VeriCab

13.09.2019

Team Members: .

Aditya Joshi , Animesh Kumar , Rashmika Calve , Arul Selvi , Riya George ,Lokeshwar Tabjula, Vignesh Iyer , Yugansh Singh

Verizon India
Olympia Technology Park
Chennai, T.N.-600032

VeriCab Application

By

VeriCab Team

For the Verizon India training project

Verizon India

Olympia Technology Park

Guindy

Chennai-600032

2019

ABSTRACT

This Car Rental Application “VeriCab” project is designed to aid the car rental company to enable renting of cars through an online system. It helps the users to search for available cars view profile and book the cars for the time period. It has a user-friendly interface which helps the user to check for cars and rent them for the period specified. They could also make payment online. The rental cars shall be categorized into economy, premium,sedan,etc. Based on the type of car required by the customer, the user will get a list of drivers with ratings. According to the rating the user can easily manage his/her booking. The use of internet technology has made it easy for the customers to rent a car any time. This Car Rental Application makes the bookings easy. It saves time and labour. This Application asks the user for information such as the date and time of journey, type of car etc. Using these details, this application shall help the customer to book a car for the journey. It also gives access to admin to see all the bookings and driver details so that if the admin wants to remove a driver according to rating i.e below 3 star. This application gives a Payment gateway to pay online.

ACKNOWLEDGEMENT

In completing this VeriCab Application project we have been fortunate enough to have help, support and encouragement from many people. I would like to acknowledge them for their cooperation.

First and foremost deeply thankful to our scrum master SelvaRaju, for his wonderful guidance during this project work in the field of Full Stack Development, at Verizon India. We are also thankful for his continuous feedback and encouragement throughout this project work. His broad knowledge and hardworking attitude has left us with very deep impressions and they will greatly benefit us throughout our life.

We would like to thank our company Verizon India for their support throughout this project work.

TABLE OF CONTENTS

List of Figures

1.0 INTRODUCTION

A country like India where most of the population belongs to the middle class family. Most of them can't afford a car. In this era, people have less time and more work that force them to travel to different places to work, business meetings and tourism.

The public vehicle is not the best option. Public vehicles are crowded, generally not running on time e.g. trains, buses etc. Here is our main problem to reach at own place on time with comfort. Thus, Car renting come up with a solution. People who can't afford a car.

By this application they can book a car for trips, marriage, business meetings, office to home, home to office, home to a market where ever they want to travel as simple as that. VeriCab specializing in renting cars to clients.

It is an online system through which clients view available cars, signup, login, view profile and book car.In this it can either be logged in as admin or user/clients.They even do not need to drive themselves driver is available with car. Prices are very less with all comfort. And for ease there are different payment modes.

2.0 OVERALL DESCRIPTION:

2.1Description:

Any member can register and view available cab services.

Only registered members can book a rental cab.

About Us page will give the information of our team members.

ContactUs page is available to contact Admin for queries.

There are two roles available: User and Admin.

- User can view and book cabs.
- An Admin has some extra privileges including all privilege of firing cab driver.
- ❑ Admin can add cars, edit cars information and remove cars.
- ❑ Admin can add cab driver, edit cab driver information and can remove cab driver.

2.2Using the code:

1. Attach the database in your "MySQL Server".

2. Run the application on Microsoft Visual Studio as web site.

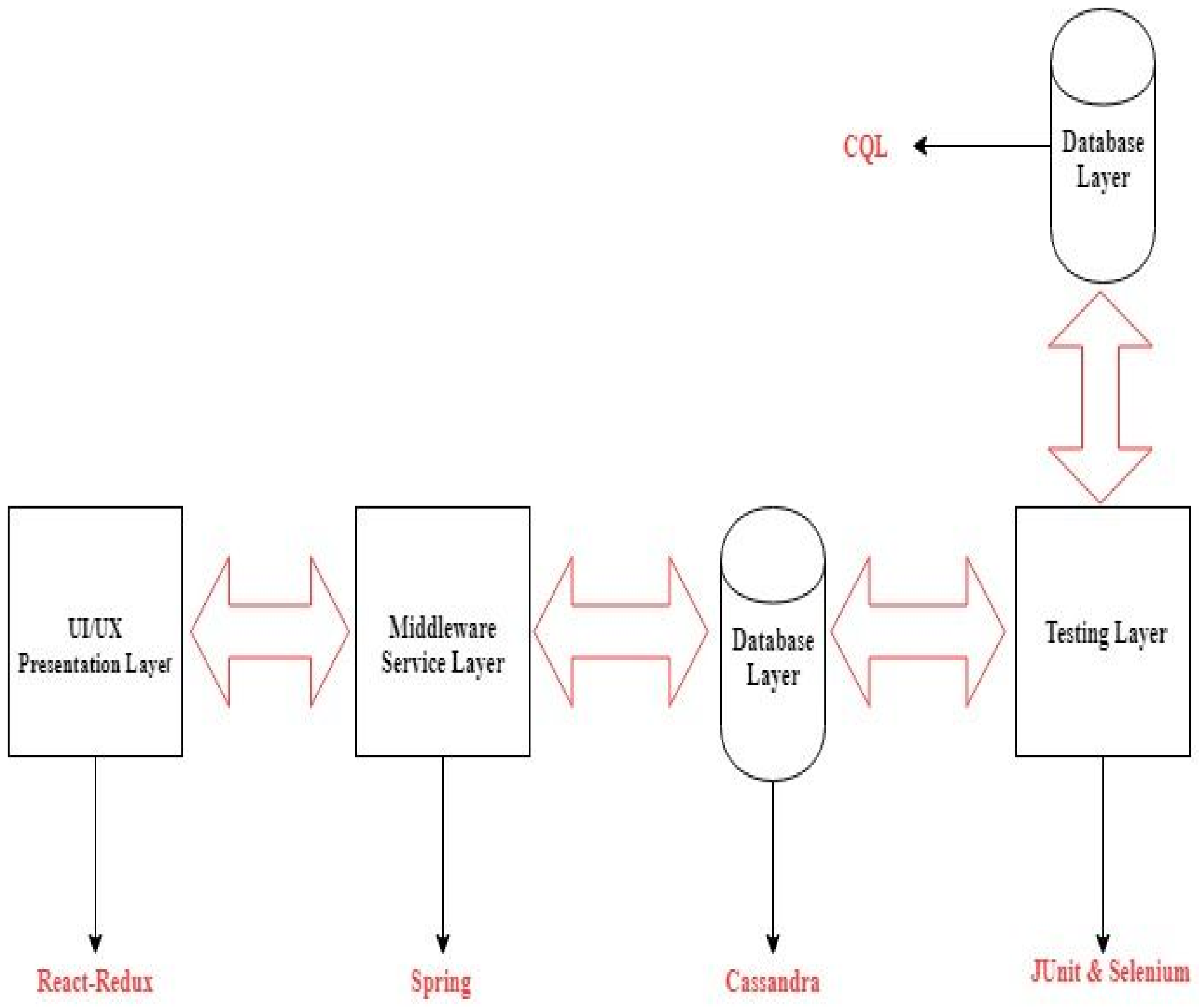
3. Locate the database.

2.3 Web Pages details:

- ❖ Home Page
- ❖ Our Team Page
- ❖ Login Page
- ❖ Admin Page
- ❖ Register Page
- ❖ Passenger Page
- ❖ Driver Page
- ❖ Payment Page

