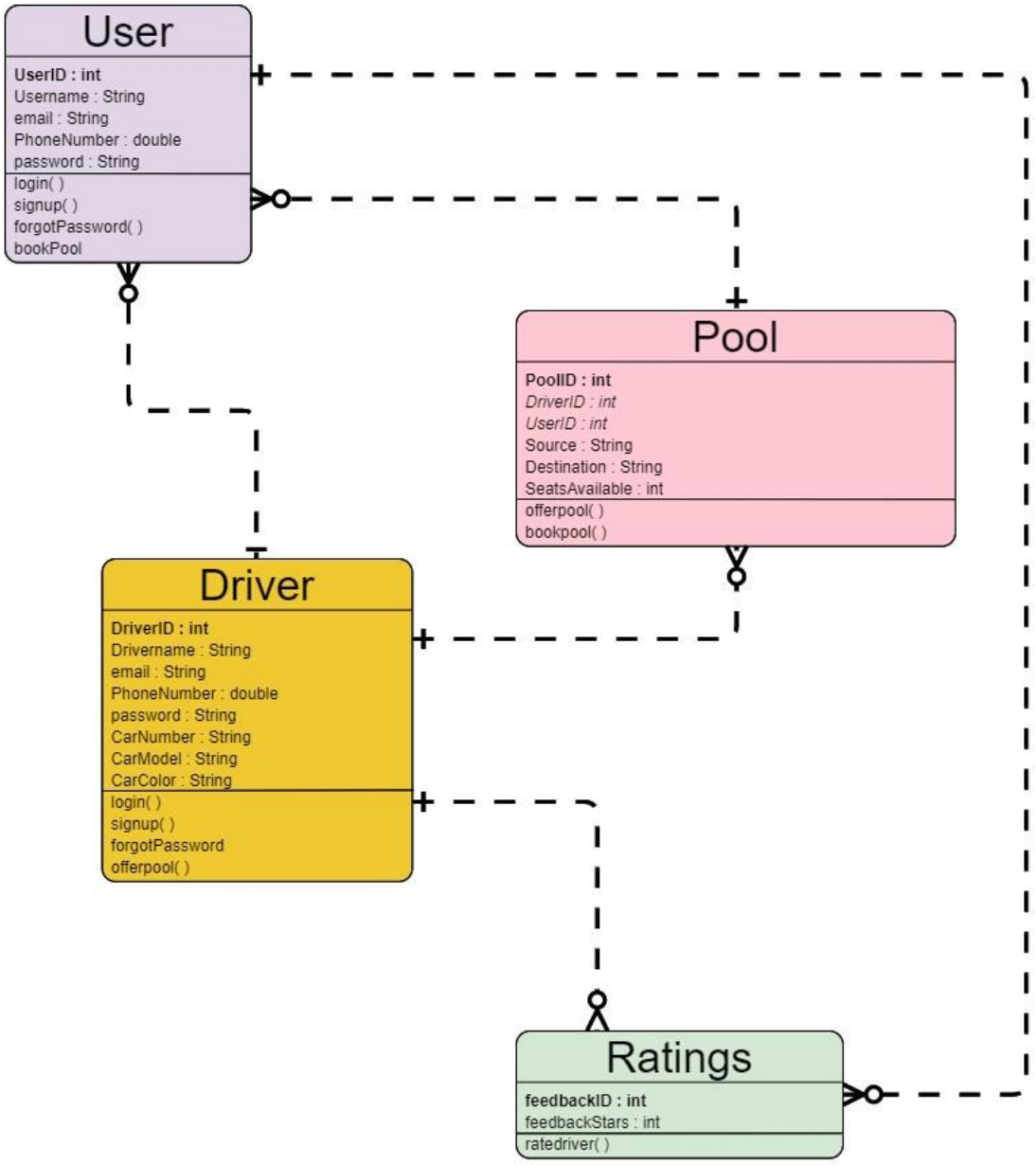
8.3 General Constraints

We are building a platform which will be able to carry the data of each if any complications arise each day that will be handled and solved in a correct manner without creating any kind of obstacle for our application. GO CAR App is develop on Android Studio and there is no hardware required. The Limitations will be those users who are not registered in our app will not be able to use our Application. Secondly to use the App network and GPS location in mandatory. Moreover, the users with the fake email address will not be able to registered in our GO CAR App

8.4 Data Design.



8.5 Program Structure



8.6 Alternatives Considered:

N/A

9. Detailed description of component:

9.1 Section Overview

This section will cover the basis of implementing the system. In the section 3.2 we will components and also the component template description.

9.2 Component and Detail (include a sub-section for each component)

9.2.1 Component 1 forget password screen:

9.2.1.1 Description:

Through this forget password screen the user can reset their password.

9.2.1.2 Data Members:

It is a type of class in which user can use this component for resting the password and it is visible to everyone.

9.2.1.3 Methods:

- We create the class of forgot password and encompass the function of forget password.
- For resting the password a link will send to user on email to reset the password.
- After the user inputted his details it will store in firebase.
- If there will be any issue it will display error message.
- User will redirect to first screen login as (User/Driver).

9.2.1.4 Forget password screen component:

Identification	Forget Password screen
Type	Class
Purpose	The purpose of this screen/class is to reset the password if user forget his/her password they can reset it. Only register user would be able to use this feature.
Function	The forget password screen contain field like email address and when user will enter the email & press the button then user will send to login user/driver screen.
Subordinates	The forget password screen contain the links to following screen: User/driver Login Screen
Dependencies	The forget password screen contain the links to following screen: User/driver Login Screen
Interfaces	The links on the screen are at the bottom of the screen with font style normal and little grey shade color so that user can notice it easily. The screen is created using standard resolution on personal Computer. We use flutter to design forget password screen.
Resources	The access is only for the valid users. This access is used to reset password of user and check to make sure that password have bene reset or not.
Processing	The only type of processing is required to reset the password by entering data in given fields and make sure to check link that is being send via email.
Data	The data supplied by the system are fields, the user must exist. The data given by the user is the appropriate information needed to reset the password.

9.2.2 Component 2: Register

9.2.2.1 Description:

The use of registration of an account in our application is compulsory to use the Application to access all the unique features of our application. This is both for user as well as driver.

9.2.2.2 Data Members:

It is a type of class which we used in our system through which user can register them With our application.

9.2.2.3 Methods:

- We make registration class and include registration function.
- Function will store user's credentials in firebase database.

- If something is wrong, it will give error message.
- User then redirect to OTP page once all details are filled.

9.2.2.4 User / Driver Registration Screen Component:

Identification	Registration Screen
Type	Class
Purpose	Registration screen is made for all the new users so they can register with our system to use application.
Function	There are some field in registration screen like user name email address, password, etc. When user enters all the required fields and press register button, all the data will be saved in our database and user will be directed to OTP screen.
Subordinates	The registration screen contain the links to following screen: • Login Screen • OTP Screen
Dependencies	The Registration screen links to: LoginAndSignup screen
Interfaces	The screen is designed to be easy to view using the resolution standard on mobiles. We tried to create simple and easy interface to use registration layout using flutter.
Resources	Database access requirements access to the violator information found in the appropriate database. This access is for the valid users.
Processing	The only type of processing is required to input information into the textfield and navigating to other pages. Each page directs the user to OTP.
Data	The data supplied by the system are fields which user needs to complete by providing valid information. It is validated with a query against database.

9.2.3 Component 3 Login:

9.2.3.1 Description:

The use of login of an account in our application is that user/driver can login with their registered account to access are App features like offer ride, book ride.

9.2.3.2 Data Members:

It is a type of class which we used in our system through which user can login with their registered account and use the application.

9.2.3.3 Methods:

- We will make login class and include login function.
- Function will store user's credential in firebase.
- After the user inputted his details it will be verified through firebase. If something is wrong it will give error message Redirect to dashboard.

9.2.3.4 Login screen component:

Identification	Login screen
Type	Class
Purpose	The login screen assures that the only authorized users and driver can access the application
Function	There are few fields in login screen like email address and password, and if user enters correct details so user will be redirected to dashboard else error will be showed.
Subordinates	The Login screen contain the links to following screen: home Screen
Dependencies	The Login screen links to: LoginAndSignup screen
Interfaces	The screen is designed to be easy to view using the resolution standard on mobiles. We tried to create simple and easy to use login layout.
Resources	The Database access requirements access to the violator information found in the appropriate database. This access is for all valid users (like driver/user).
Processing	The only type of processing is required to input information into the textboxes and navigating to next pages. Each page directs the user to main page/ main screen.
Data	The data for registering is the email address and password entered by the user. It is validated with a query against database for validating.

9.2.4 Component 4 OTP Verification:

9.2.4.1 Description:

The use of OTP screen in our application is that during registration stage user will verify it's email address by writing 6 digit code.

9.2.4.2 Data Members:

It is a type of class which we used in our system through which user will enter email address and a verification code will be send via email.

9.2.4.3 Methods:

- We will OTP class and include OTP function.
- User will enter his/her email address and click on submit.
- Function will send verification code in mentioned email address.
- If code is correct user will be directed to first screen or we can say user/driver login screen.
- If something is wrong it will give error message.

9.2.4.4 OTP screen component:

Identification	OTP Screen
Type	Class
Purpose	The OTP screen assure that only verified users can access the application
Function	There are some fields in the registration screen like email address, once user enter email address a code will send be send to verify there account after he/she clicks on Send OTP button.
Subordinates	The OTP screen contain the links to following screen: • Home Screen
Dependencies	The OTP Screen screen links to: Signup screen
Interfaces	The screen is designed to be easy to view using the resolution standard on mobiles. We tried to create simple and easy to use registration layout.
Resources	OTP function is used to send OTP to users.
Processing	Once user enter his/her email address, a 6 six digit code will be send to that address for verification process.
Data	The data supplied by the system are fields. User will enter email and click send button and code will be send. The internal data or data on screen is OTP Code which user will get.

9.2.5 Component 5 Offer Ride Screen:

9.2.5.1 Description:

The use of Offer Ride Screen in our application is that during the ride offering stage driver will post the details like source, destination, date&time so that users can avail that service.

9.2.5.2 Data Members:

It is a type of class which we used in our system through which driver enters details like source, destination, date&time and after offering the ride it will been showed to other users.

9.2.5.3 Methods:

- We will Offer Ride class and include offer ride function.
- Driver will enter his/her source, destination, date&time and click on offer ride.
- Function will generate ride and will been showed to other users to avail the ride.
- If something is inappropriate or incomplete it will give error message.

9.2.5.4 Offer Ride screen component:

Identification	Offer Ride Screen
Type	Class
Purpose	The offer ride screen main use is to offer ride to the user so that other can avail that service. Driver offer the ride for car sharing.
Function	There are some fields in this screen are like source, destination, date&time, once user enter all the details and click on submit the details will be saved in database and ride will be generated and on available rides screen the offer pool will be visible.
Subordinates	The Offer Ride screen links to: • Available rides screen • Home screen
Dependencies	The Offer Ride screen links to: • Home screen
Interfaces	The screen is designed to be easy to view using the resolution standard on mobiles. We tried to create simple and easy to use offer ride layout.
Resources	Database access requirements access is used to store all details.
Processing	The processing which is involve in our system is the inputting information into textboxes and storing that data. And a offer ride request will be generated.
Data	The data supplied by the system are fields. The data will be added by driver which will be accurate.

9.2.6 Component 6 Book Ride Verification:

9.2.6.1 Description:

The use of Book Ride Screen in our application is that during the ride booking stage users will search the rides available according to their requirement (their source and destination) and if the ride/ride detail is feasible so user can book that ride.

9.2.6.2 Data Members:

It is a type of class which we used in our system through which user will enters details like his/her source, destination and after searching the ride it will been show the user the available rides and if it match his requirements the user will book that ride.

9.2.6.3 Methods:

- We will add book Ride class and include book ride function.
- User will enter his/her source, destination and click on search ride.
- Function will generate list of available ride and it will been showed to user to avail the ride service and view all available rides and book according to their feasibility.
- If something is inappropriate or incomplete it will give error message.

9.2.6.4 Book Ride screen component:

Identification	Book Ride Screen
Type	Class
Purpose	The Book ride screen main use is to book a ride so that users can avail that service which other drivers are offering.
Function	There are some fields in this screen are like source, destination once user enter all the details and click on search ride the list of available rides will been showed from the database.
Subordinates	The Book ride screen links to: • Available rides screen • Home screen
Dependencies	The Book ride screen links to: □ Home screen
Interfaces	The screen is designed to be easy to view using the resolution standard on mobiles. We tried to create simple and easy to use book ride layout.
Resources	Database access requirements access is used to fetch all available rides.
Processing	The processing which is involve in our system is the inputting information into textboxes and searching the available rides by entering source and destination.
Data	For the data internal to the component is provided by system fields.

9.2.7 Component 7 Rating Screen:

9.2.7.1 Description:

The use of rating screen is to get the feedback from the user regarding experience with service.

9.2.7.2 Data Members:

It is a type of class which we used in our system to get the rating in stars form by the user and store it in our database.

9.2.7.3 Methods:

- We will add rating class and include rating function.
- User will enter on rating and a popup will be shown. User will select the stars and submit.
- If everything is provided it will be stored in database.

9.2.7.4 Rating screen component:

Identification	Rating Screen
Type	Class
Purpose	The Rating screen assures that user can give feedback/ratings about the services.
Function	The user will click on rating option and select the stars with the comment and click on submit button.
Subordinates	The Rating screen links to: Home screen
Dependencies	The Rating screen links to book ride screen only after completion of ride this screen will come.
Interfaces	The screen is designed to be easy to view using the resolution standard on mobiles. We tried to create simple and easy to use Rating layout.
Resources	Database access requirements access is used to store given rating into our system.
Processing	The only type of processing is required to select the star and click on submit
Data	The data supplied by the system are rating stars

9.2.8 Component 7 Available Rides Screen:

9.2.8.1 Description:

The use of Available Rides screen is to get the list of pool offer by the driver to the user so user can book ride according to his/her requirement if it is feasible.

9.2.8.2 Data Members:

It is a type of class which we used in our system to give list of available (ride/pool) so user can book any of them.

9.2.8.3 Methods:

- We will add Available ride class and include Available ride function.
- After User will details on booking screen he/she will be redirected towards Available ride screen.
- User will select any of the ride according to his/her need and will booked the ride.
- If everything is proper and user booked that ride it will be stored in database it will be stored in database.

9.2.8.4 Available Rides screen component:

Identification	Available ride Screen
Type	Class
Purpose	To Show all the Available Rides to user.
Function	All the nAvailable ride will be shown to the user in the list form. And then if he/she want to book they can book that (ride/pool).
Subordinates	The Available Ride screen links to: Main screen
Dependencies	The Available Ride screen links to: • Offer ride screen • Book ride screen
Interfaces	The screen is designed to be easy to view using the resolution standard on mobiles. We tried to create simple and easy to use Available ride layout.
Resources	Database access requirements access is used to fetch the rides available and also to store the booking if someone have booked the ride.

Processing	The only type of processing is required when any user book the ride from the available ride list.
Data	The data supplied by the system are ride booked details

9.2.9 Component 7 Available Rides Screen:

9.2.9.1 Description:

The use of the send notification screen in our application is that the driver can receive notifications related to ride bookings.

9.2.9.2 Data Members:

It's a type of class which we used in our system through which drivers can receive notifications.

9.2.9.3 Methods:

- We make send notification class and notification function.
- Drivers will receive notifications about rides if any user has booked a ride.
- Driver see notifications.
- Notification will include that new booing has been done.
- If something is wrong it will give error message.

9.2.9.4 Available Rides screen component:

Identification	Send Notification Screen
Type	Class
Purpose	To send notification to the driver and driver can see the notification.
Function	Driver will received notification if any new booking have been done for that ride.
Subordinates	The Available Ride screen links to: • Main screen • Login screen
Dependencies	The Available Ride screen links to: Main screen
Interfaces	The screen is designed to be easy to view using the resolution standard on mobiles. We tried to create simple and easy to use Available ride layout.
Resources	Not device tokens will be needed in start.
Processing	The only type of processing is required when any user book the ride from the available ride list then it will just notify the driver.
Data	The data supplied by the system are ride booked notify

10. User Interface Design:

10.1 Section Overview

In this section we will cover the rules of interface design, GUI Components, and Detailed Description.

- **In section 4.2:** we will introduce the interface design by describing the objectives of design rules.
- **In section 4.3**: In this section we will give an introduction to the GUI components.
- **In section 4.4:** It is a detailed description of user interface consisting the images of screen.

10.2 Interface Design Rules

- **Users accessibility**: In our system there are two main users one is Driver and other is User (the one who isn't the driver just a normal user) these users have own login systems and have the full access control to use the functions which are provided in their accounts.
- **Design and Consistency:** Restaurants Food waste app is very user friendly application and easy to understand for users. We have create very simple UI of our application by using flutter and its widgets. Each links and functions are properly labelled clearly so that user didn't face any kind of difficulty during using our application.

10.3 GUI Components

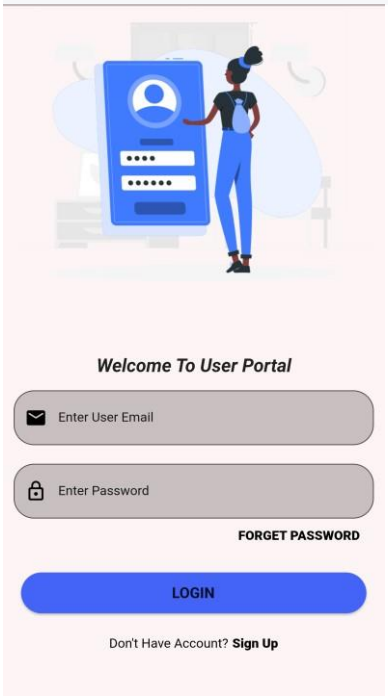
- Buttons
- text fields o icons o Text view o text span o app bar o Card Widget o Snack Bar o Flutter Toast o Splash screen o Carousel slider o Button navigation bar
- Images (like network or assets images)

10.4 Detailed Description

Provide a detailed description of the user interface including screen images. You may prefer to reference an appendix containing the screen snapshots.


GO CAR
SAFAR KARO ARAAM SE.