2.4 Project Detail:

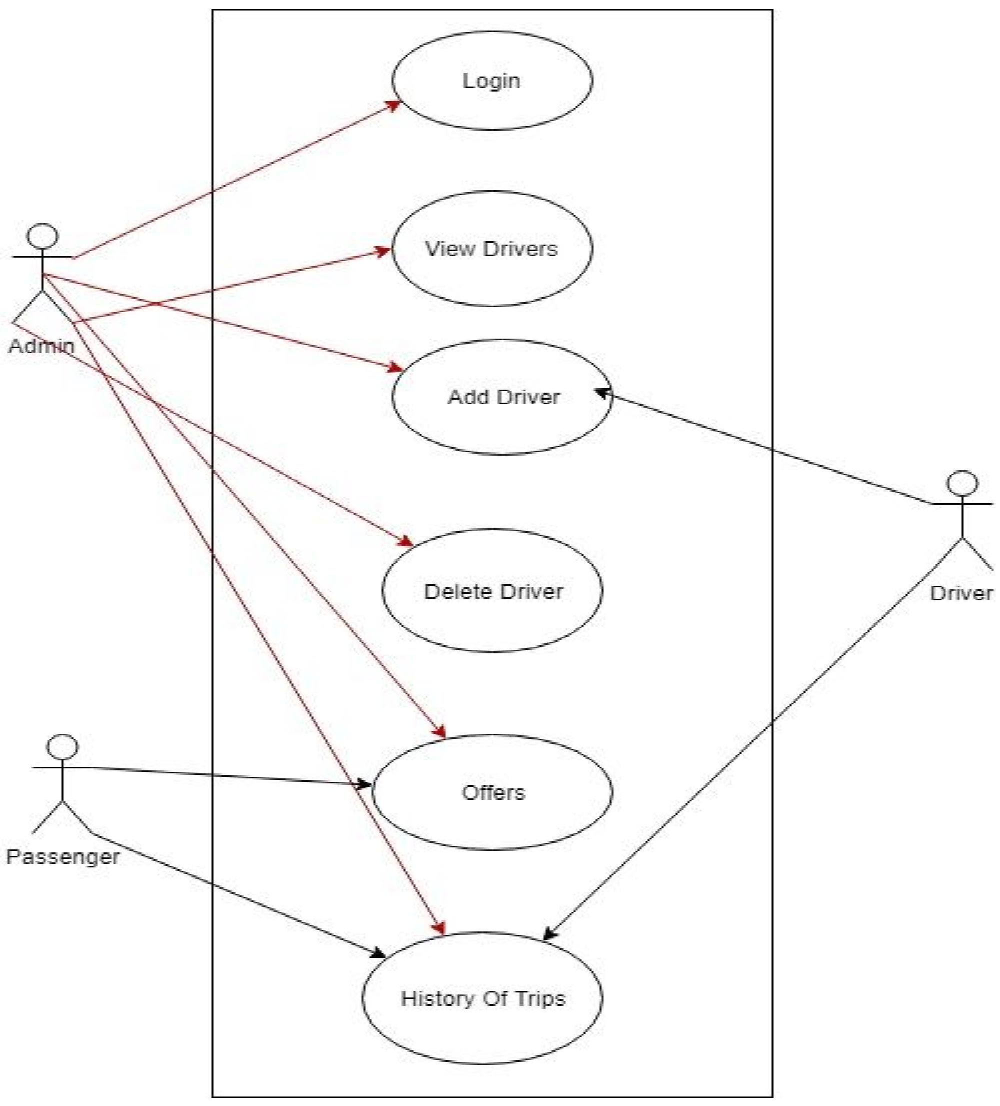
Layer Architecture



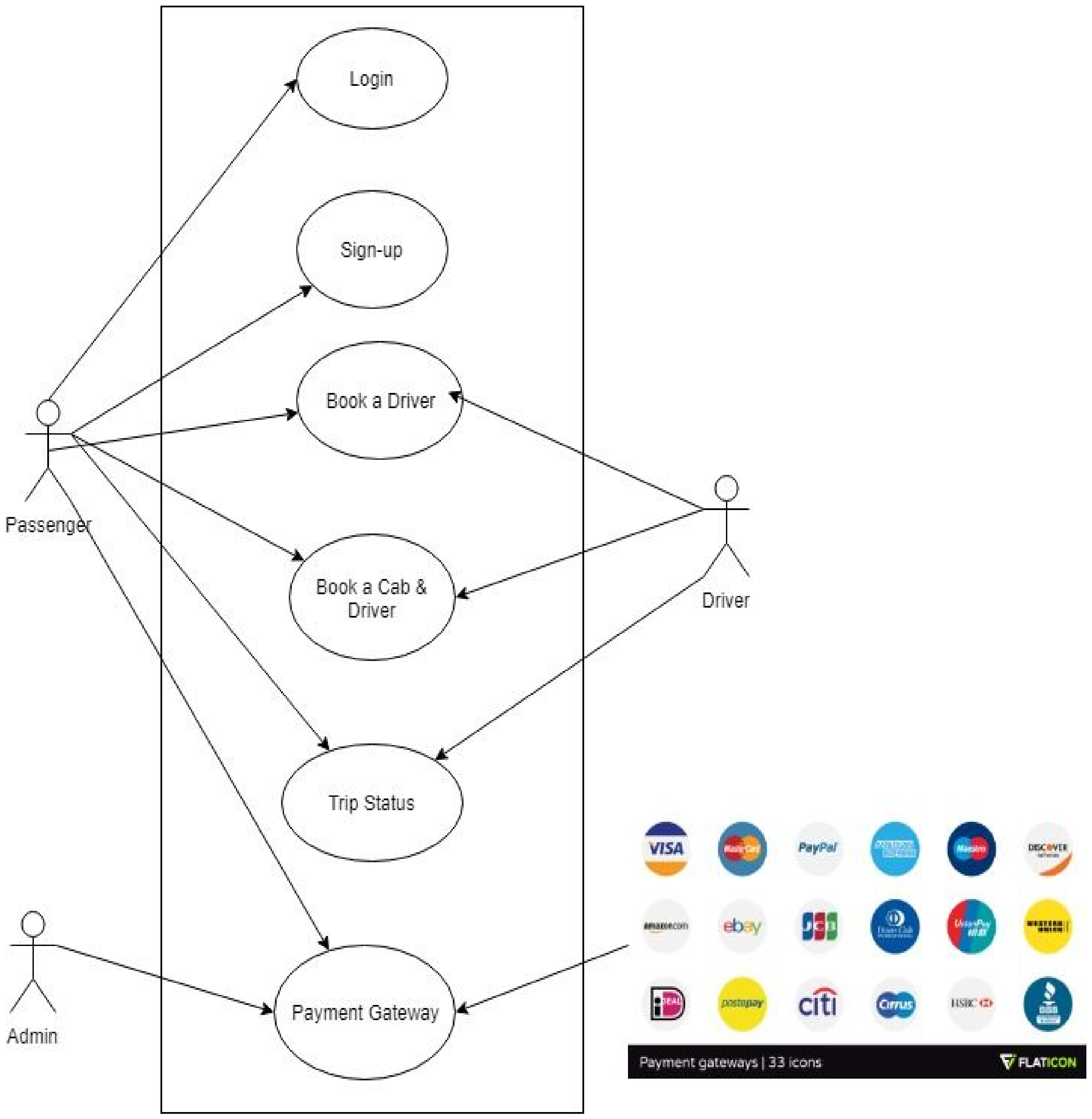
3.0 System Requirements:

3.1 Use Case Diagrams

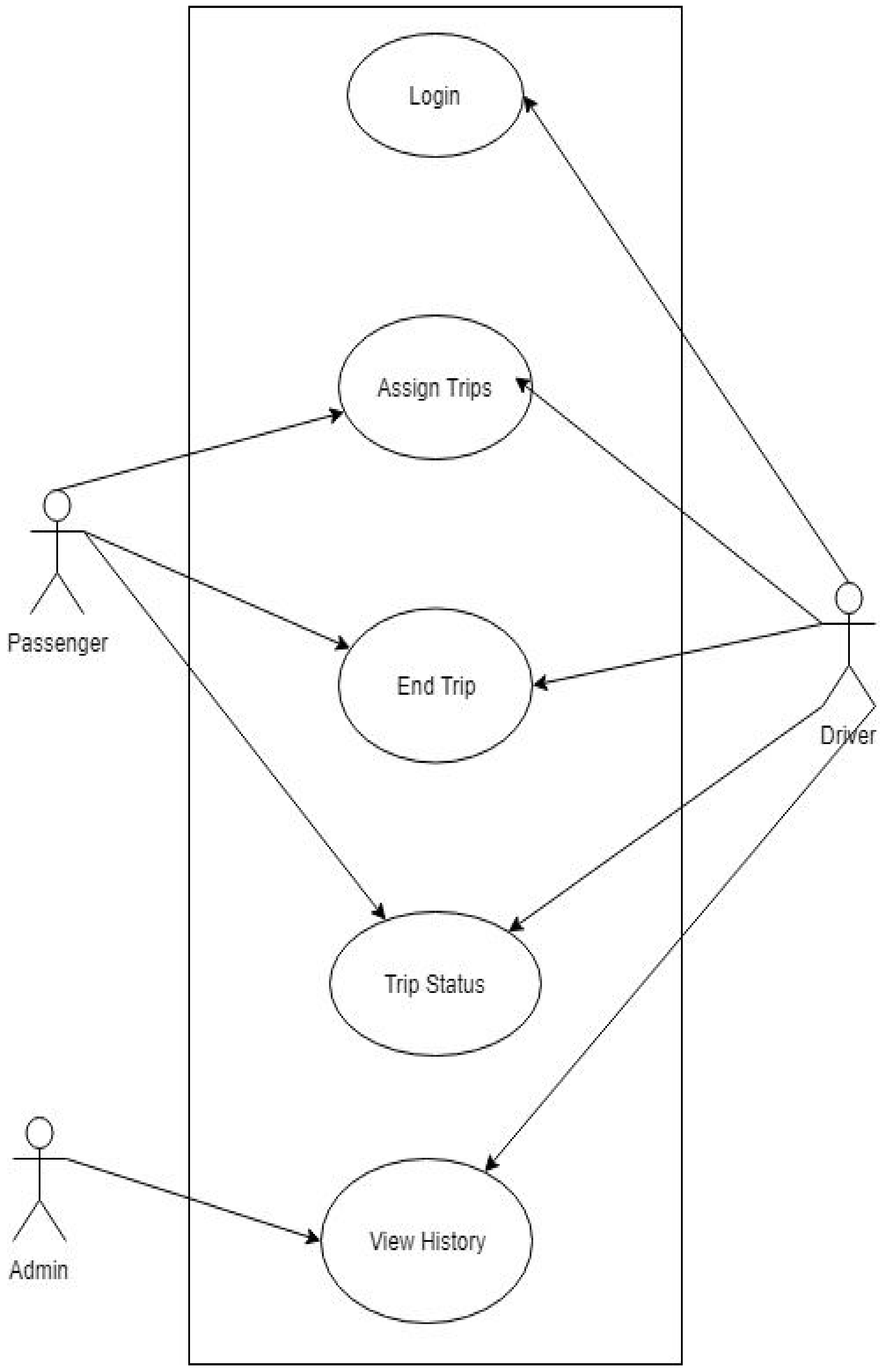
ADMIN PAGE



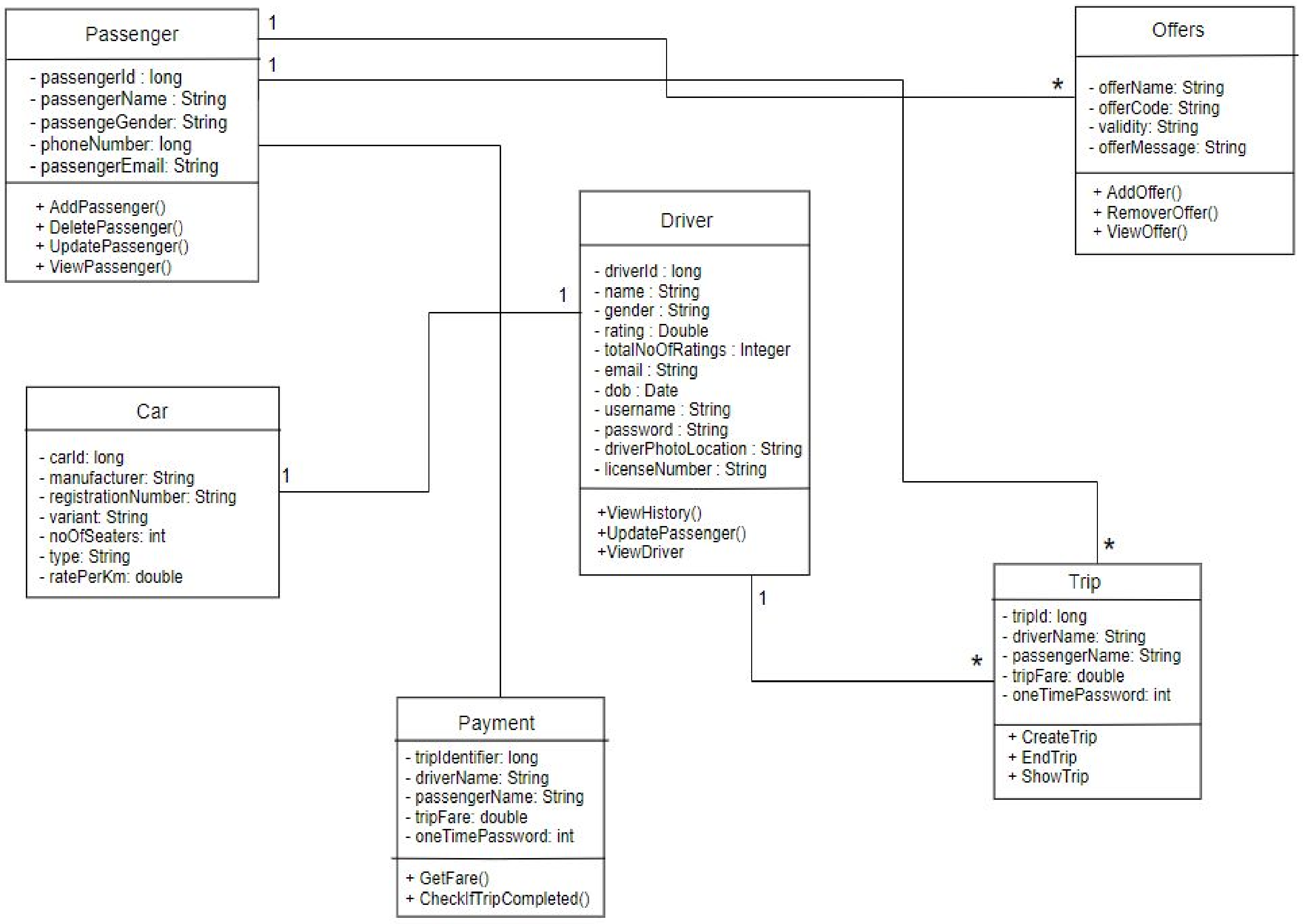
PASSENGER PAGE



DRIVER PAGE

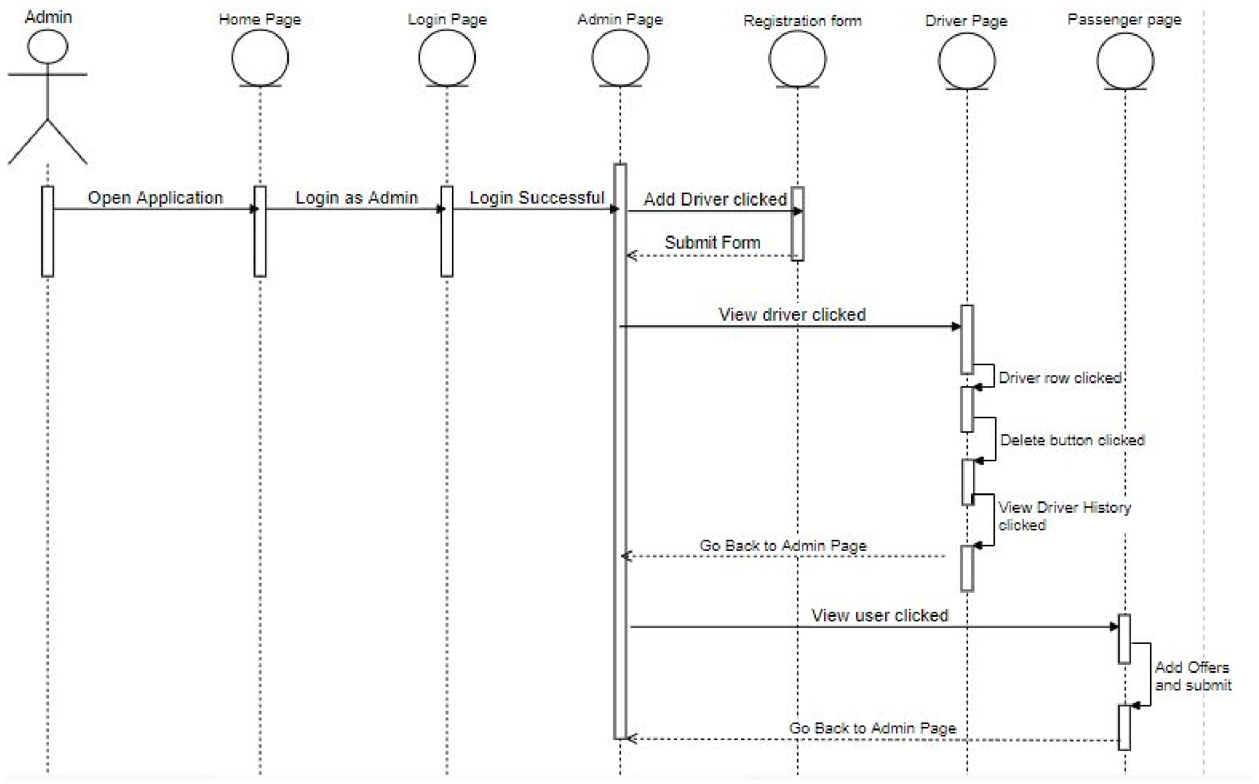

Login
Assign Trips
End Trip
Trip Status
View History
Passenger
Driver
Admin

3.2 CLASS DIAGRAM

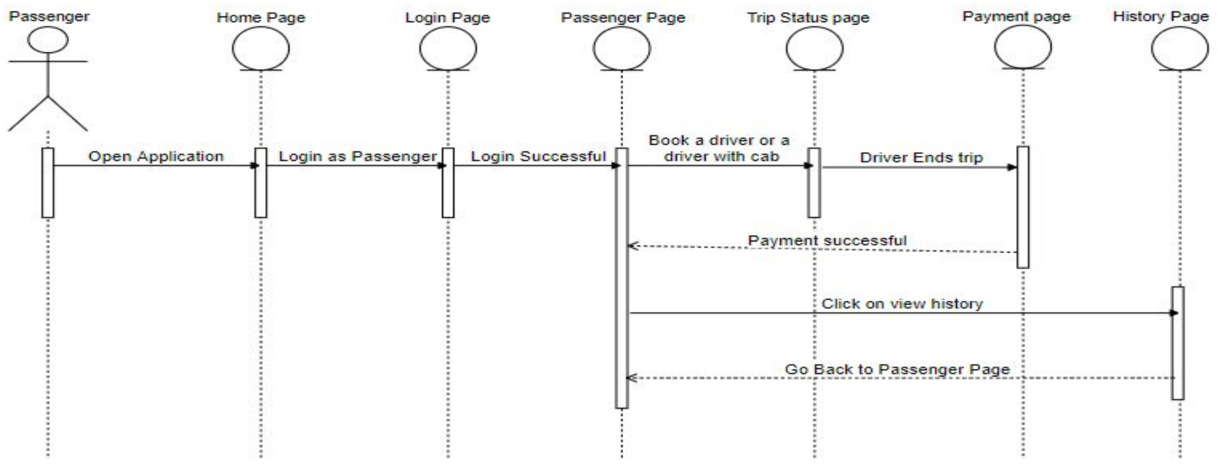