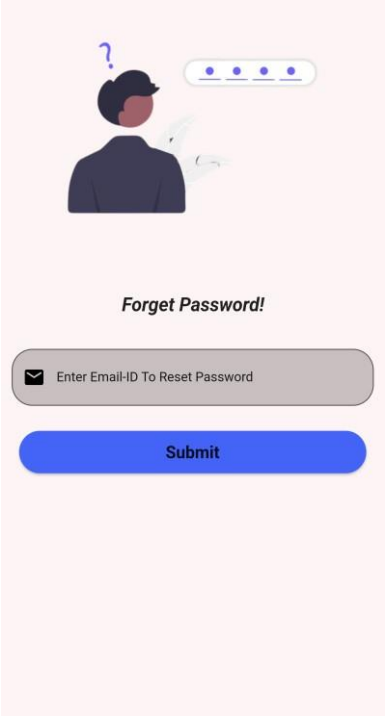

GO CAR
SAFAR KARO ARAAM SE.
LOGIN AS USER
OR
LOGIN AS DRIVER

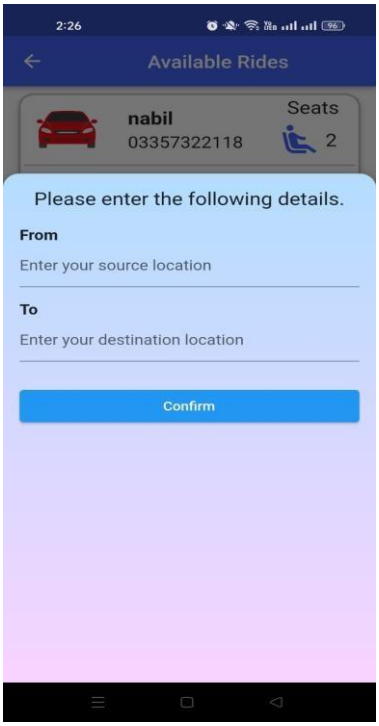

Welcome To User Portal
Enter User Email
Enter Password
FORGET PASSWORD
LOGIN
Don't Have Account? Sign Up

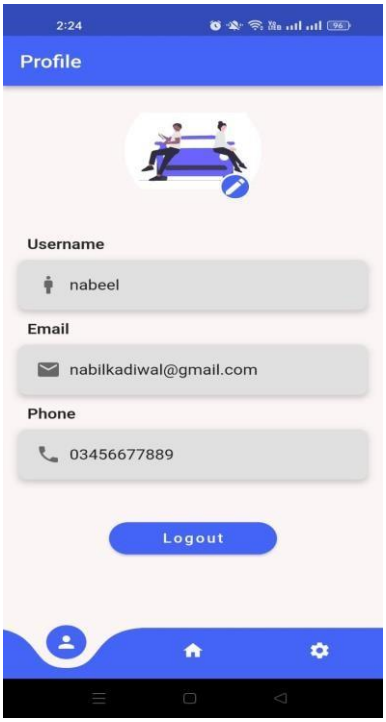

Welcome To Driver Portal
Enter User Email
Enter Password
FORGET PASSWORD
LOGIN
Don't Have Account? Sign Up

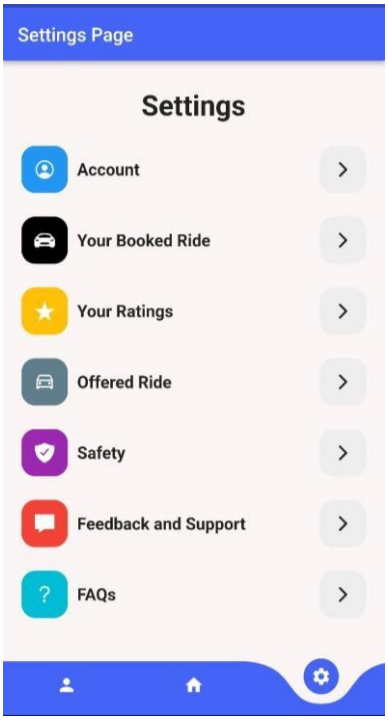

Please Enter User Details!
Enter Student ID
Enter user Email
Enter Password
Enter Phone Number
SIGN UP
Already Have An Account? Login

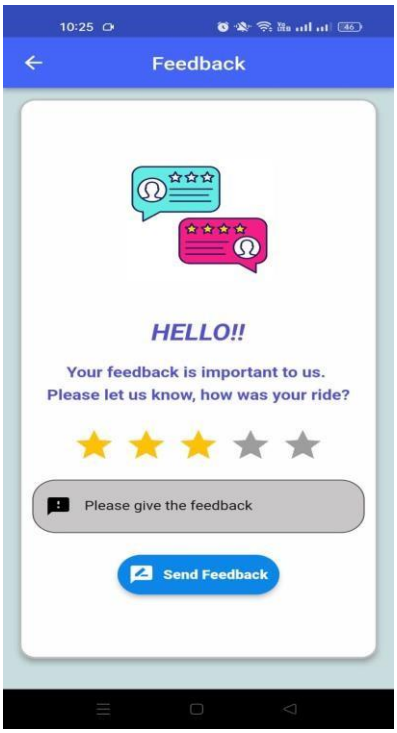

Please Enter Driver Details!
Enter Student ID
Enter Email-ID
Enter Password
Enter Phone Number
Enter Car Name
Enter Car Number
Enter Car Color
SIGN UP
Already Have An Account? Login

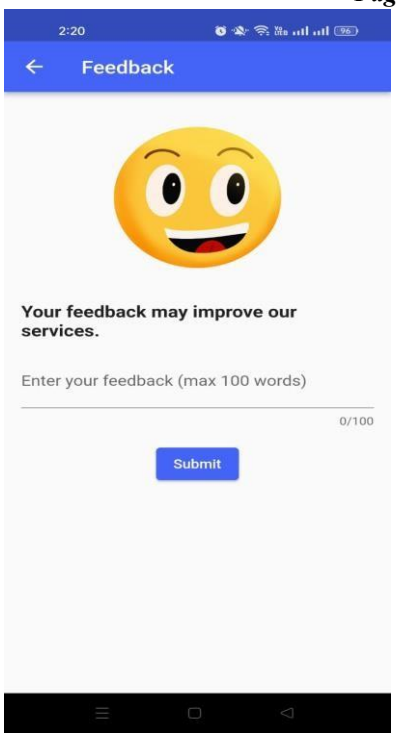

Forget Password!
Enter Email-ID To Reset Password
Submit

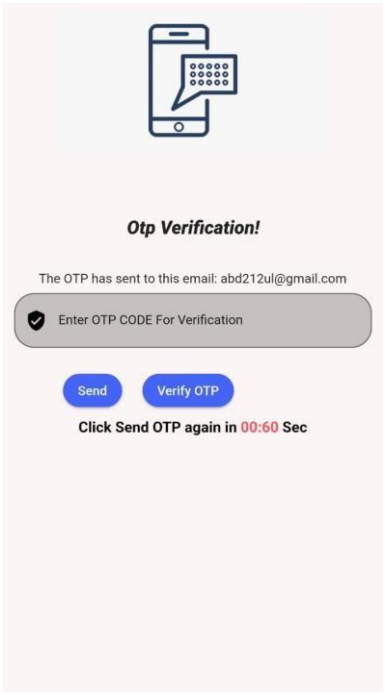

2:26
Available Rides
nabil
03357322118
Seats
2
Please enter the following details.
From
Enter your source location
To
Enter your destination location
Confirm

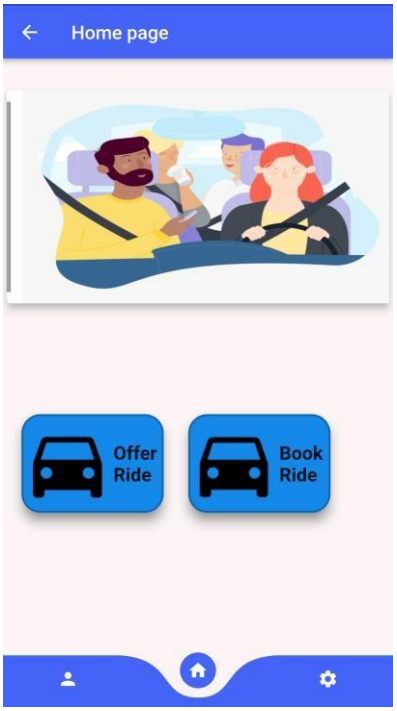

2:24
Profile
Username
nabeel
Email
nabilkadiwal@gmail.com
Phone
03456677889
Logout











11. Reuse and relationships to other products:

To avoid spending time doing someone else's work twice, we set out to make any existing code usable from the start. We can employ open source technologies because we are developing

applications for specific purposes. First, since Dart is free and enables us to run our application on both Android and IOS, we chose to implement our code in it. Because it is open source and offers a wide variety of plug-ins, Android Studio was chosen because it enables the usage of pre-existing applications. Additionally, we utilize Firebase for databases because it has a fantastic user interface, is free to use, and has numerous capabilities that enhance the user experience for clients.

12. Design decisions and trade offs

The design decision to make the application more efficient, app was made to help users to get benefits of carpooling. Modularity provides encapsulation for the important pieces of the system. Using encapsulation, we are able to change important parts of the system without changing the whole system. We design our application which will be easy to use and satisfy user needs.

13. Pseudo code for components

1. User registration:

- Enter details for registration
- If (Email == correct form && is not empty)
- Else If (user name == correct form && is not empty))
- Else If (phone number == correct form && is not empty))
- User registered successfully
- Else
- Show Error Message
- Redirect to Registration Page.

2. Driver registration:

- Enter details for registration
- If (Email == correct form && is not empty)
- Else If (user name == correct form && is not empty))
- Else If (phone number == correct form && is not empty))
- Else If (car number == correct form && is not empty))
- Else If (car model == correct form && is not empty))
- Else If (car color == correct form && is not empty))
- User registered successfully
- Else
- Show Error Message
- Redirect to Registration Page.

3. Login:
- Enter your username and password;
- If (Username and Password == Valid))
- Redirect to Dashboard.
- Else
- {Show Error message}
- Redirect to login page.

4. Forgot Password:
- Enter email to reset password
- If (Email == valid)
- Send mail to that email in which reset link is there.
- Else
- {Show Error Message}
- Redirect to forget page.

5. Rating and Review:
- Login
- After ride completion
- Click on ratings
- Mark the stars
- Click on submit
- Rating will be stored in database

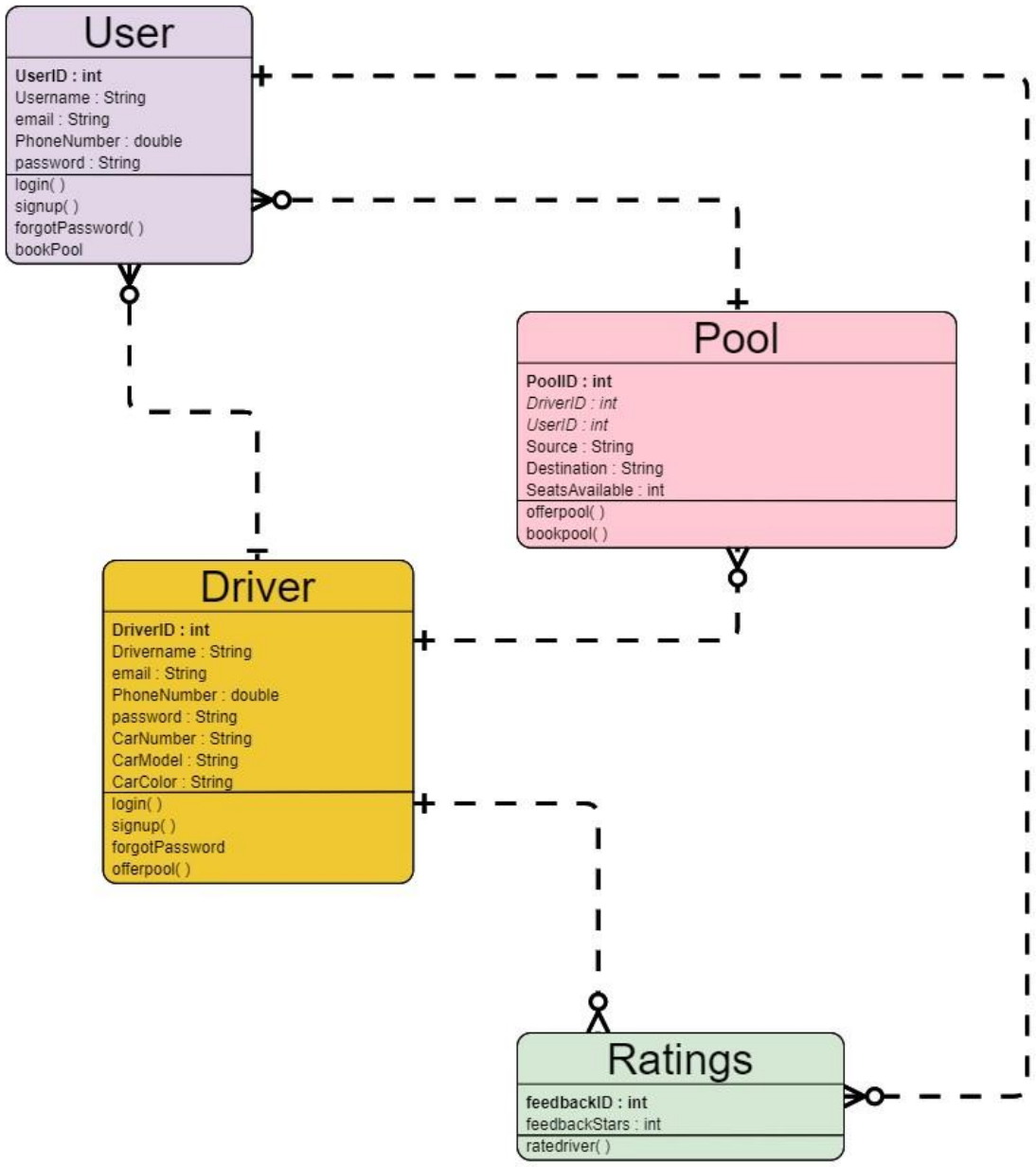
6. Book rides:
- If (user == Login)
- You will able to see list of rides/pool available.
- Select the desire ride you want after clicking on “BOOK NOW” Button.
- Entering all the details and clicking on confirm button.
- If successful your ride has been book

7. Available rides:
- If (user == Login)
- After entering details and clicking on search button
- You will able to see list of rides/pool available in this screen.
- If offer ride is feasible then book that ride
- Your booking details will be stored in database.

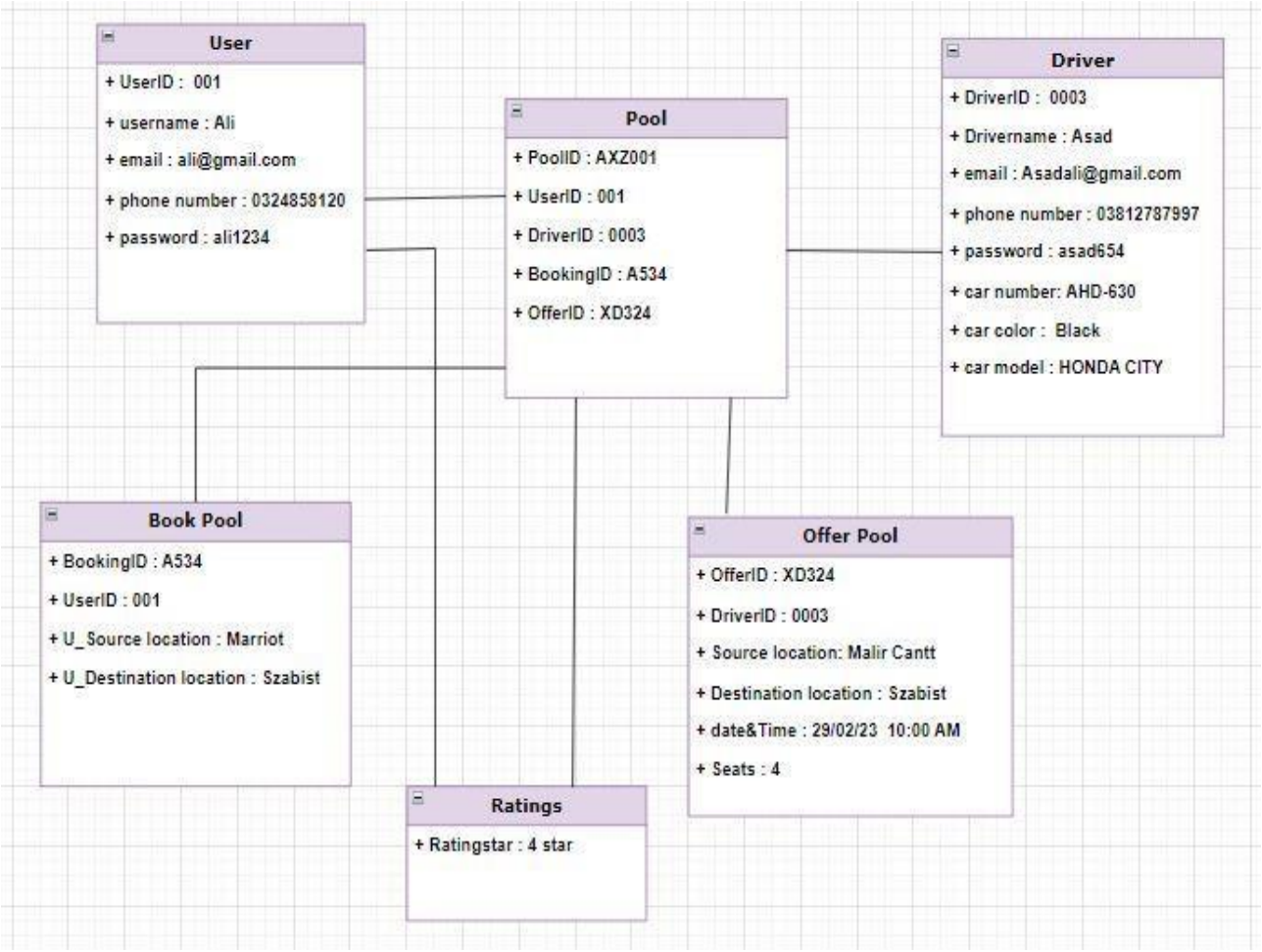
8. Book rides:
- If (user == Login)
- Entering all the details
- If (enter details == correct form && is not empty)
- You will able to offer pool to users
- Else
- {Show Error message “unable to offer the ride”}