3.3 SEQUENCE DIAGRAM

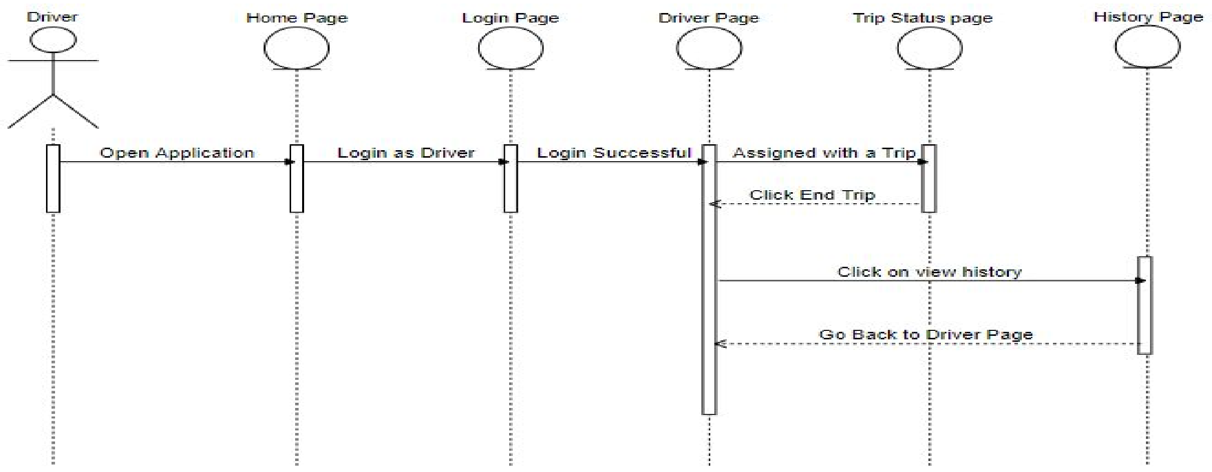
Admin



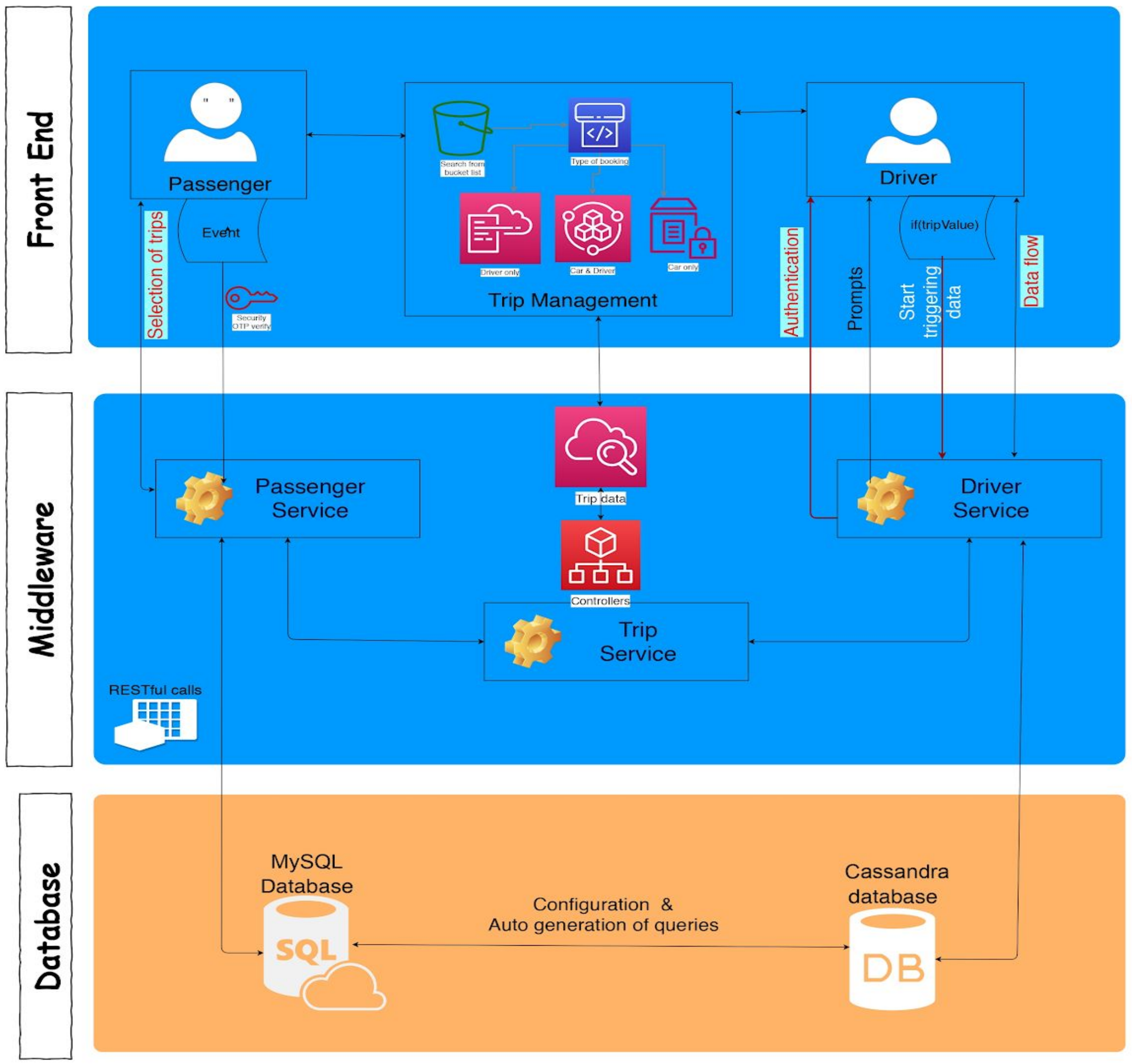
Passenger



Driver



3.4 HIGH LEVEL DIAGRAM



High level Architecture Diagram for VeriCab project

1. Aditya Joshi
2. Animesh Kumar
3. Arulselvi A R
4. Lokeshwar Tabjula
5. Rashmika calve
6. Riya George
7. Vignesh R
8. Yugansh Singh

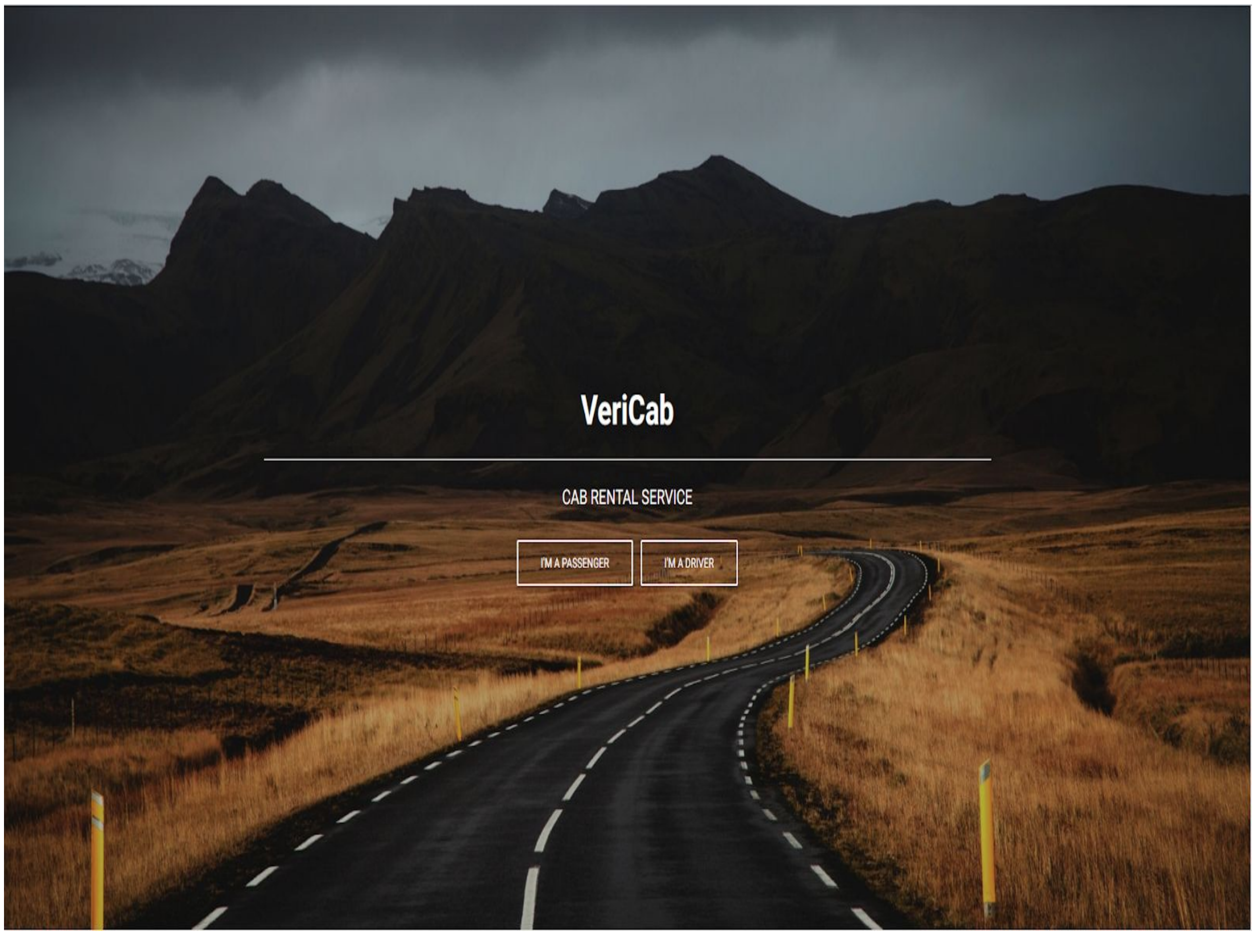
VeriCab

verizon

4.0 VERICAB APPLICATION

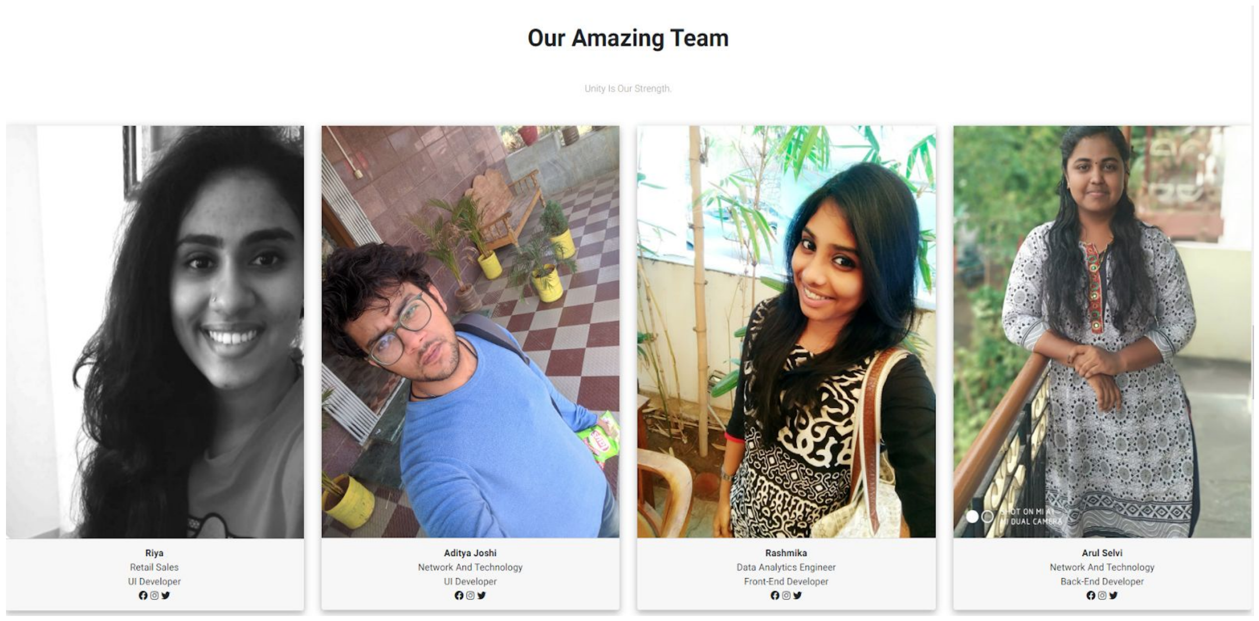
4.1 HOME PAGE

This page is a simple template page of our application in which there are two buttons i.e. I am a Passenger and I am a Driver. If you click it on the “I am a Passenger” button then it will go to Login button then it will redirect it to Login page.If you click it on the “I am a driver” button then it will redirect to Driver page.



4.2 OUR TEAM PAGE

Our Team Page in which it will display pictures along with details of team members who contributed in creating and making this application run. This will show the name of the team member, the name of the field in which they contributed and their social media links so that others can connect with them easily.



Our Amazing Team

Unity Is Our Strength.