14. Appendices

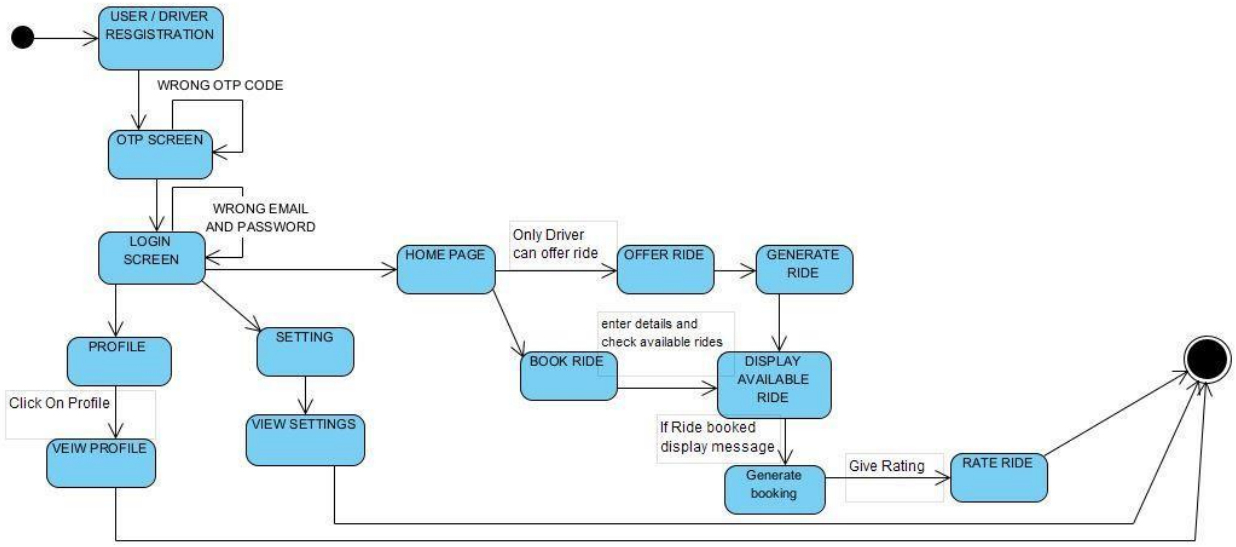
14.1 Class Diagram

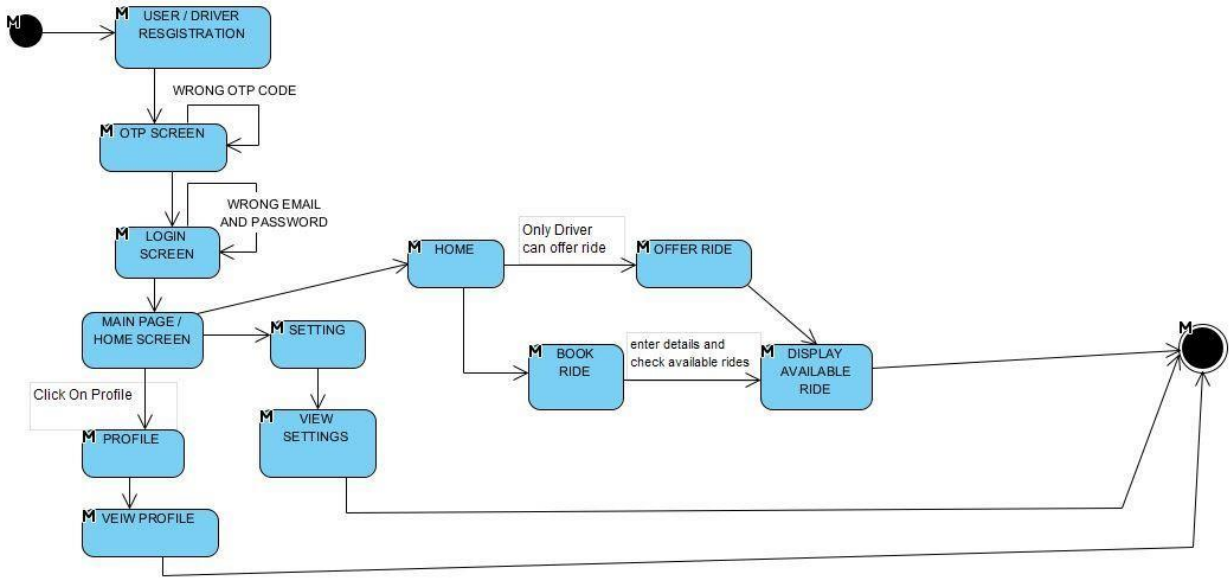


14.2 Object Diagram:



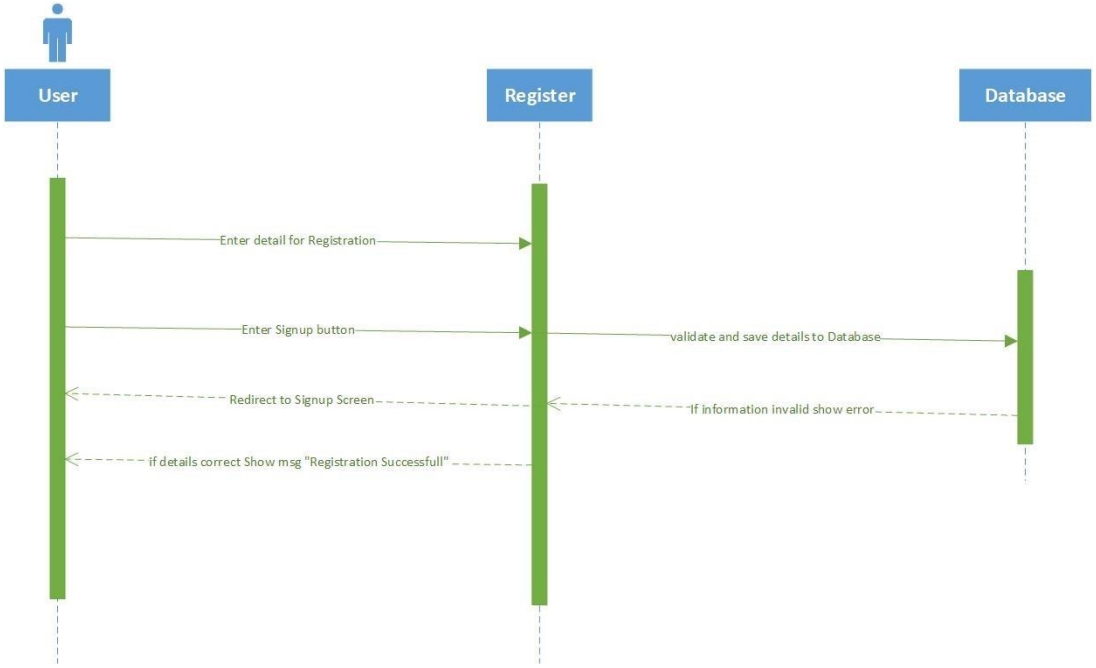
14.3 State Chart Diagram:



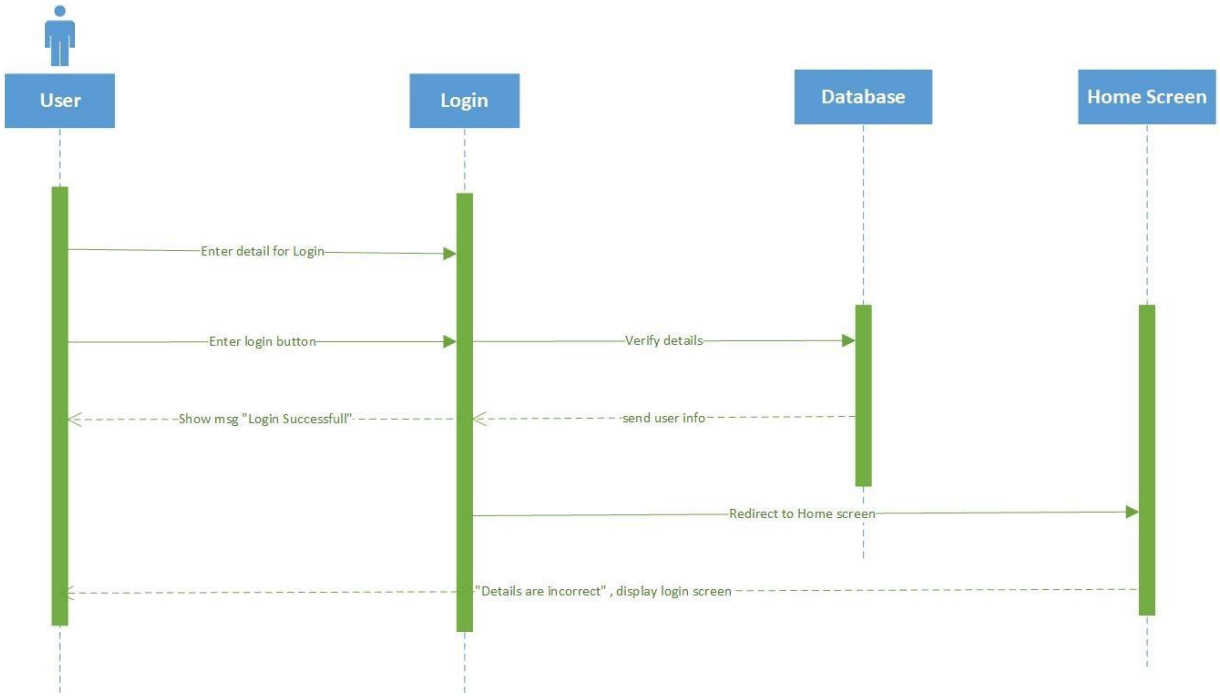


14.4 Sequence diagram:

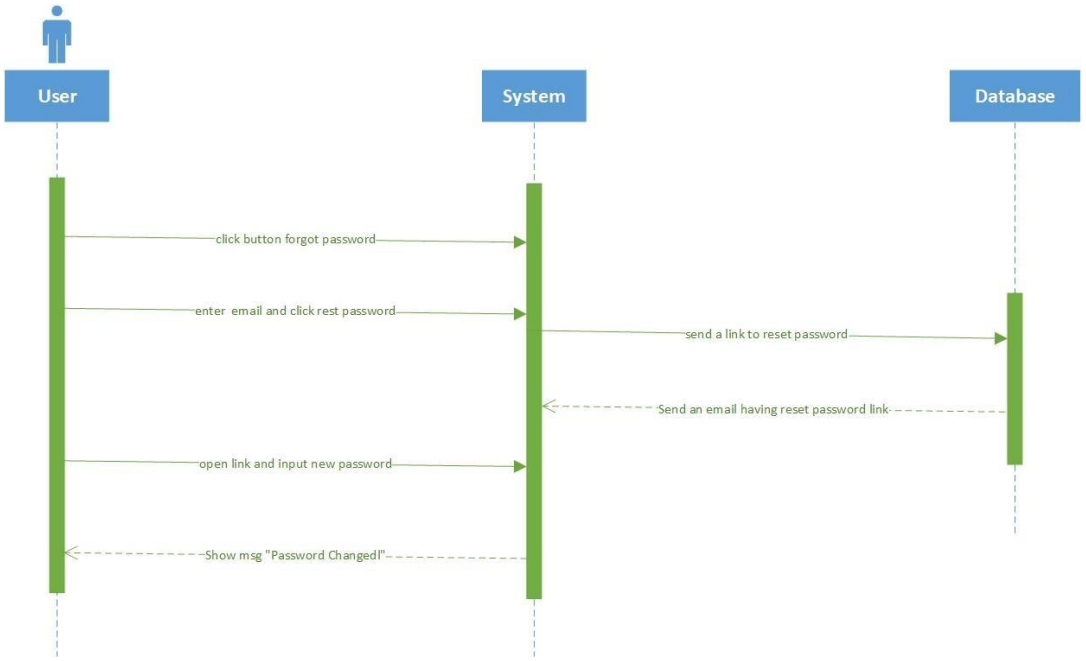
14.4.1 Registration:



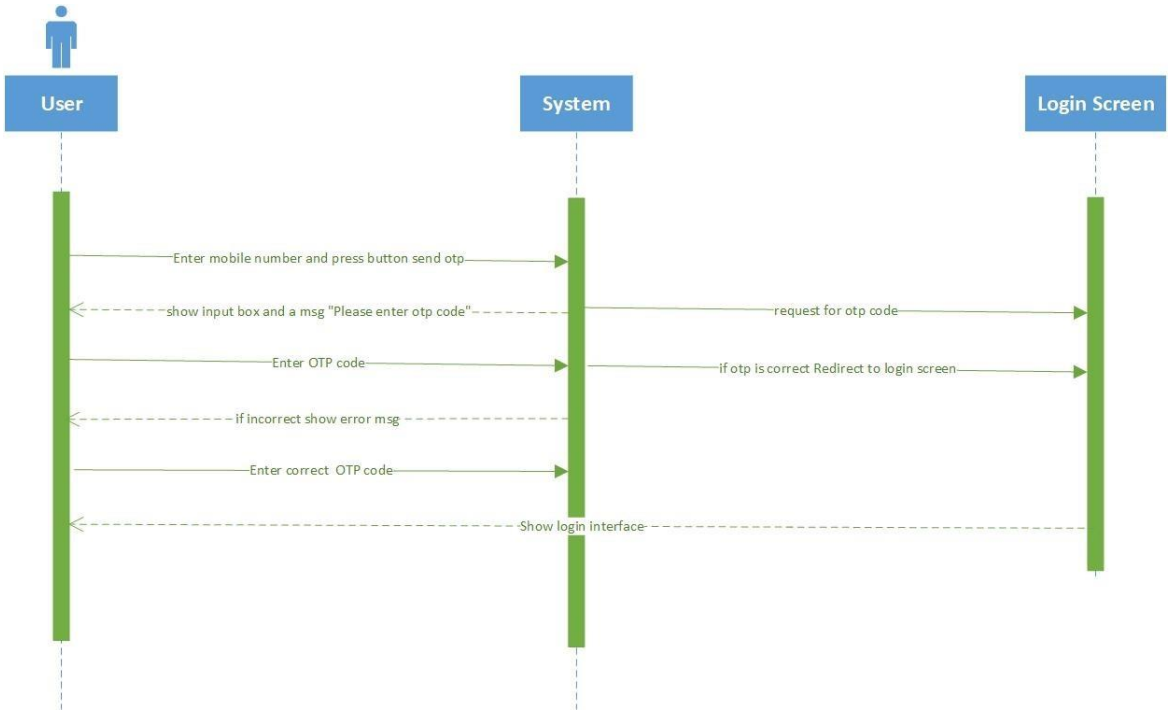
14.4.2 Login:



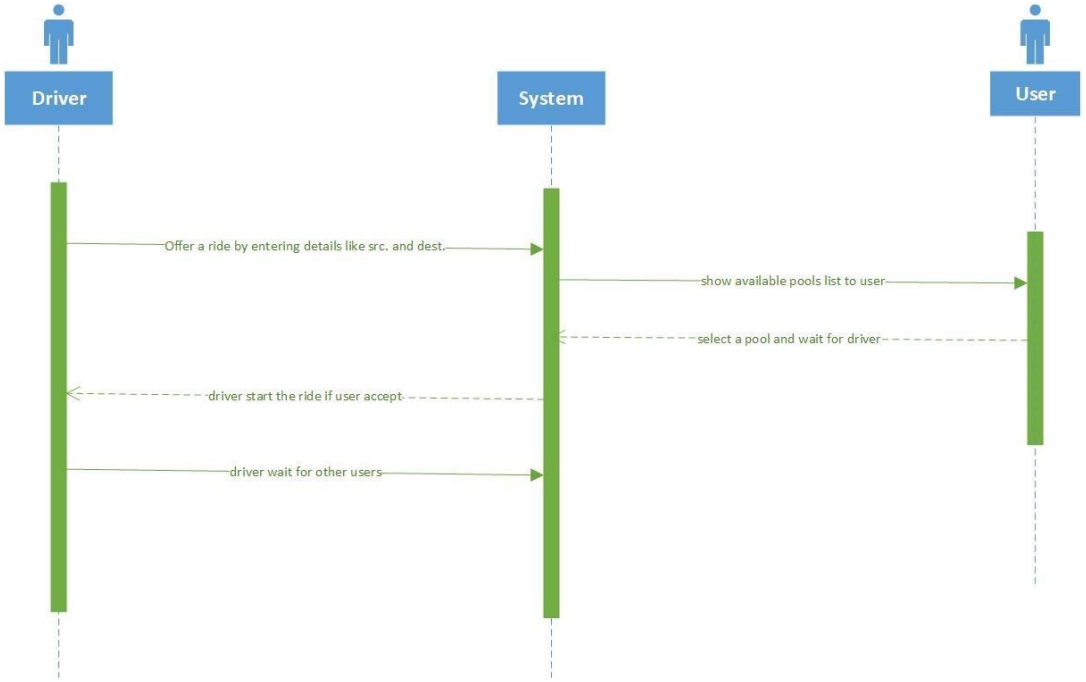
14.4.3 Forgot password:



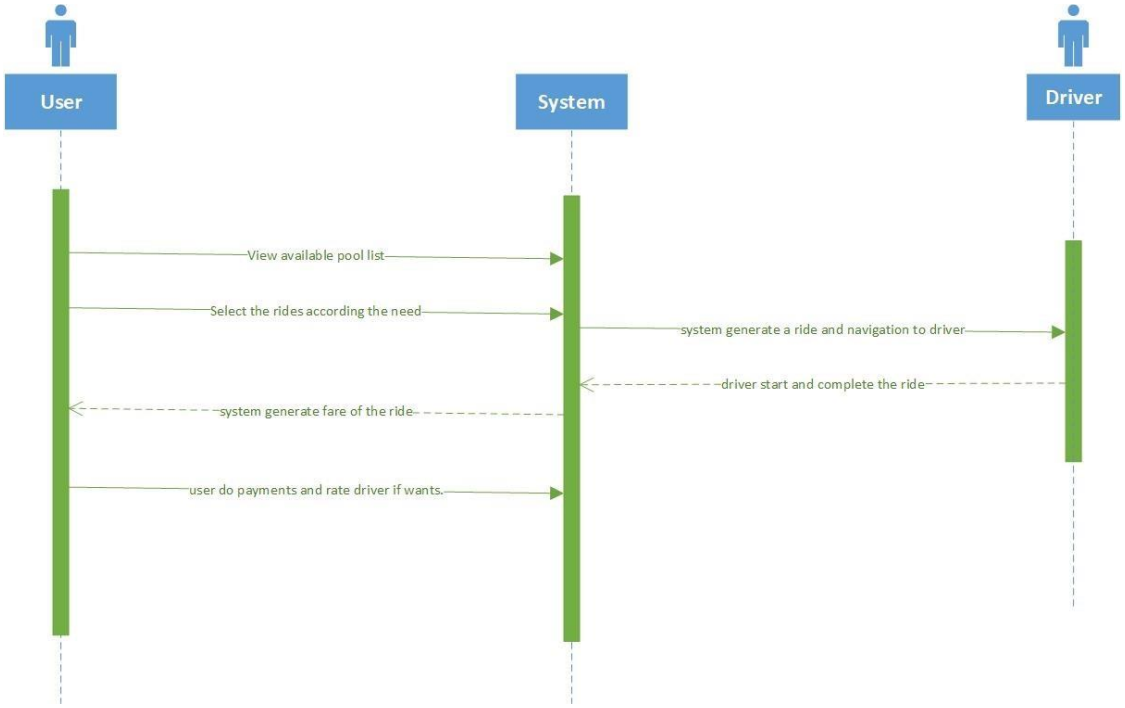
14.4.4 OTP verification:



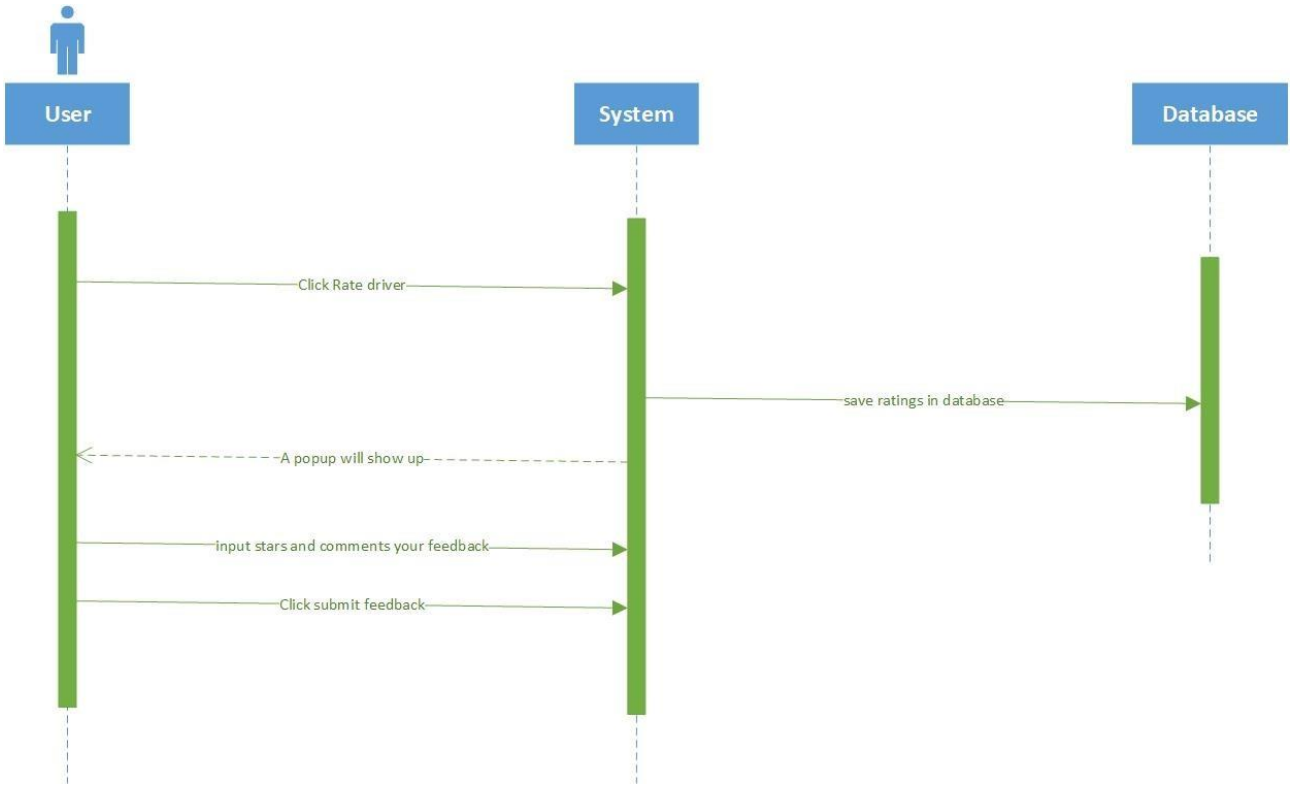
14.4.5 Offer Ride:



14.4.6 Book Ride:

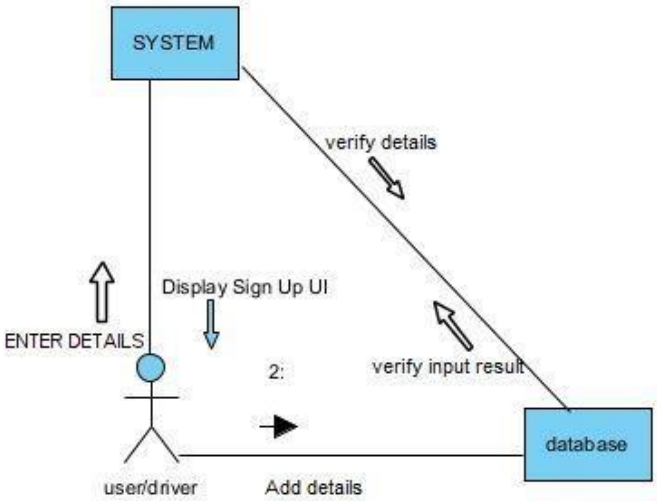


14.4.7 Ratings:

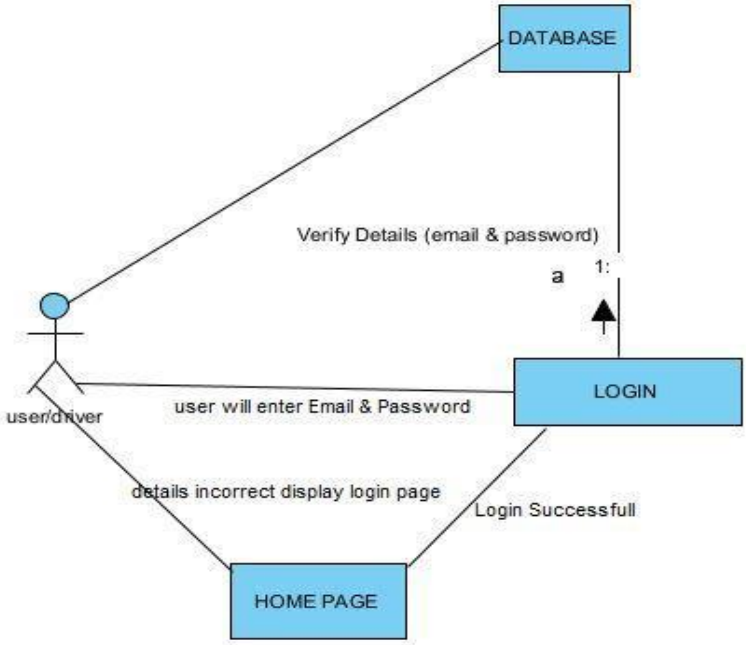


14.5 Collaboration Diagram:

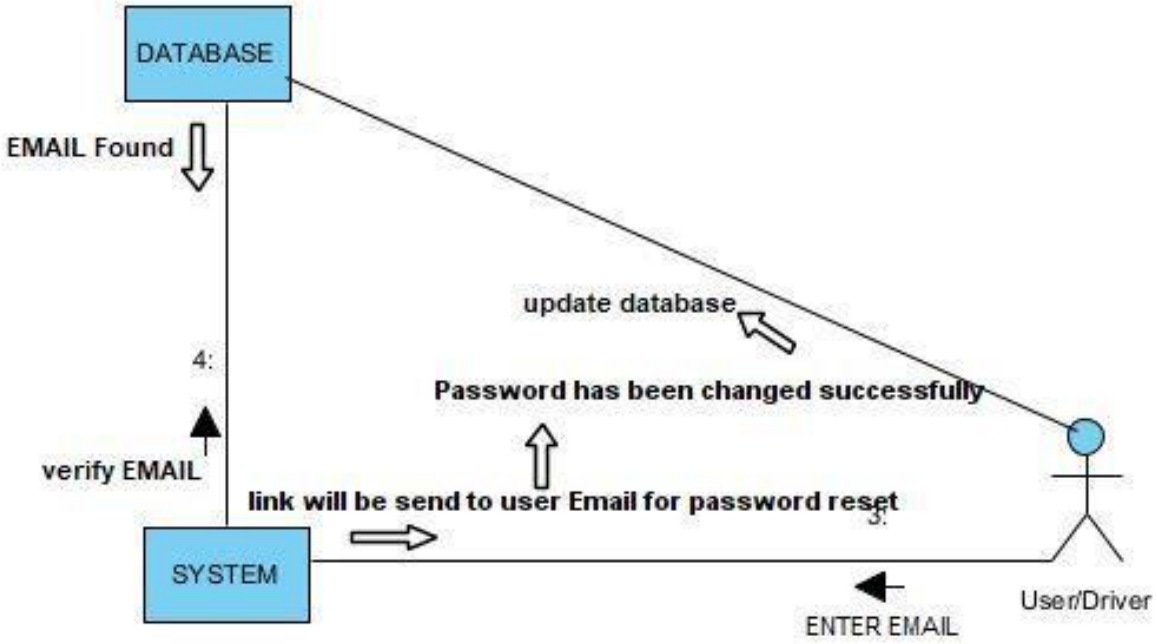
14.5.1 Registration:



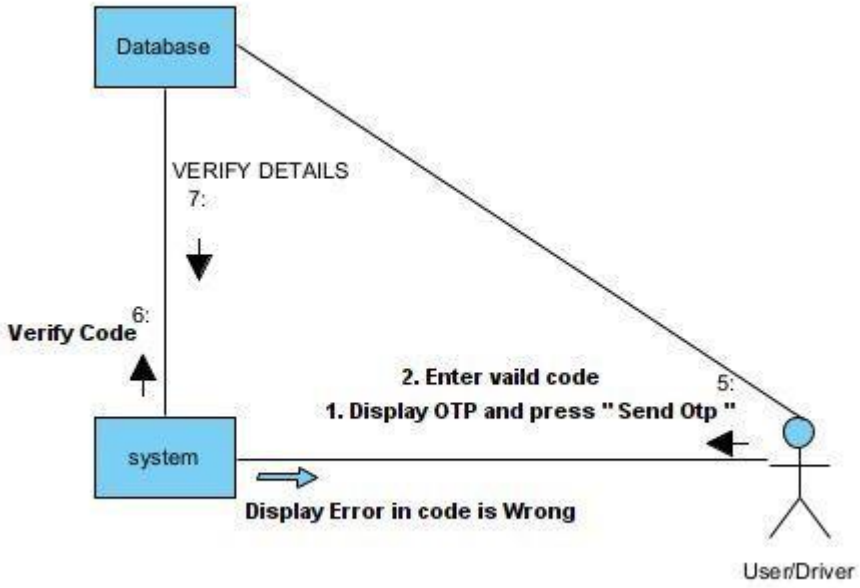
14.5.2 Login:



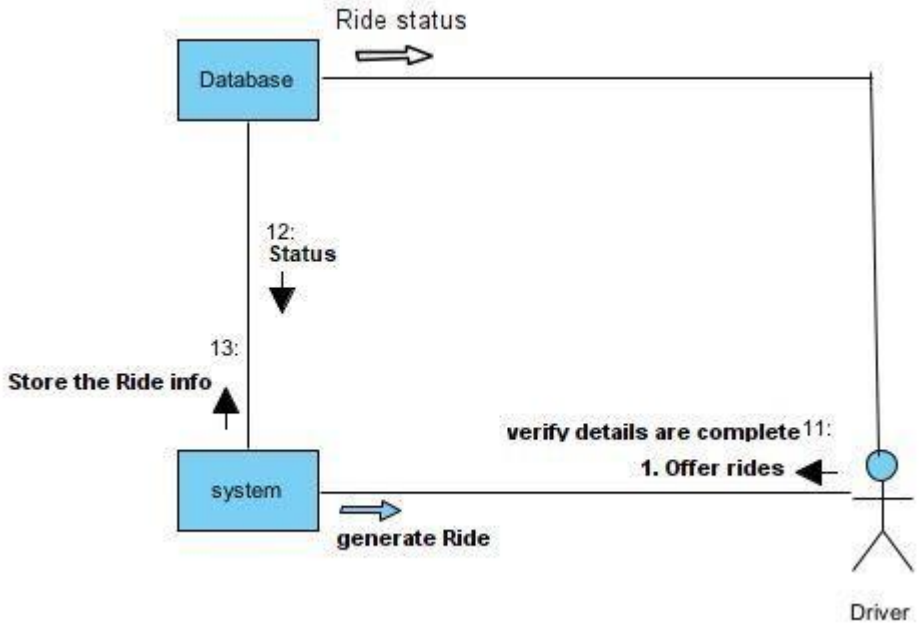
14.5.3 Forgot password:



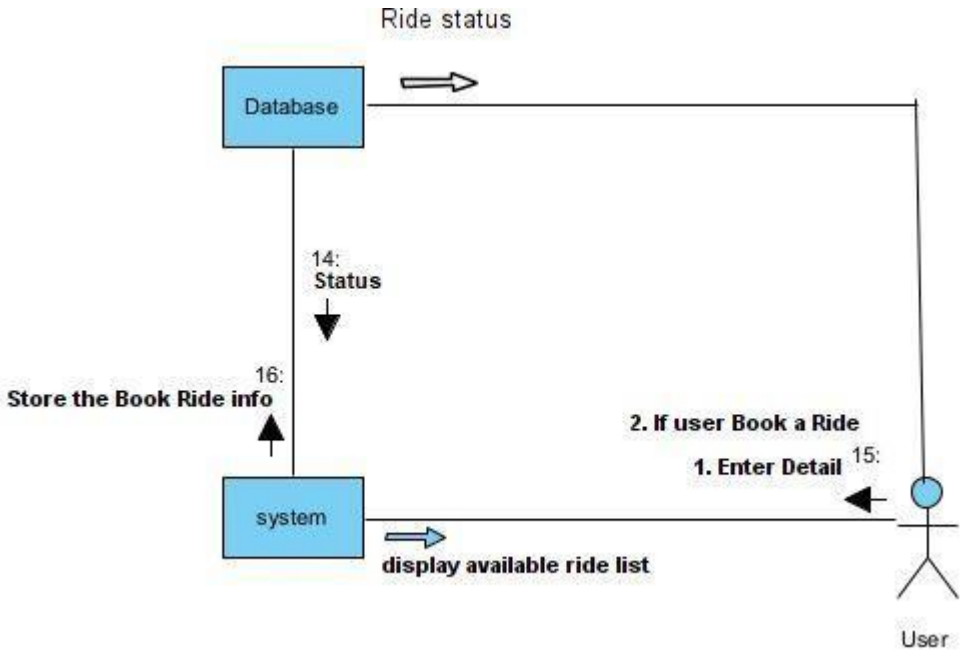
14.5.4 OTP verification:



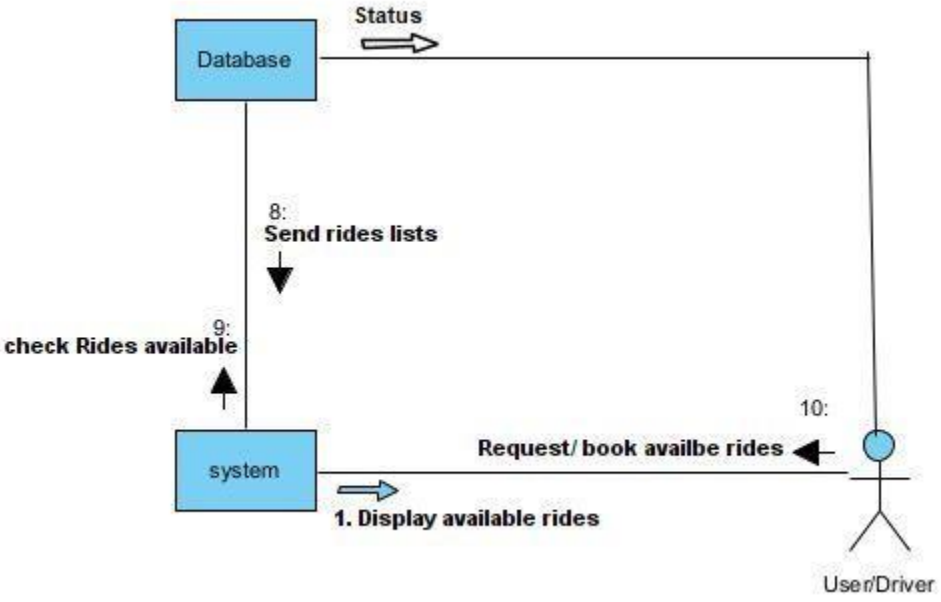
14.5.5 Offer Ride:



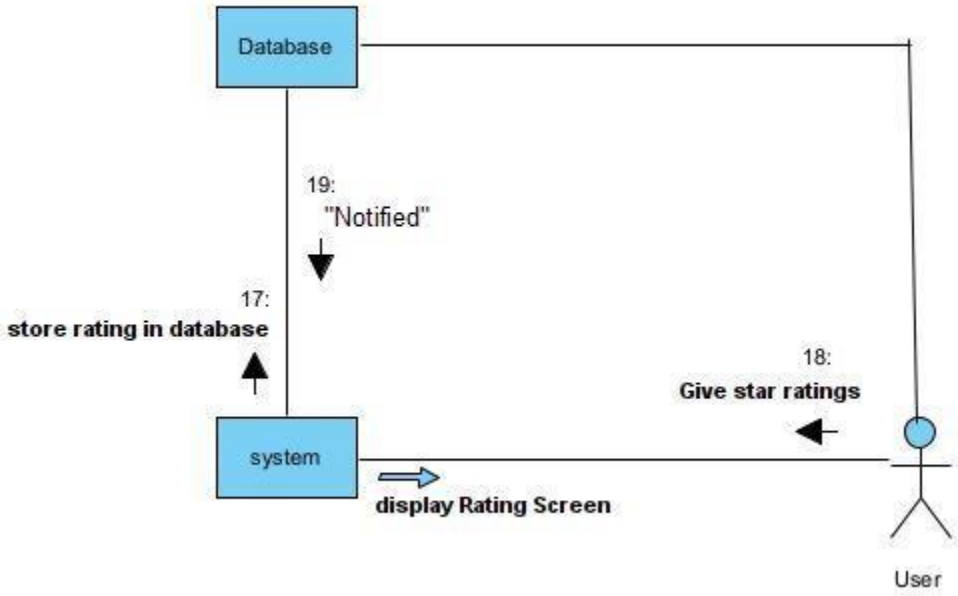
14.5.6 Book Ride:



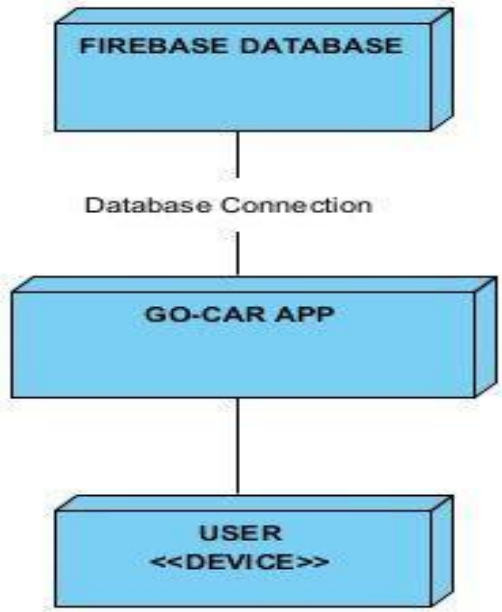
14.5.7 Available Ride:



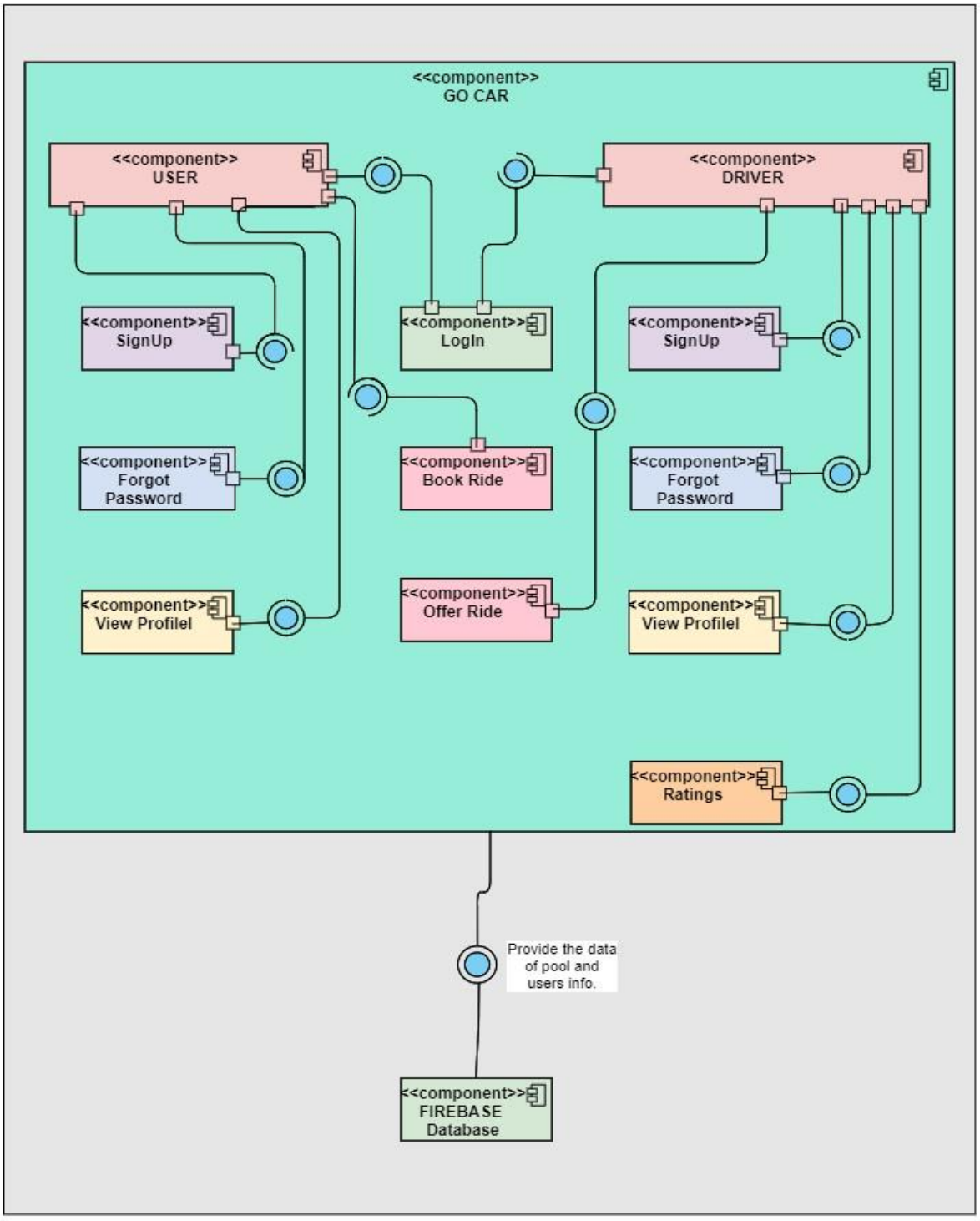
14.5.8 Ratings:



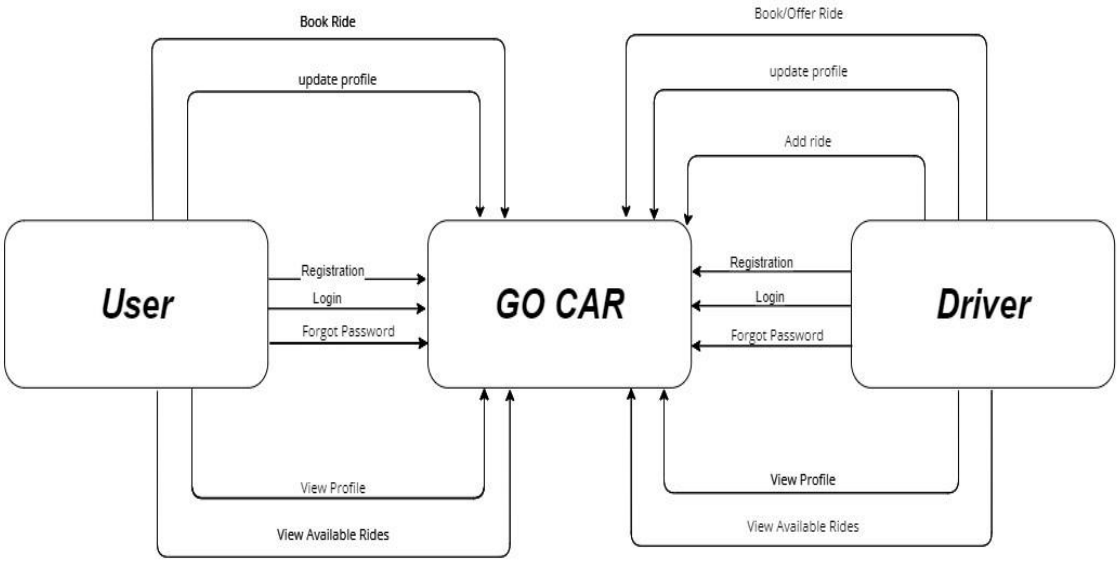
14.6 Deployment diagram:



14.7 Component Diagram:

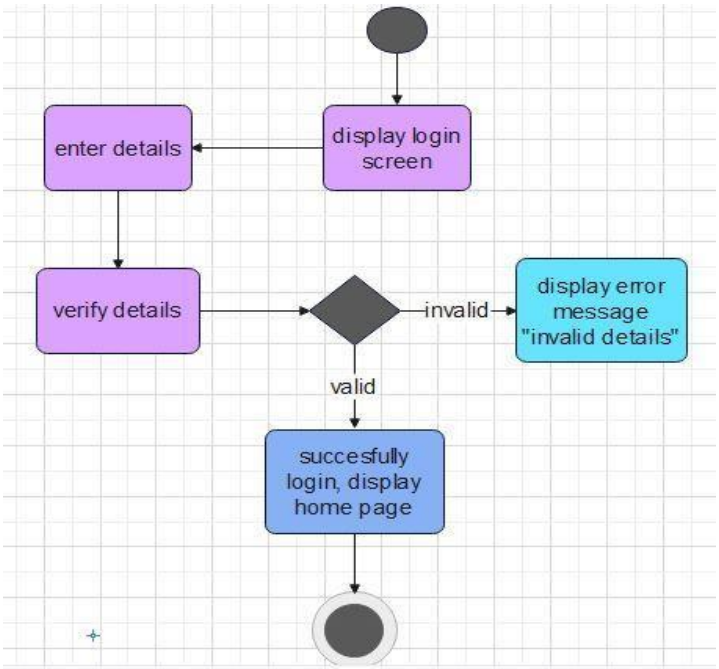


14.8 System Block Diagram:

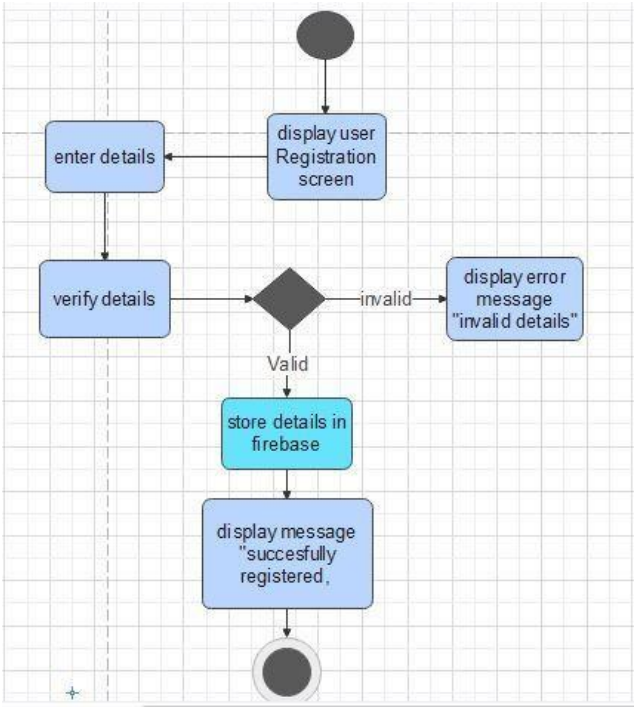


14.9 Activity Diagram:

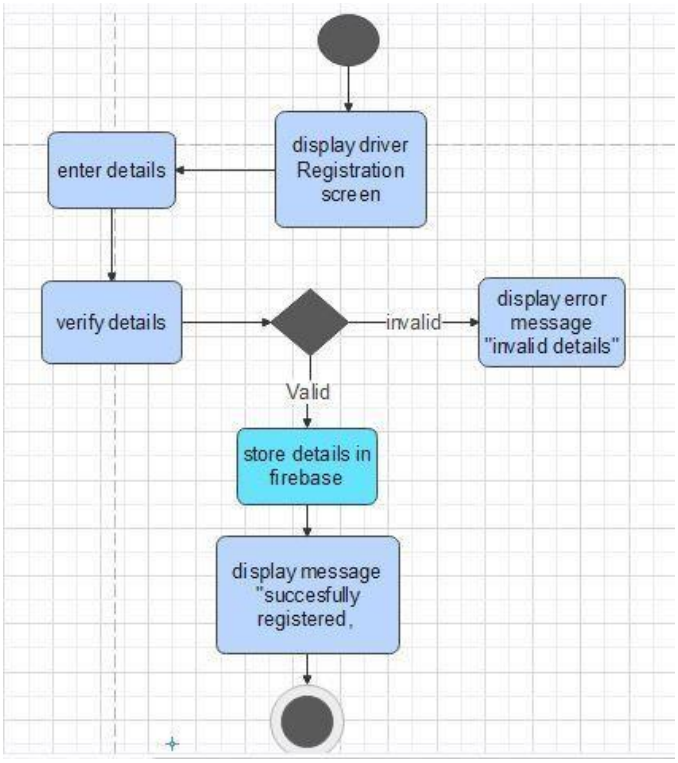
14.9.1 Login:



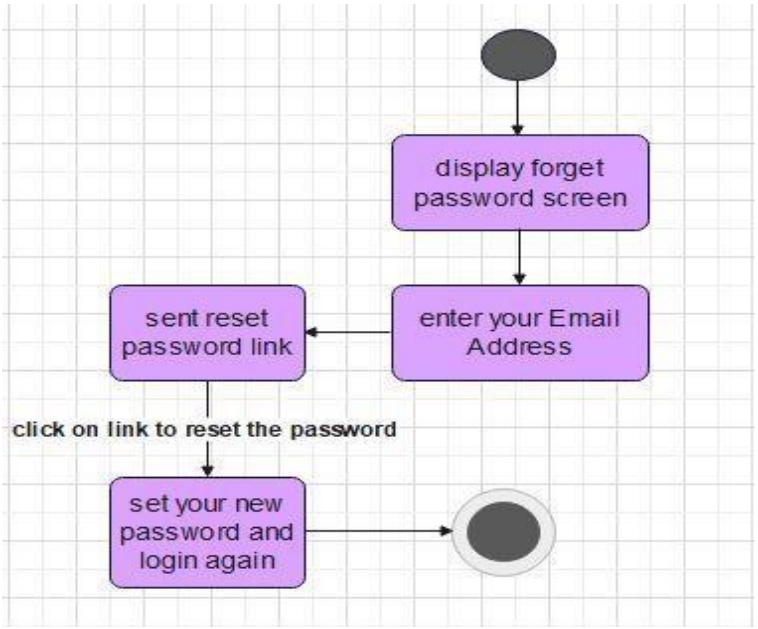
14.9.2 User Registration:



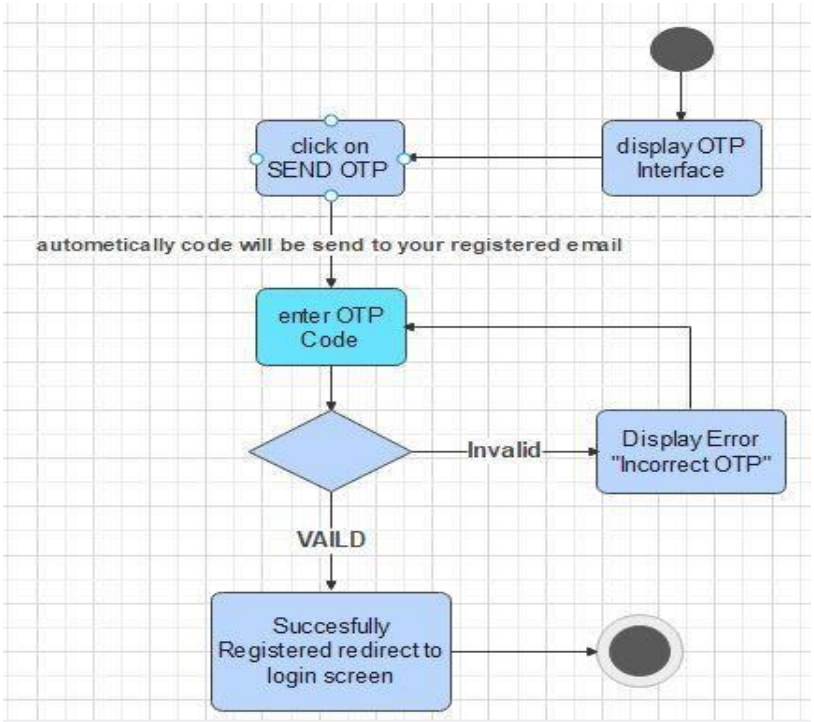
14.9.3 Driver Registration:



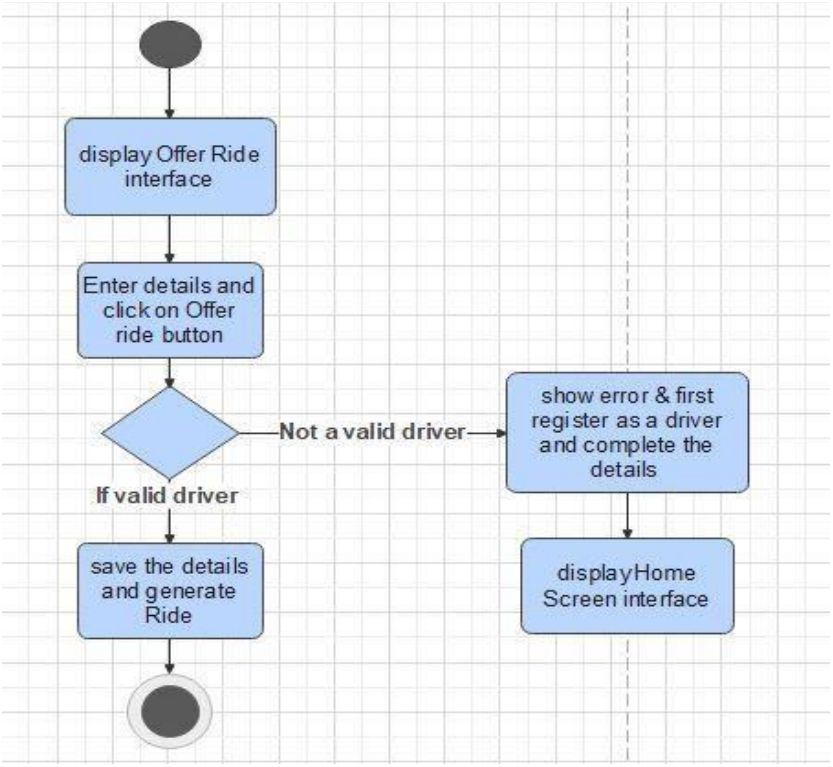
14.9.4 Forgot password:



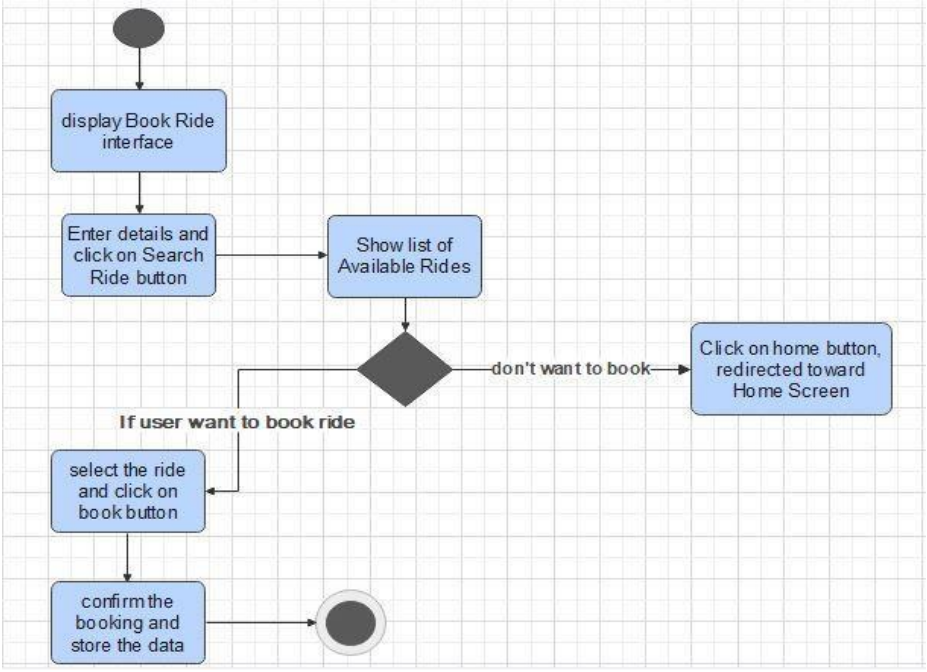
14.9.5 OTP verification:



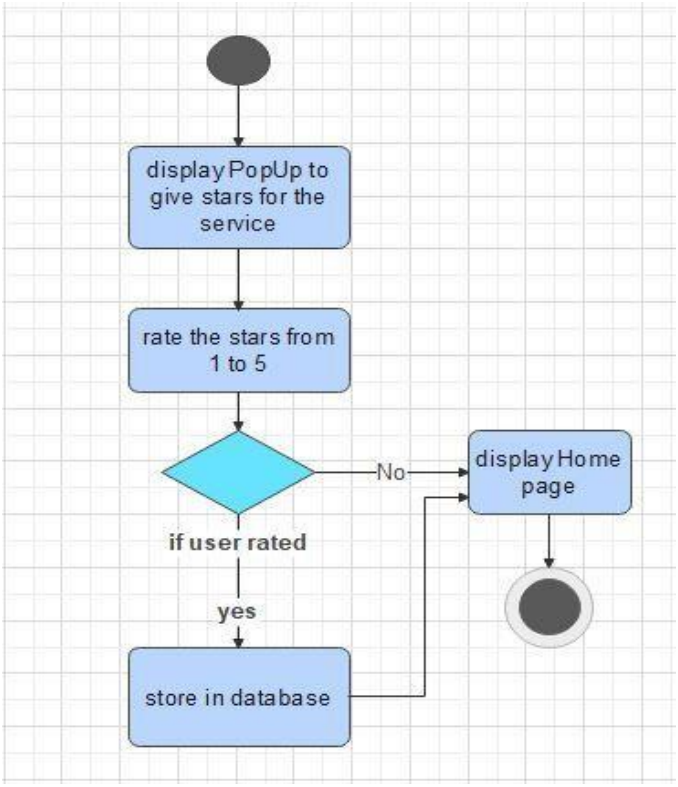
14.9.6 Offer Ride:



14.9.7 Book Ride:



14.9.8 Ratings:



User Manual:

15. General Information

15.1 Introduction

The “Go-Car” is an application, which acts like a carpooling service provider between user/customer and driver. It will provide a platform where user can use our service to book a ride/seat for carpooling for ease of transportation in today’s time.