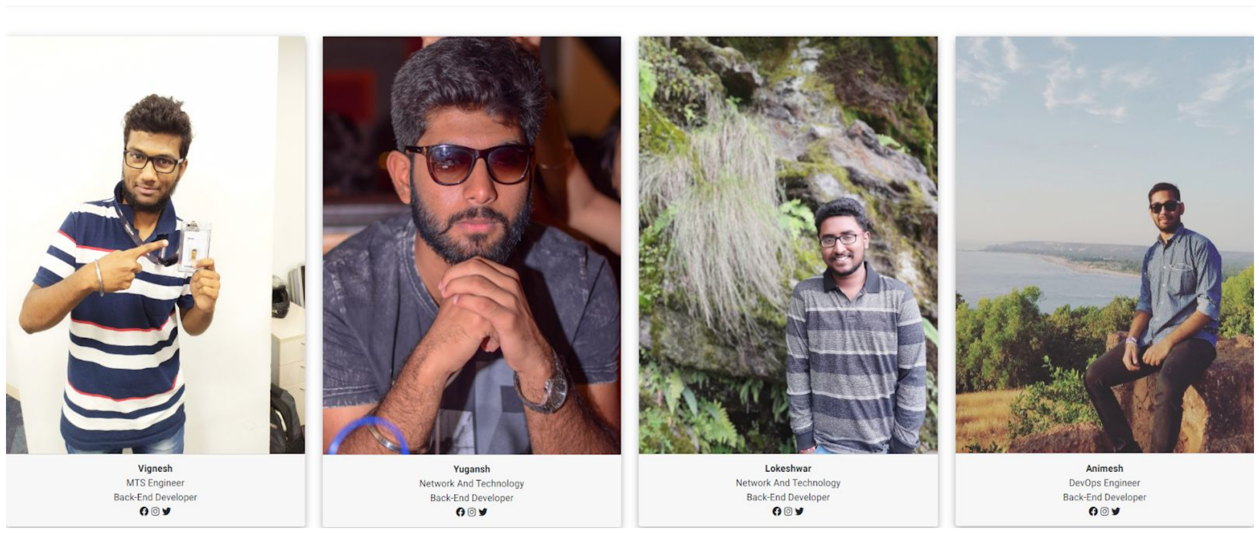
Riya
Retail Sales
UI Developer

Aditya Joshi
Network And Technology
UI Developer

Rashmika
Data Analytics Engineer
Front-End Developer

Arul Selvi
Network And Technology
Back-End Developer

vb



Vignesh
MTS Engineer
Back-End Developer

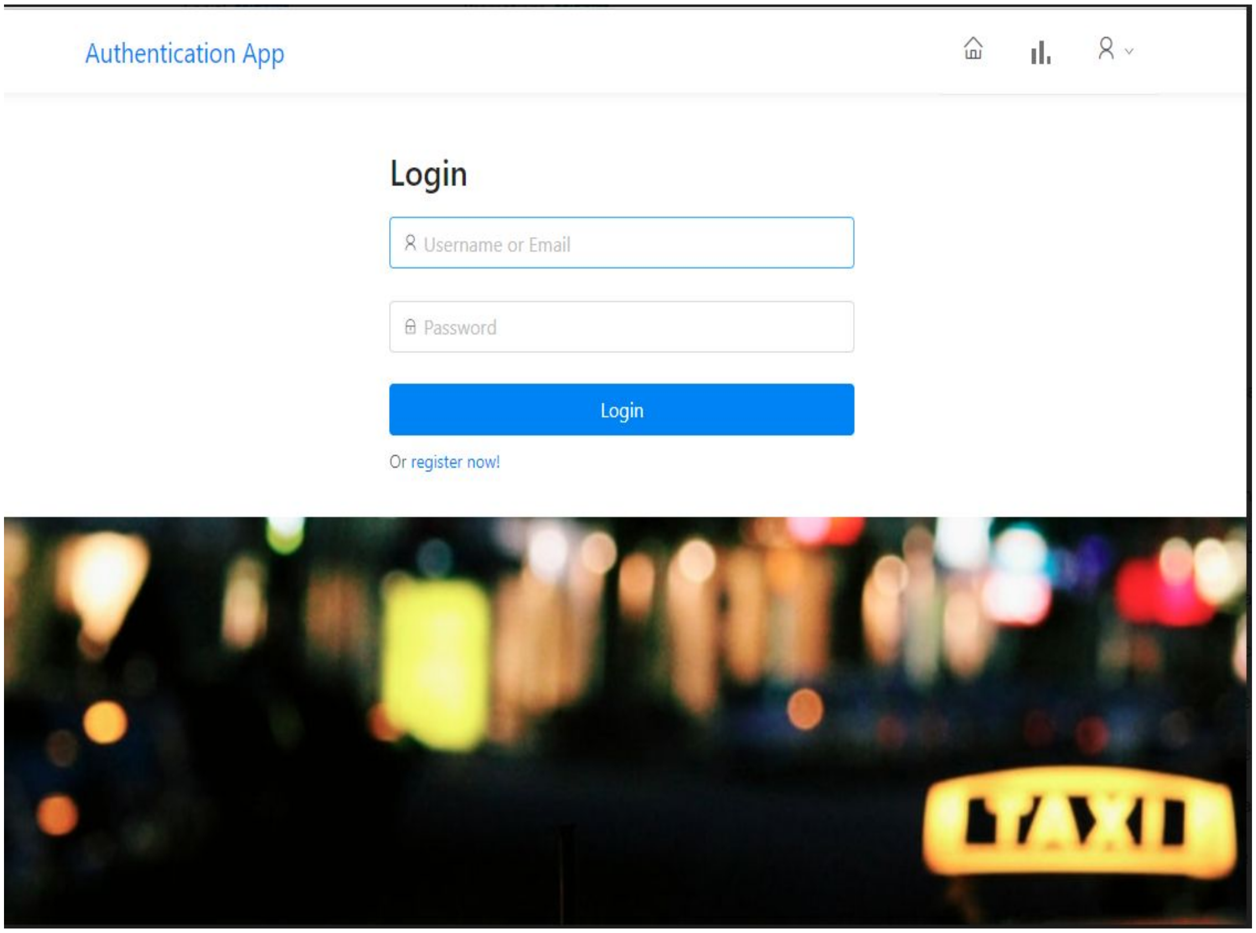
Yugansh
Network And Technology
Back-End Developer

Lokeshwar
Network And Technology
Back-End Developer

Animesh
DevOps Engineer
Back-End Developer

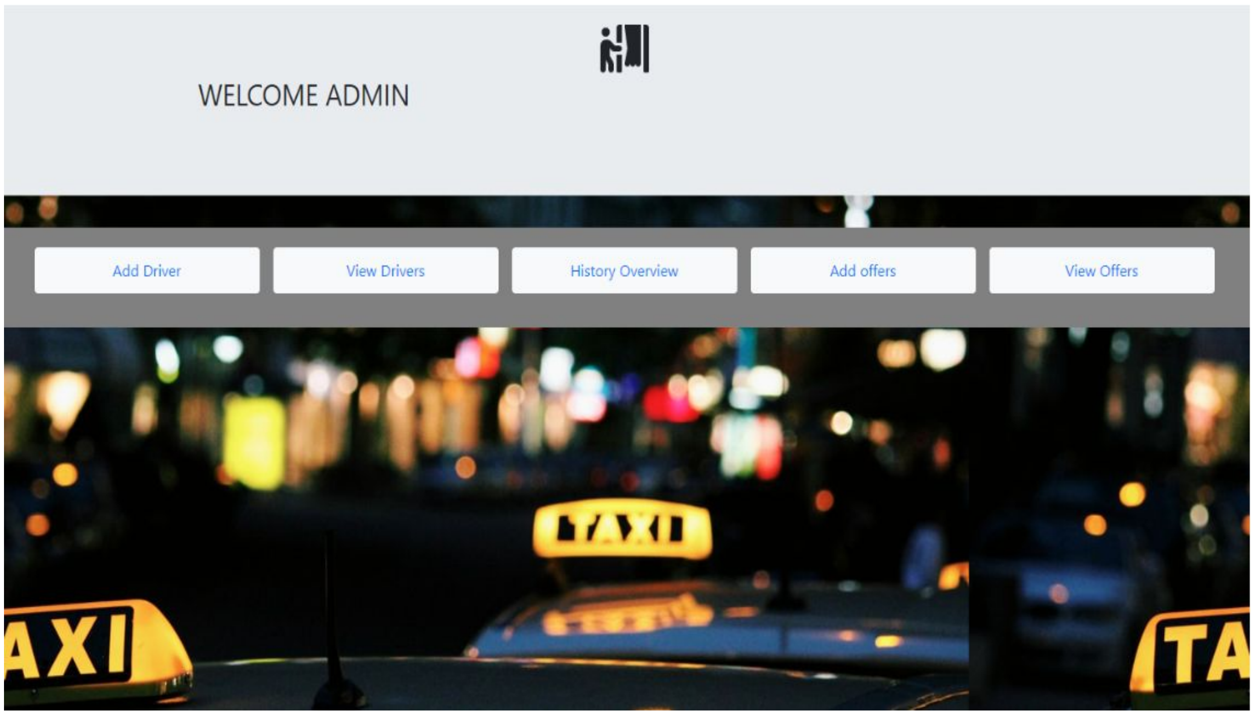
4.3 LOGIN PAGE

It is the first page which appears when we open this application. So “Login Page” comprises of Login button where it asks for user-name and password .It validates and verifies the credentials and creates and manages jwt token and session using Authentication service. Then according to the user-name it checks whether it is logged in as an Admin or Driver or Passenger. Then it is redirected to their respective page.



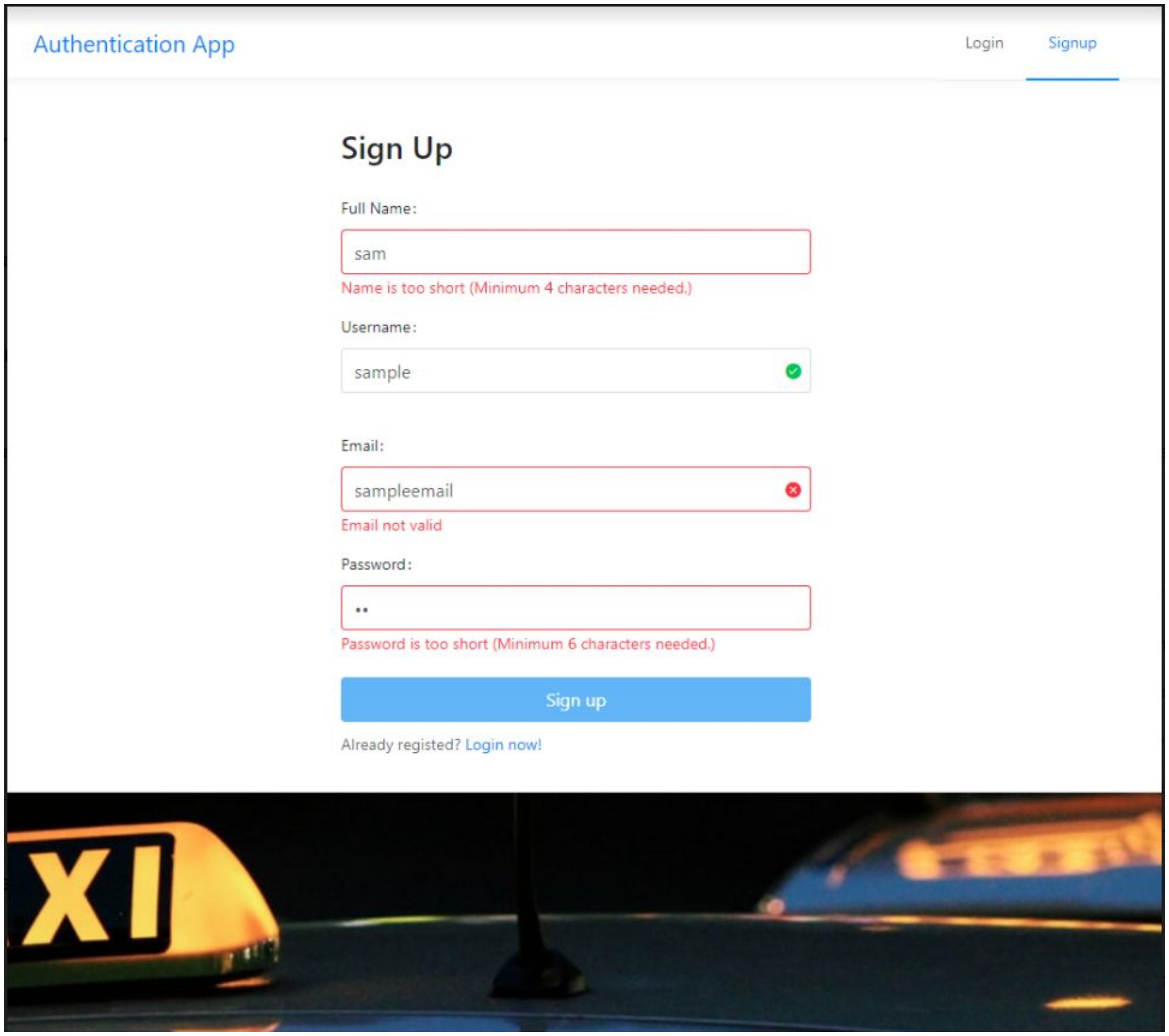
4.4 ADMIN PAGE

Once you are logged in as an Admin then you will get 4 options as Offers, View Drivers, Add Driver and History of all trips. So Admin Service implementation that takes place which does CRUD operations on drivers. Admin UI implementation basically facilitates management of drivers , offers and history of trips. Admin can add offers so that gives push notification to the passengers which is updated to the passenger database. In view divers admin can see all the names of drivers with delete option placed with their respective name so that admin can delete the selected driver based on their ratings. Being an admin you are privileged to see the history of all trips.



4.5 REGISTER PAGE

Register page comes when your user-name is not matched as admin as it authenticates with the admin database. Once you have landed on this page you will be asked to sign up either as a passenger or a driver. Once you have registered it will go into the respective database for updation. Then it assigns a role using Authentication Service.



Sign Up

Full Name: sam

Name is too short (Minimum 4 characters needed.)