15.2 Project Scope

Our project aim is to build an efficient application to give people interactive interface platform where they can avail carpooling service and to solve their expense problem through our service and feature. This project will be built on Flutter-Dart and Firebase.

15.3 System Overview:

We created the friendly user interface with which any user can interact with in a good manner. In our system there are two main users one is Driver and other is User (the one who isn’t the driver just a normal user) these users have own login systems and have the full access control to use the functions which are provided in their accounts.

15.4 User Access level:

Users have to create an account at the first time in order to use app, book or offer pool and operate other functions.

15.5 Contingencies

In any case of attack by third party user must be careful about the application version. Means if he is using older version which may have vulnerabilities. User must be up to date with the application to avoid any disaster. It can be done by play store auto update when a user connects to WIFI.

15.6 Basic Functionality:

Our app core features are book pool and offer pool. User can find a pool for available list and can book the ride according to his need. User can enter details of his/her source and

destination and then perform booking process. Also, a driver should offer a ride in order to start a ride. They both can access these features in their respective home screen.

15.7 Contact and Support:

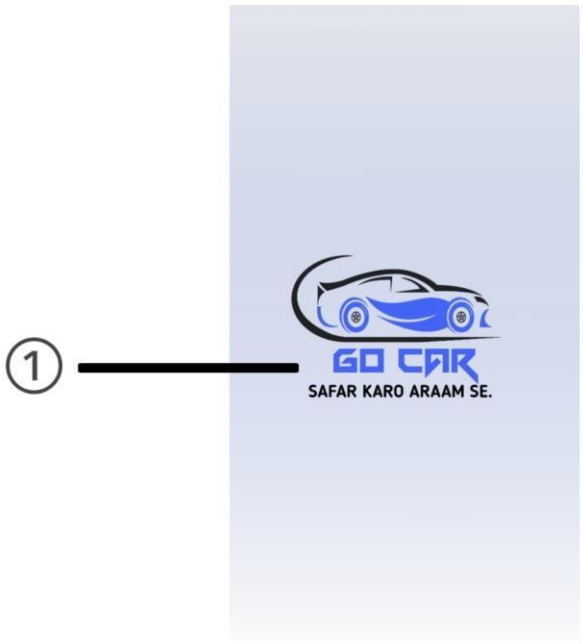
If you have any queries, you can reach us by email, phone or in-app feedback.

Phone: 03172404046

Email: aligocar04@gmail.com

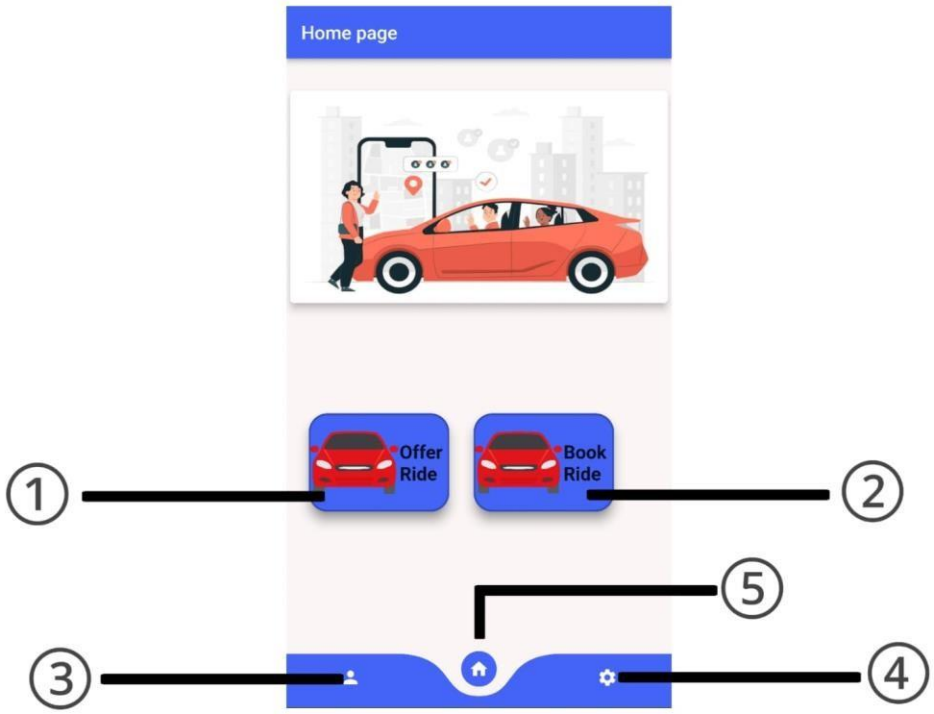
16. System feature

16.1 Splash Screen:



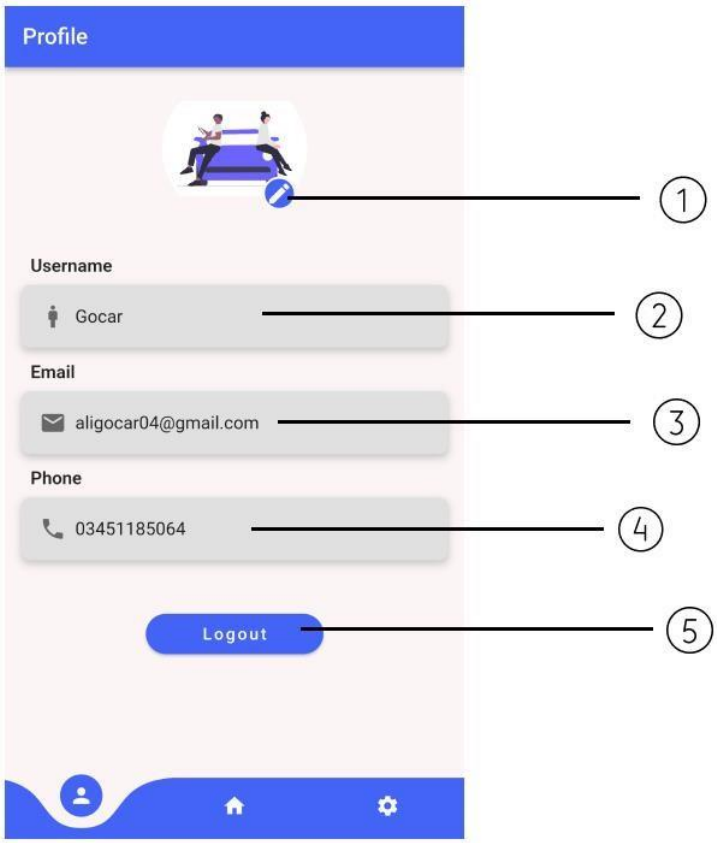
1) User or Driver will see splash screen at very first time.

16.2 Home screen:



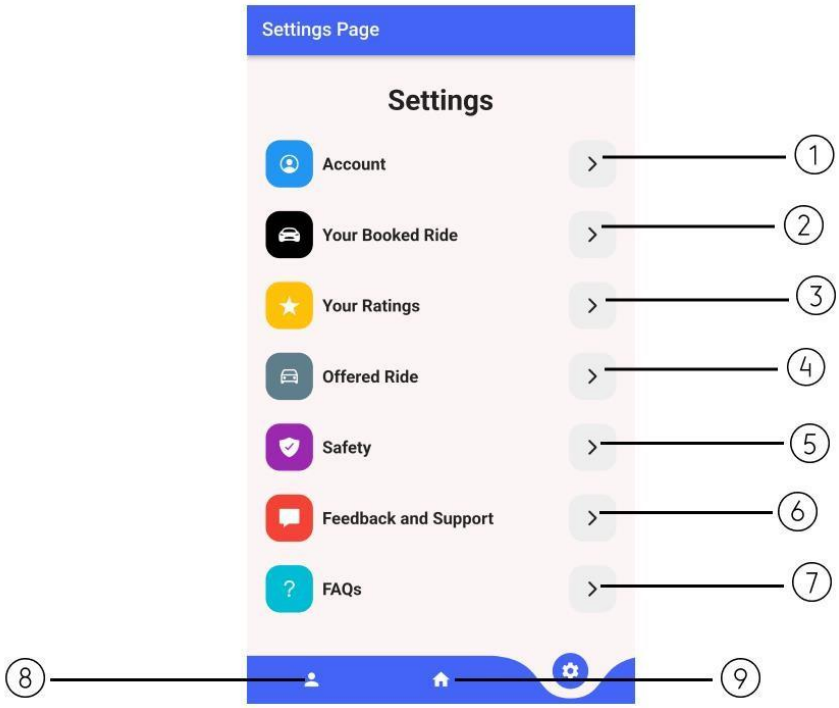
1) Driver clicks the button to offer ride.
2) User clicks the button to Book ride.
3) Click on button to view profile info.
4) Click on button for setting.
5) Click on button to view home screen.

16.3 Profile Screen:



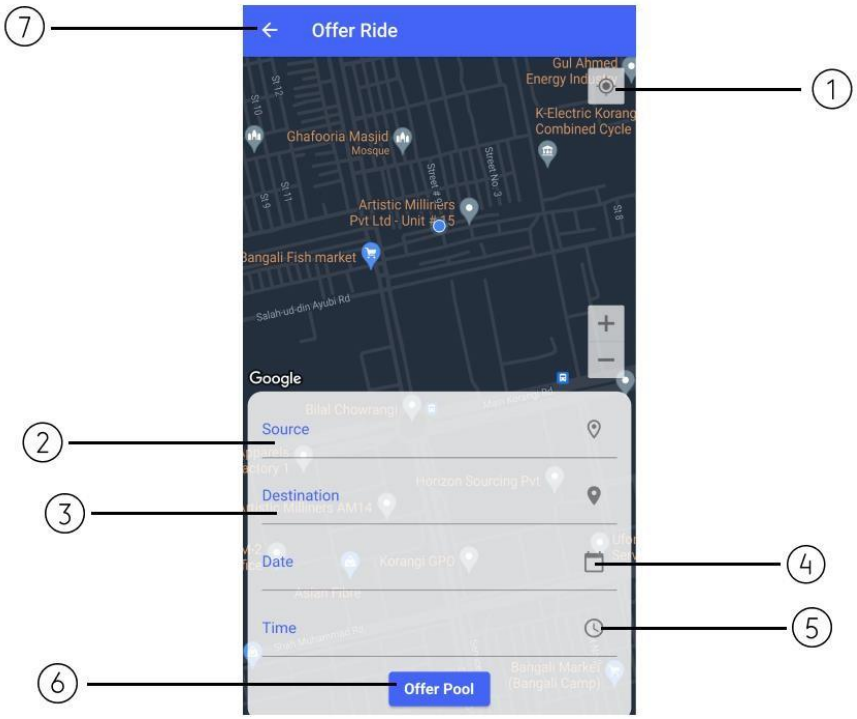
1) Click button to change photo.
2) User / driver name.
3) User / driver email.
4) User / driver phone number.
5) Click button to logout the id.

16.4 Driver Settings:



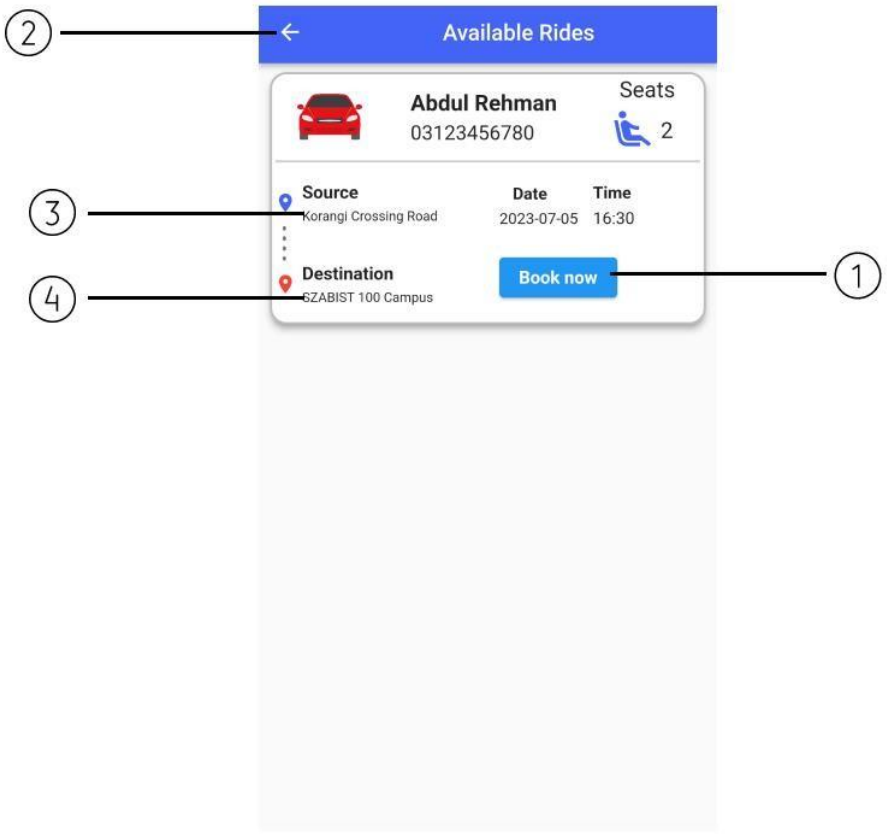
1) Click button to see account logged in.
2) Click button to see booked ride. 3) Click button to see ratings.
4) Click button to see Offered ride.
5) Click button to see safety.
6) Click button to give feedback.
7) Click button to see FAQs.
8) Click button to see profile information.
9) Click button to see home screen.

16.5 Offer Ride/Pool:



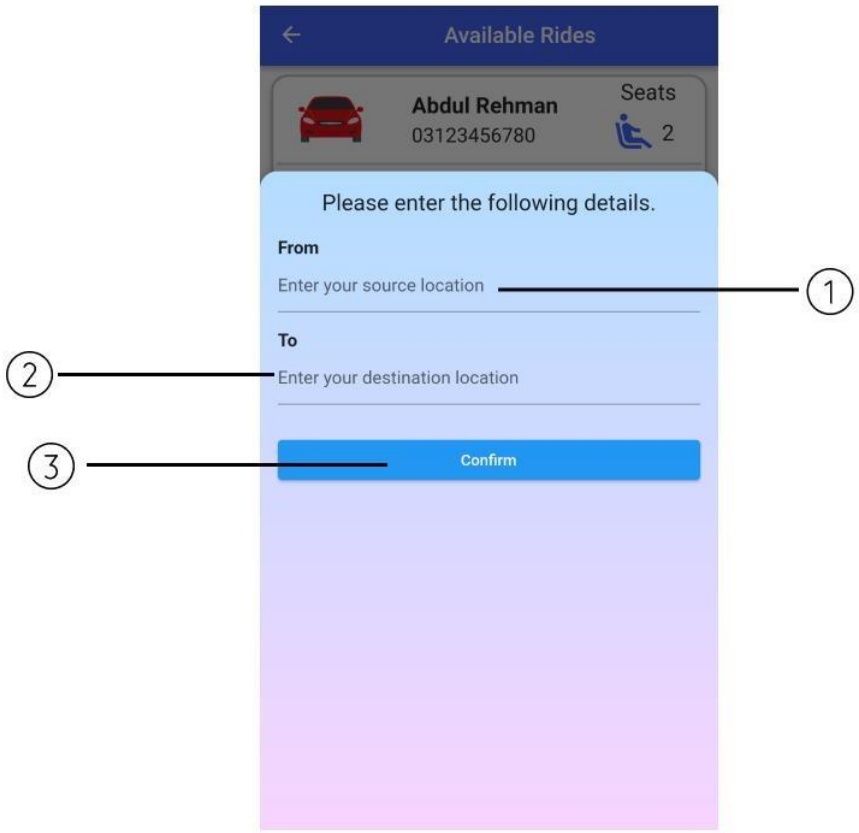
1) Click button to get current location.
2) Click to enter source location.
3) Click to enter destination location.
4) Click to select calendar.
5) Click to select time.
6) Click button to offer a pool.
7) Click button to go to previous screen.

16.6 Book ride/ pool:



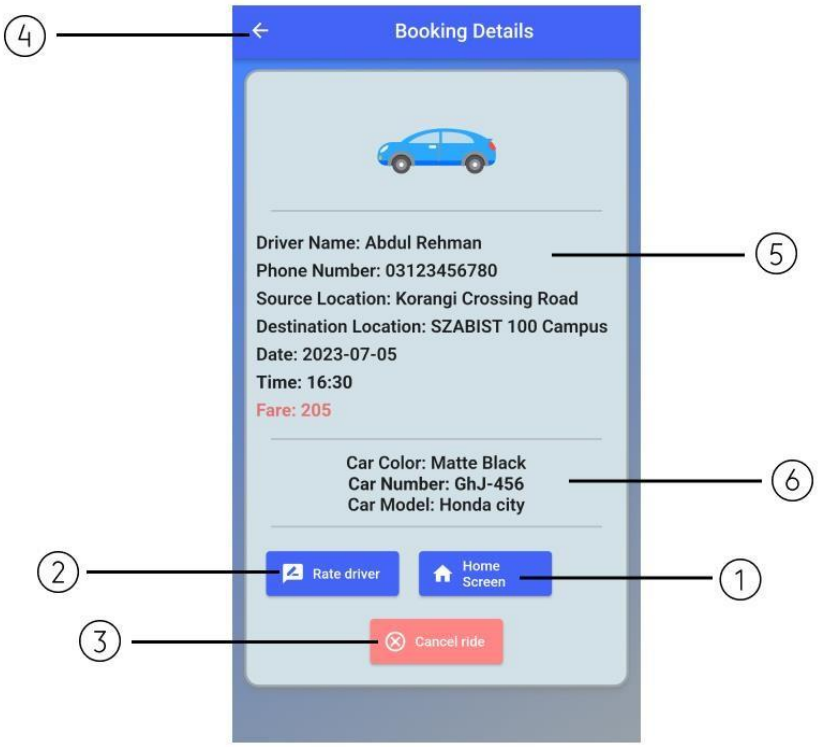
1) Clicks the button to Book ride 2)
Click button to go to previous screen. 3)
Users can view the source location.
4) Users can view the destination location.

16.7 User Enters Pre-Booking details:



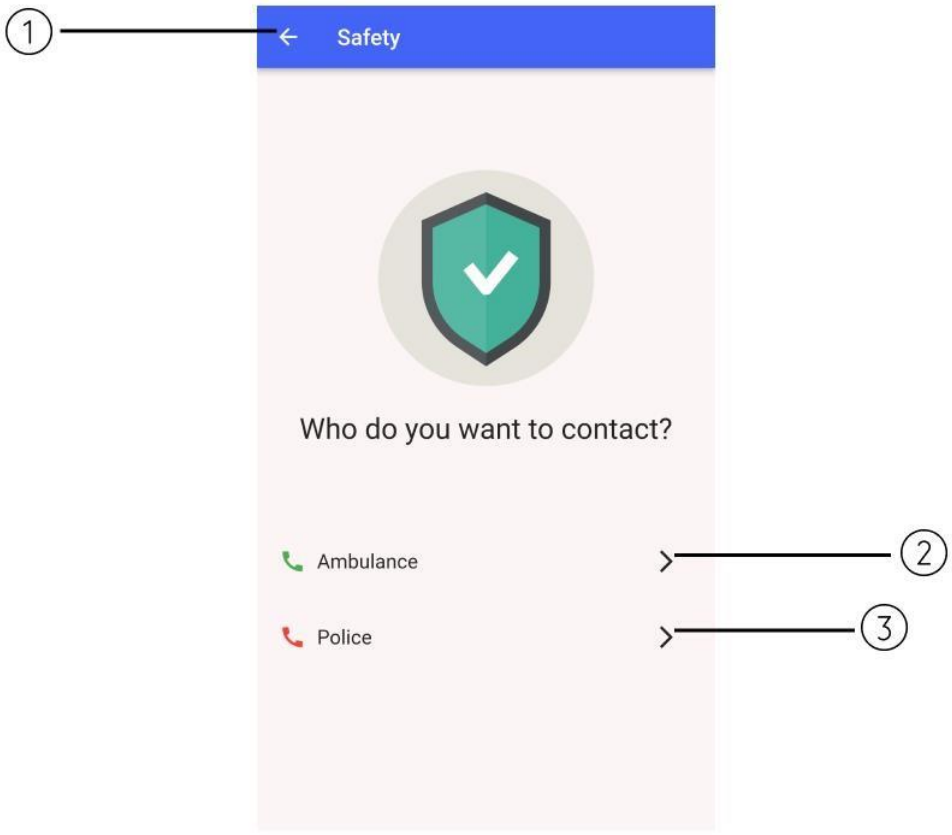
1) User enter source.
2) User enter destination.
3) User confirm for ride booking.

16.8 User current ride:



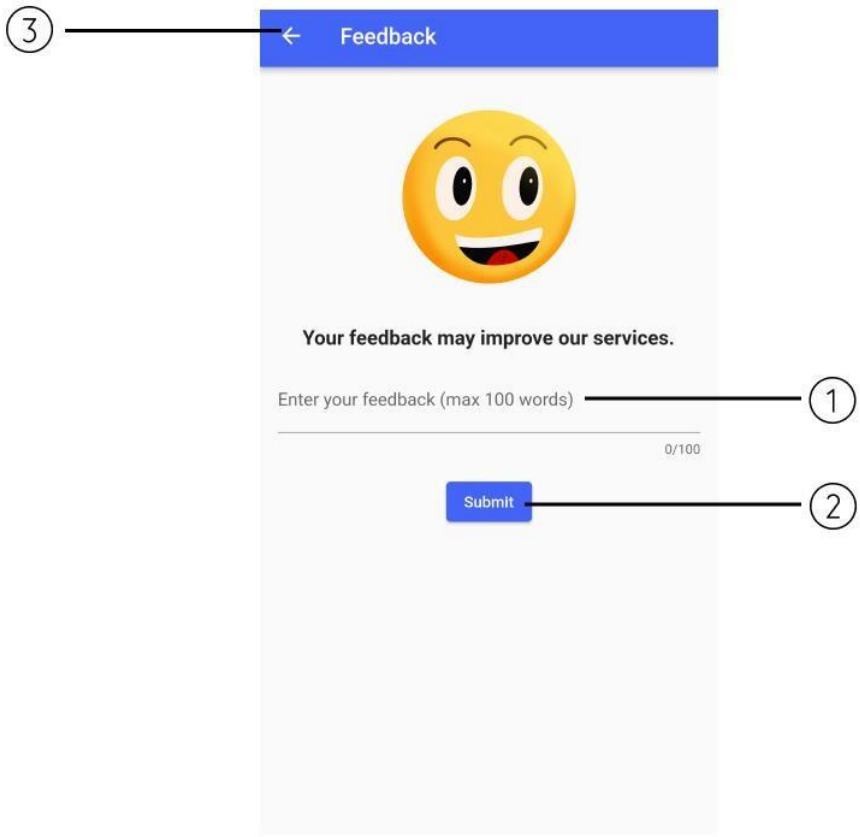
1) Go to home screen.
2) Rate driver.
3) Cancel the current ride.
4) Click button to go to previous screen. 5) Ride information.
6) Car information.

16.9 User/ Driver Safety:



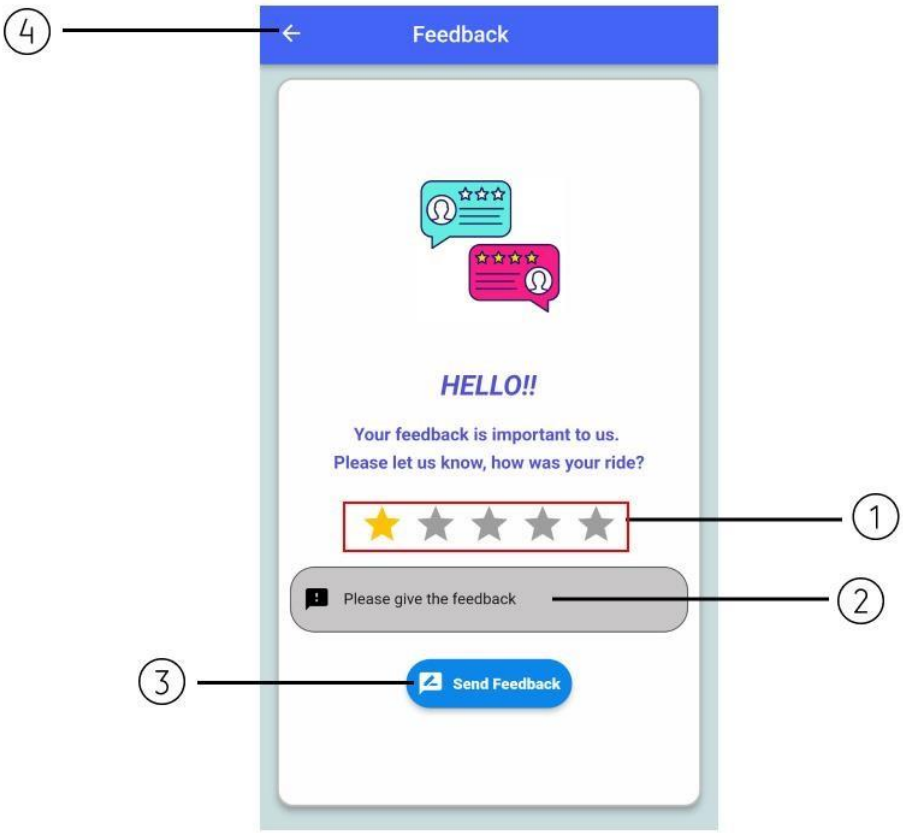
1) Click button to go to previous screen.
2) Click button to call ambulance. 3) Click button to call police.

16.10 Feedback to the application:



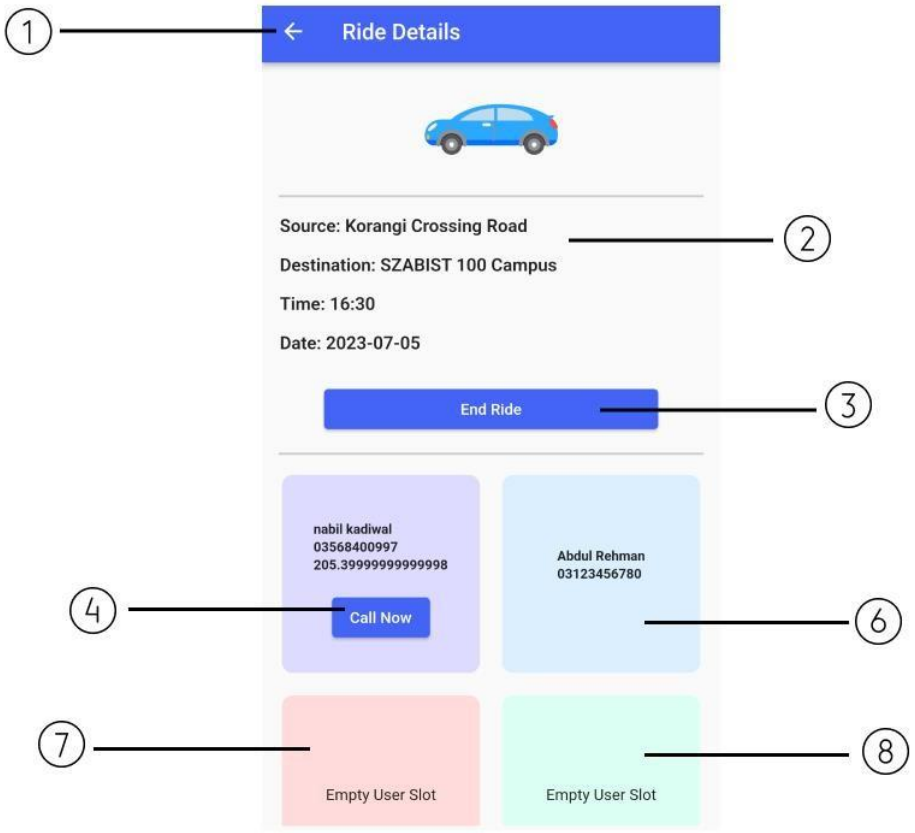
1) Click to enter the feedback.
2) Click to submit button.
3) Click button to go to previous screen.

16.11 User rates driver:



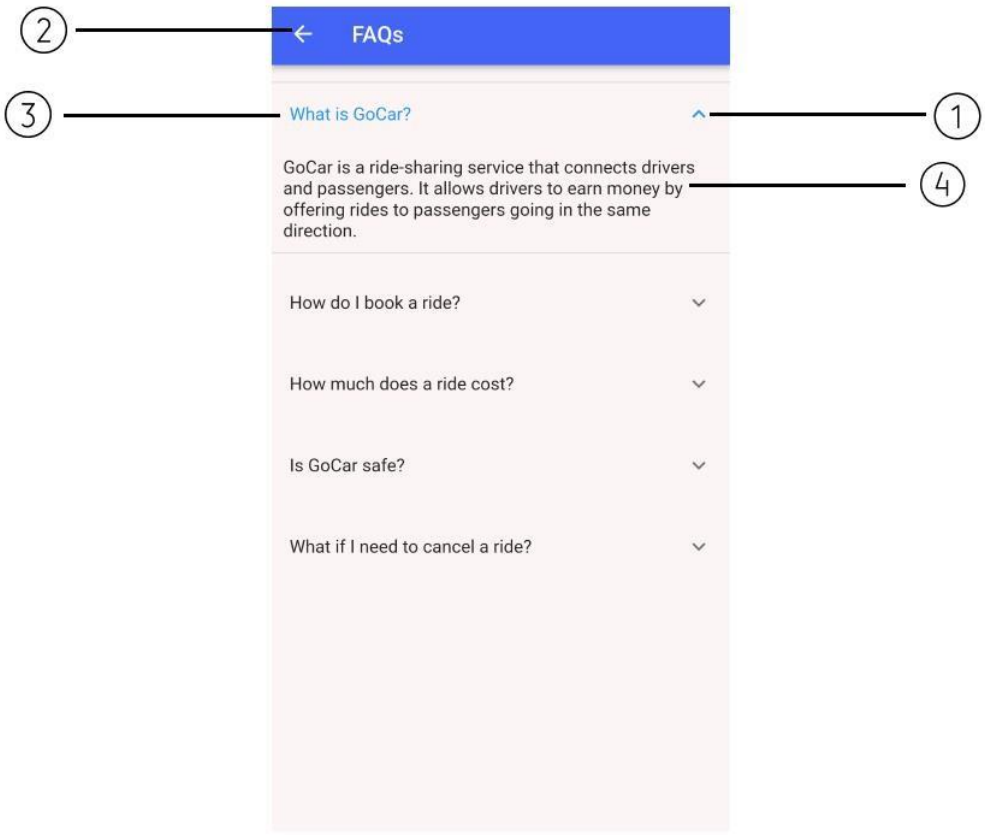
1) Select stars.
2) Enter the description for rating.
3) Click to send feedback.
4) Click button to go to previous screen.

16.12 Driver's offered ride:



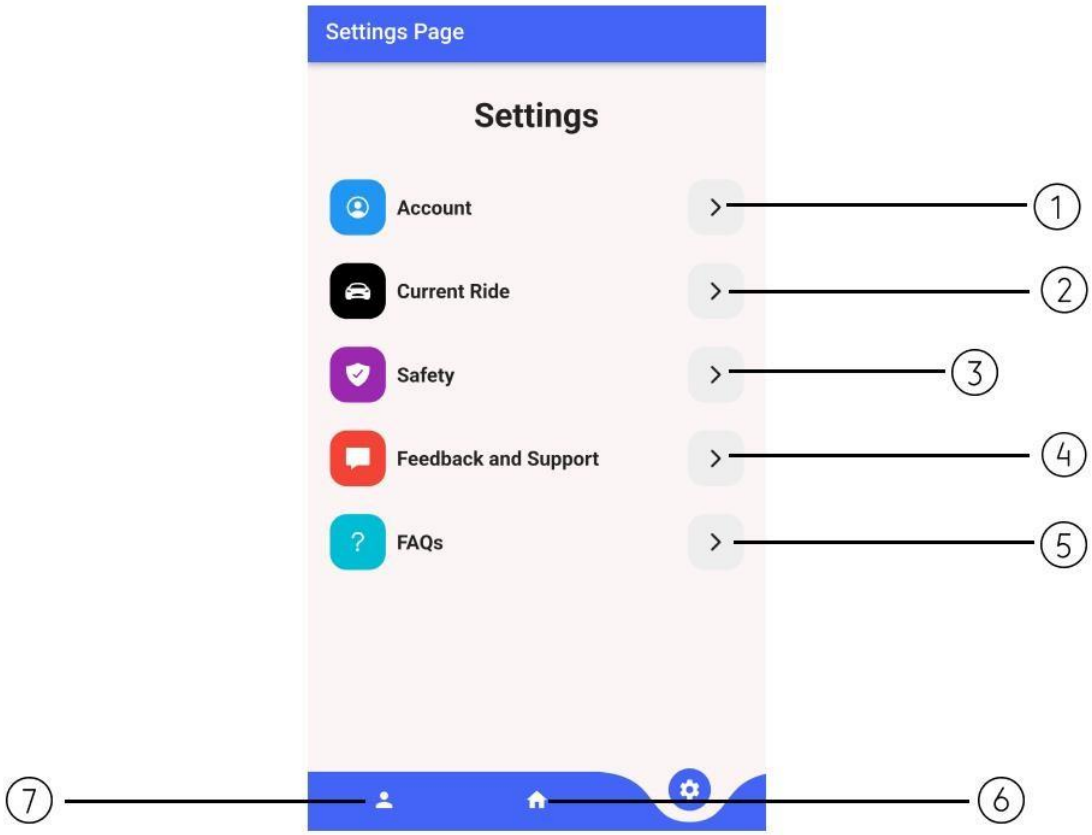
1) Click button to go to previous screen.
2) Ride information.
3) End ride button for driver.
4) Call button for calling the users.
5) User details.
6) User details.
7) User details.
8) User details.

16.13 Frequently asked questions (FAQs):



1) Click to see/ unsee the answer.
2) Click button to go to previous screen.
3) Question frequently asked by users.
4) Answer in description.

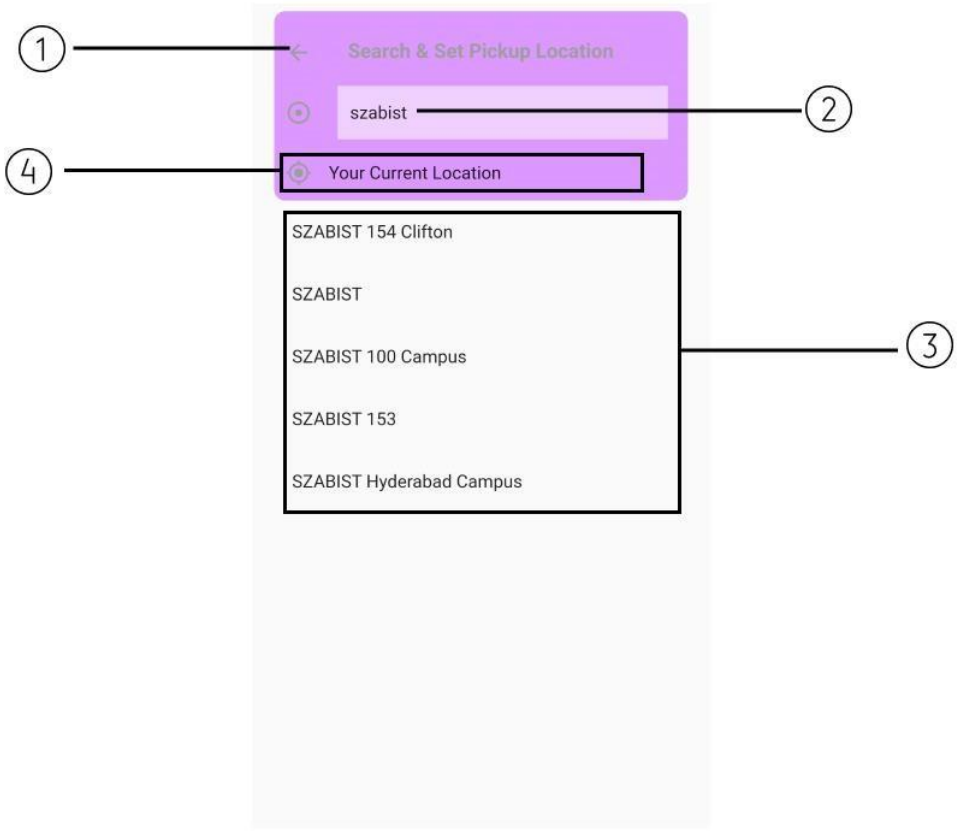
16.14 User settings:



1) Click button to see account logged in.
2) Click button to see booked ride.
3) Click button to see safety.
4) Click button to give feedback.
5) Click button to see FAQs.
6) Click button to see home screen.
7) Click button to see profile information.

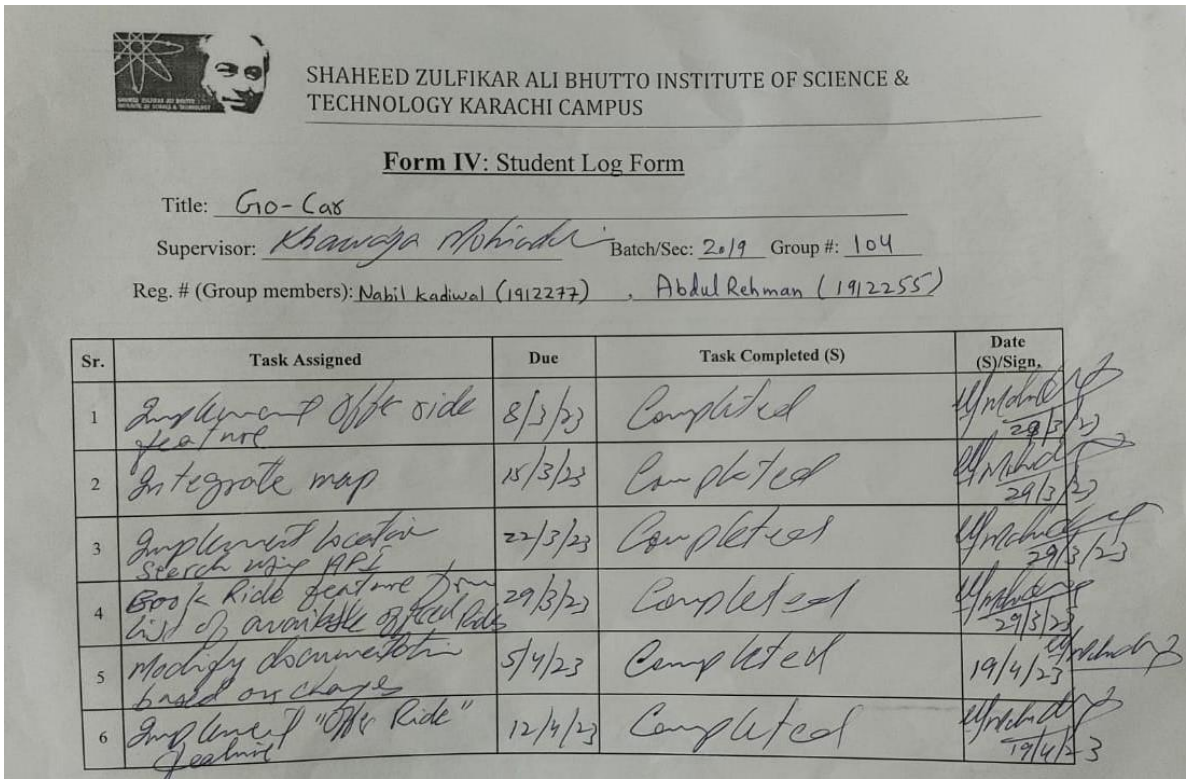
‘

16.15 User/ Driver searching for places:



1) Click button to go to previous screen. 2)
Click to search the address.
3) Click any address to set the source.
4) Click to set source address as current address

17. Log sheets



SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE & TECHNOLOGY KARACHI CAMPUS

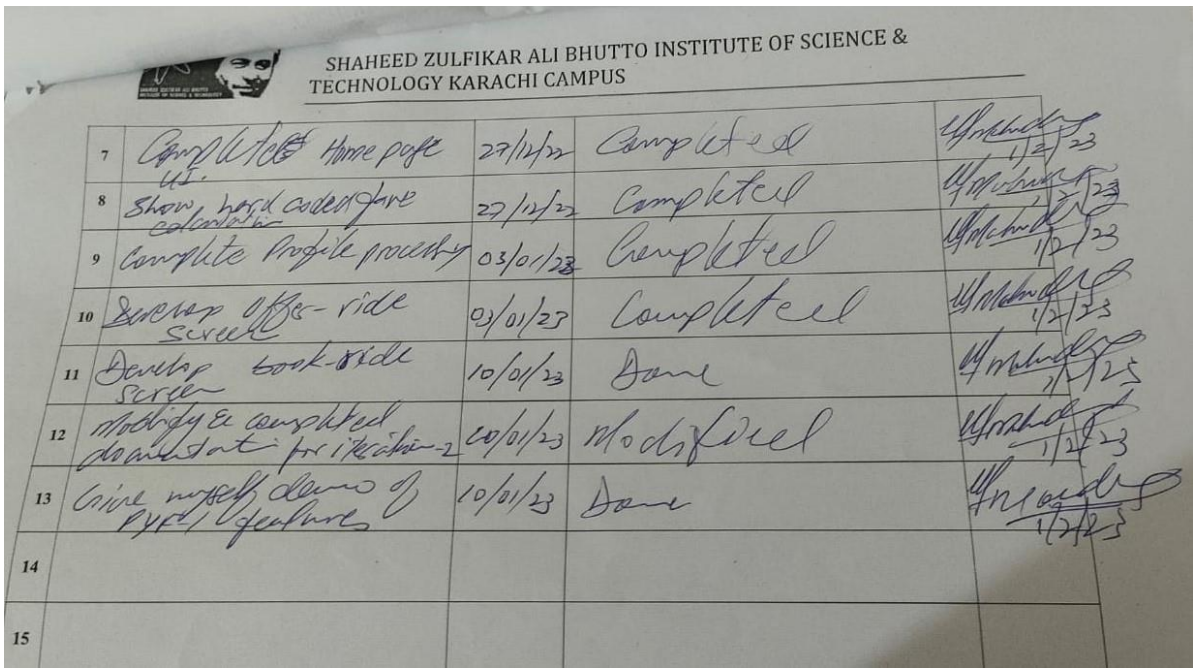
Form IV: Student Log Form

Title: Go-Car

Supervisor: Khawaja Mohiuddin Batch/Sec: 2019 Group #: 104

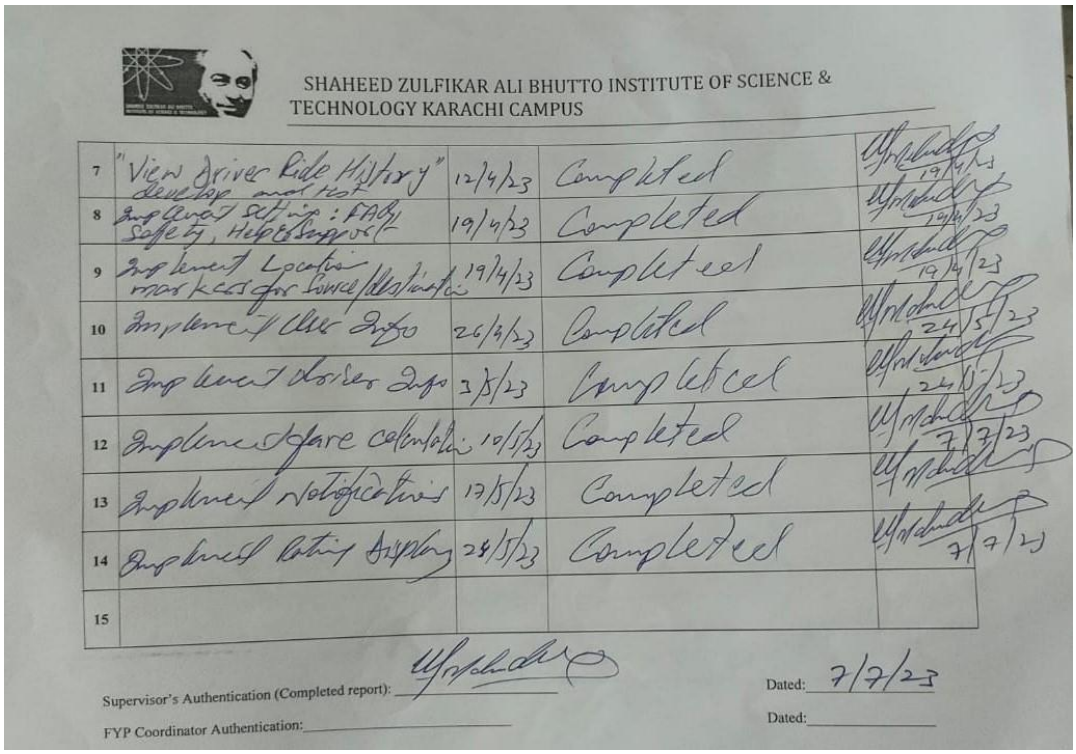
Reg. # (Group members): Nabil kadiwal (1912277) , Abdul Rehman (1912255)

Sr.	Task Assigned	Due	Task Completed (S)	Date (S)/Sign.
1	Implement offer ride feature	8/3/23	Completed	29/3/23
2	Integrate map	15/3/23	Completed	29/3/23
3	Implement location search using API	22/3/23	Completed	29/3/23
4	Book a Ride feature to list of available offered rides	29/3/23	Completed	29/3/23
5	Modify documentation based on changes	5/4/23	Completed	19/4/23
6	Implement "Offer Ride" feature	12/4/23	Completed	19/4/23



SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE & TECHNOLOGY KARACHI CAMPUS

7	Complete Home page UI.	27/12/22	Completed	1/2/23
8	Show hard coded fare calculation	27/12/22	Completed	1/2/23
9	Complete Profile page	03/01/23	Completed	1/2/23
10	Develop offer-ride screen	03/01/23	Completed	1/2/23
11	Develop book-ride screen	10/01/23	Done	1/2/23
12	Modify & complete documentation for iteration-2	10/01/23	Modified	1/2/23
13	Give mock demo of app features	10/01/23	Done	1/2/23
14				
15				

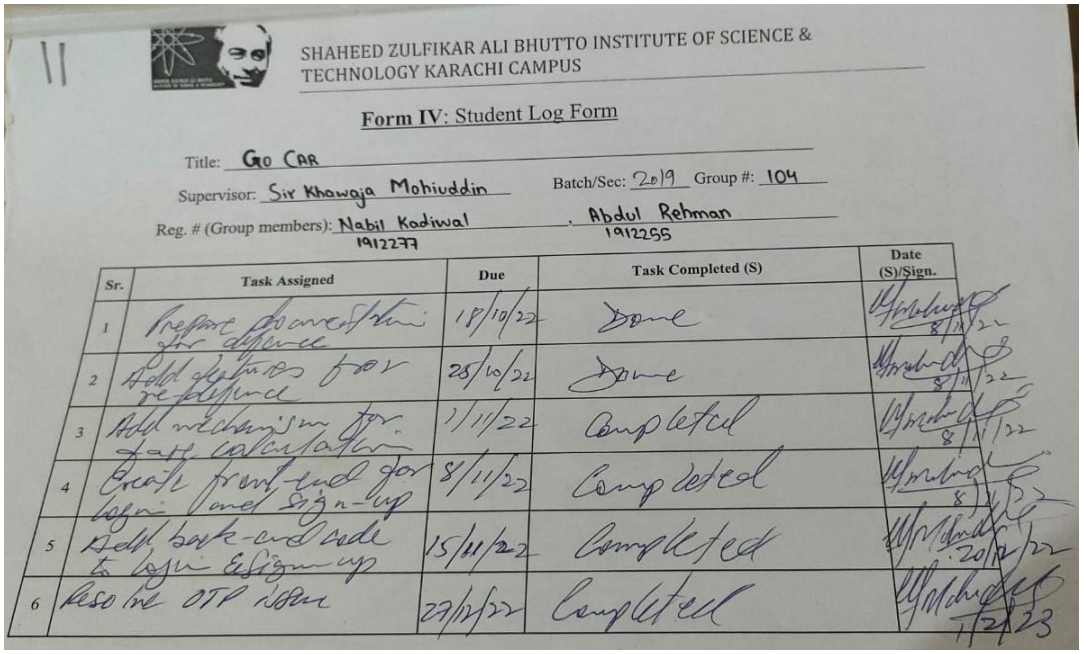


SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE & TECHNOLOGY KARACHI CAMPUS

7	"View Driver Ride History" develop and test	12/4/23	Completed	19/4/23
8	Implement Setting: FAQs, Safety, Help & Support	19/4/23	Completed	19/4/23
9	Implement Location markers for source/destination	19/4/23	Completed	19/4/23
10	Implement Ride Info	26/4/23	Completed	24/5/23
11	Implement Riders Info	3/5/23	Completed	24/5/23
12	Implement fare calculation	10/5/23	Completed	7/7/23
13	Implement notifications	17/5/23	Completed	7/7/23
14	Implement Rating Display	24/5/23	Completed	7/7/23
15				

Supervisor's Authentication (Completed report): _____ Dated: 7/7/23

FYP Coordinator Authentication: _____ Dated: _____



SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE & TECHNOLOGY KARACHI CAMPUS

Form IV: Student Log Form

Title: GO CAR

Supervisor: Sir Khawaja Mohiuddin Batch/Sec: 2019 Group #: 104

Reg. # (Group members): Nabil Kadiwal 1912277, Abdul Rehman 1912295

Sr.	Task Assigned	Due	Task Completed (S)	Date (S)/Sign.
1	Prepare document/Slides for defense	18/10/22	Done	8/11/22
2	Add features for peripherals	28/10/22	Done	8/11/22
3	Add mechanism for fare calculation	2/11/22	Completed	8/11/22
4	Create front-end for login and Sign-up	8/11/22	Completed	8/11/22
5	Add back-end code to login & Sign up	15/11/22	Completed	20/12/22
6	Resolve OTP issue	27/12/22	Completed	1/2/23

turnitin

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Nabil Kadiwal
Assignment title:	SECTIONA
Submission title:	fyp-report
File name:	Final_year_report_1912255,1912277.pdf
File size:	3.85M
Page count:	98
Word count:	11,561
Character count:	69,682
Submission date:	06-Aug-2023 04:02PM (UTC+0500)
Submission ID:	2141968799

Final Year Project Report

Go Car

Project Team:

Nabil Kadiwal 1912277
Abdul Rehman 1912255

1/08/2023

Project Supervisor:

Sir Khawaja Mohiuddin

Submitted in the partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer Science in the

Faculty of Computing and Engineering Sciences

fyp-report

ORIGINALITY REPORT

16%	15%	1%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	web.stonehill.edu Internet Source	4%
2	www.coursehero.com Internet Source	3%
3	www.cardonationonline.info Internet Source	2%
4	Submitted to Higher Education Commission Pakistan Student Paper	1%
5	scholar.ppu.edu Internet Source	1%
6	Submitted to University of Central Florida Student Paper	1%
7	Submitted to Sri Lanka Institute of Information Technology Student Paper	1%
8	ir.iba.edu.pk Internet Source	<1%
9	web.cs.dal.ca	

	Internet Source	<1%
10	Submitted to Colorado Technical University Online Student Paper	<1%
11	Submitted to University of Greenwich Student Paper	<1%
12	documents.mx Internet Source	<1%
13	www.slideshare.net Internet Source	<1%
14	technodocbox.com Internet Source	<1%
15	Submitted to Birla Institute of Technology and Science Pilani Student Paper	<1%
16	Submitted to Harrisburg University of Science and Technology Student Paper	<1%
17	adeliaoctaviani.blogspot.com Internet Source	<1%
18	Submitted to Middle East College of Information Technology Student Paper	<1%
19	www.gcreddy.net Internet Source	

		<1%
20	www.macintosh-data-recovery.com Internet Source	<1%
21	Do Hyung Kim. "Method and Implementation for Consistency Verification of DEVS Model against User Requirement", 2008 10th International Conference on Advanced Communication Technology, 2008 Publication	<1%
22	Submitted to University of Technology, Sydney Student Paper	<1%
23	pdfcoffee.com Internet Source	<1%
24	pdfslide.net Internet Source	<1%
25	de.slideshare.net Internet Source	<1%
26	ethesis.nitrkl.ac.in Internet Source	<1%
27	gretlml.univpm.it Internet Source	<1%
28	Susan K. Land, Douglas B. Smith, John W. Walz. "12207 Primary Life Cycle Processes",	<1%