Username: sample

Email: sampleemail

Email not valid

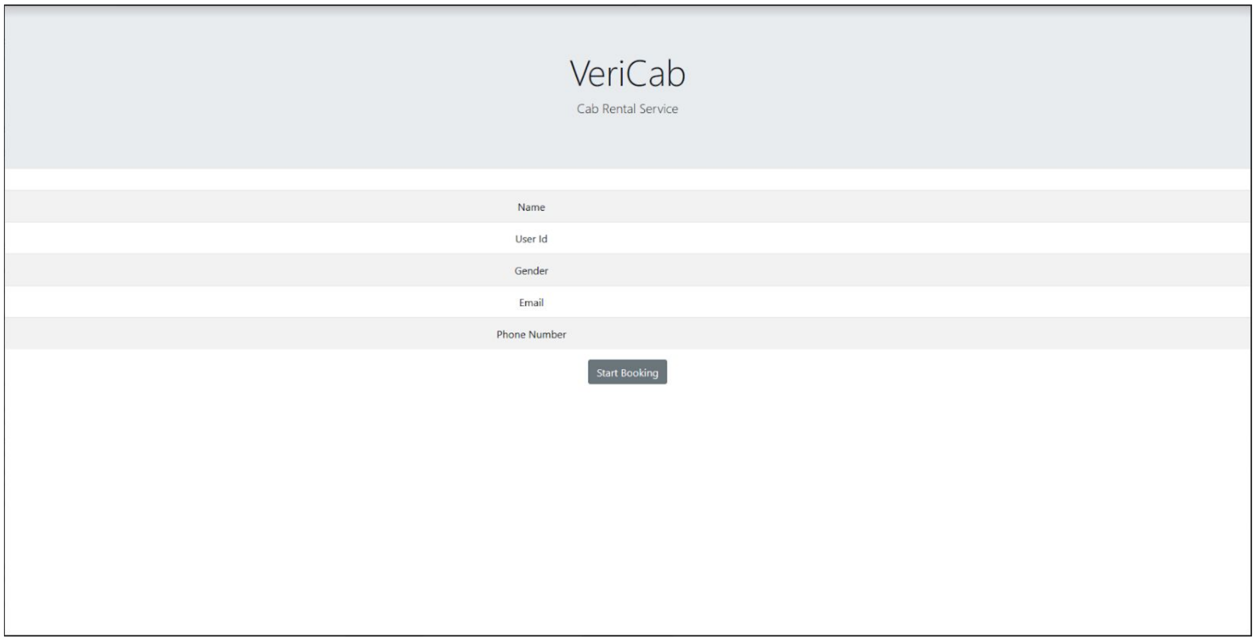
Password: ••

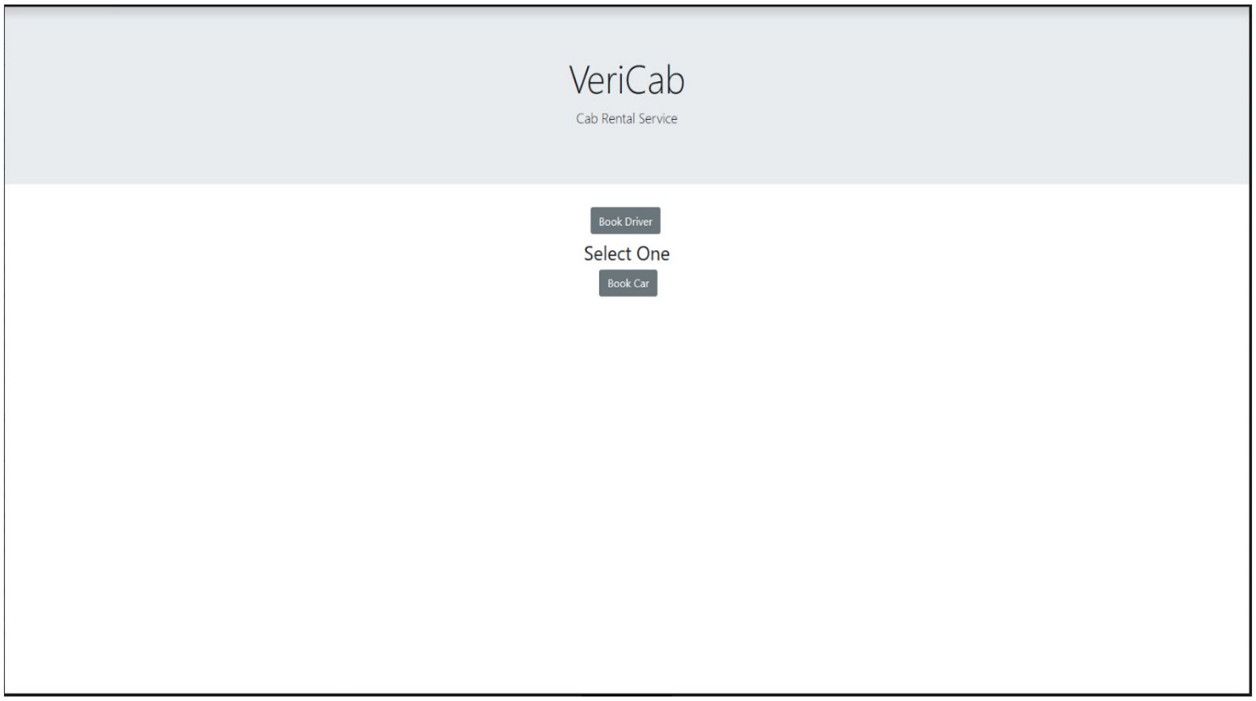
Password is too short (Minimum 6 characters needed.)

Already registed? Login now!

4.6 PASSENGER PAGE

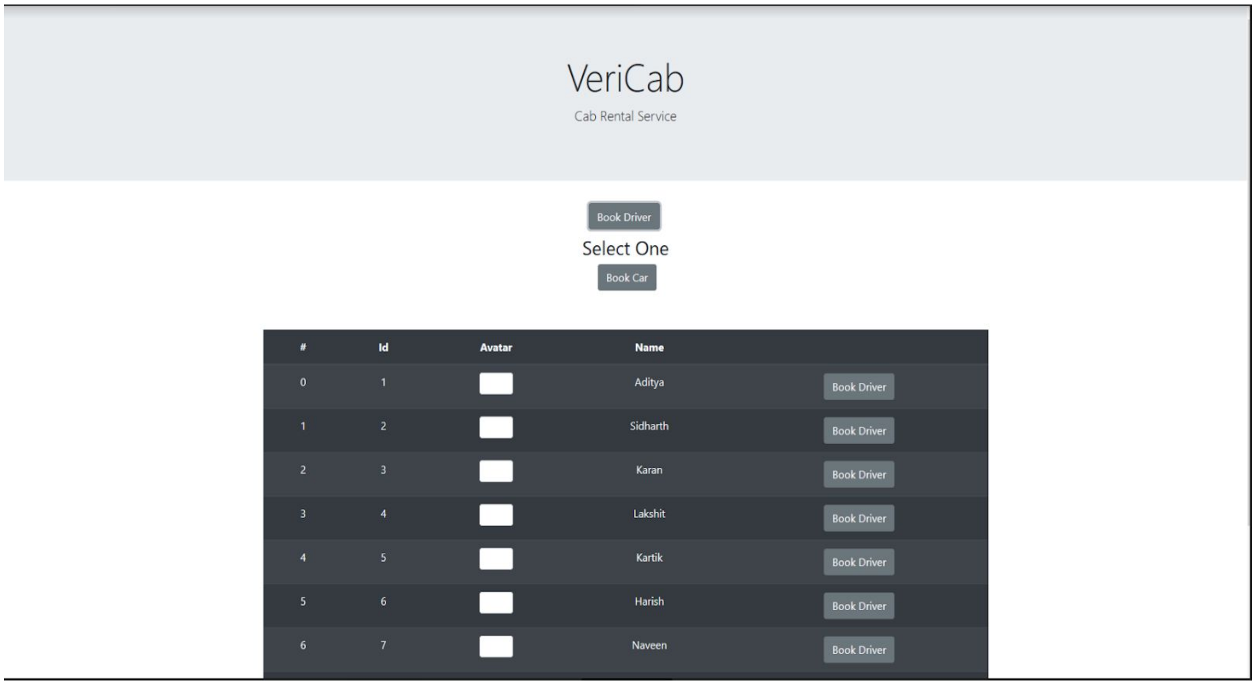
In this page you will get two options: Book a driver and Book a cab with a driver. You will get a list of cars and drivers from a driver service. Then it will create a trip by selecting any of the option using Trip Management Service. Then after completing the ride it will go into the Trip history database. You can click on Show History to view the history of all your trips using Trip Management Service. You can also view offers by clicking it on. You will also get notifications of offers from Notification Service. After trip being





VeriCab

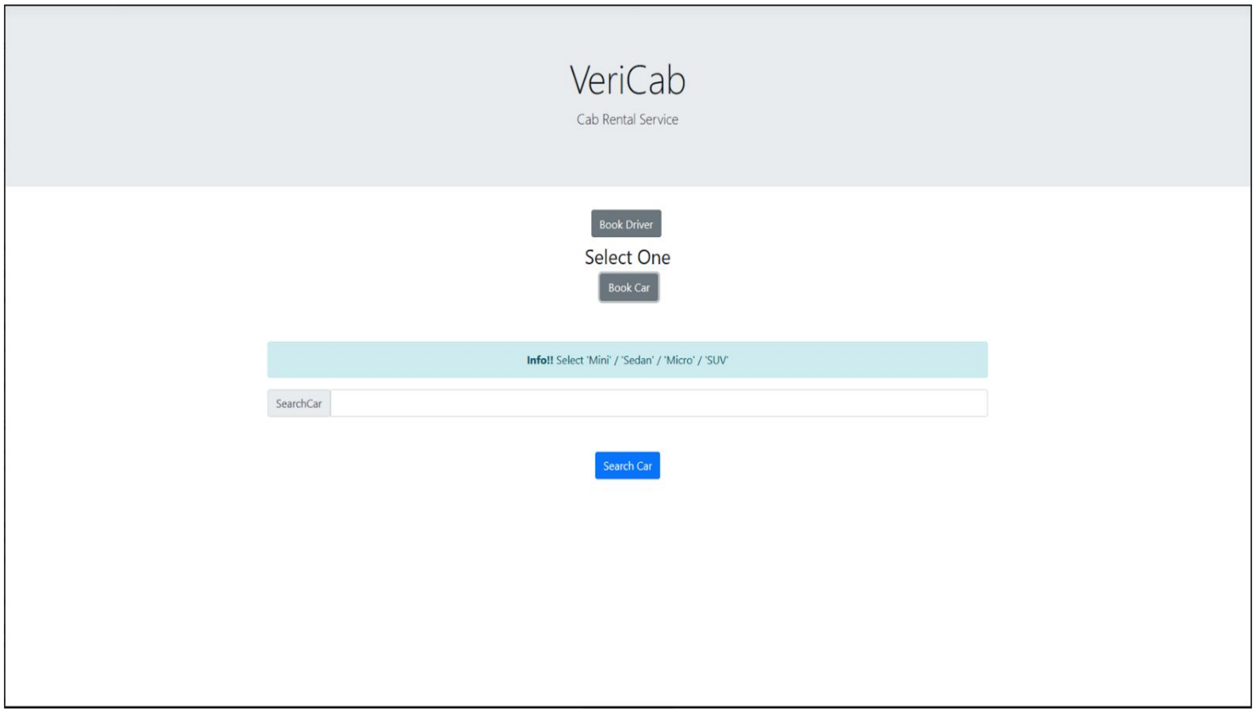
Cab Rental Service



VeriCab

Cab Rental Service

#	Id	Avatar	Name	
0	1		Aditya	Book Driver
1	2		Sidharth	Book Driver
2	3		Karan	Book Driver
3	4		Lakshit	Book Driver
4	5		Kartik	Book Driver
5	6		Harish	Book Driver
6	7		Naveen	Book Driver

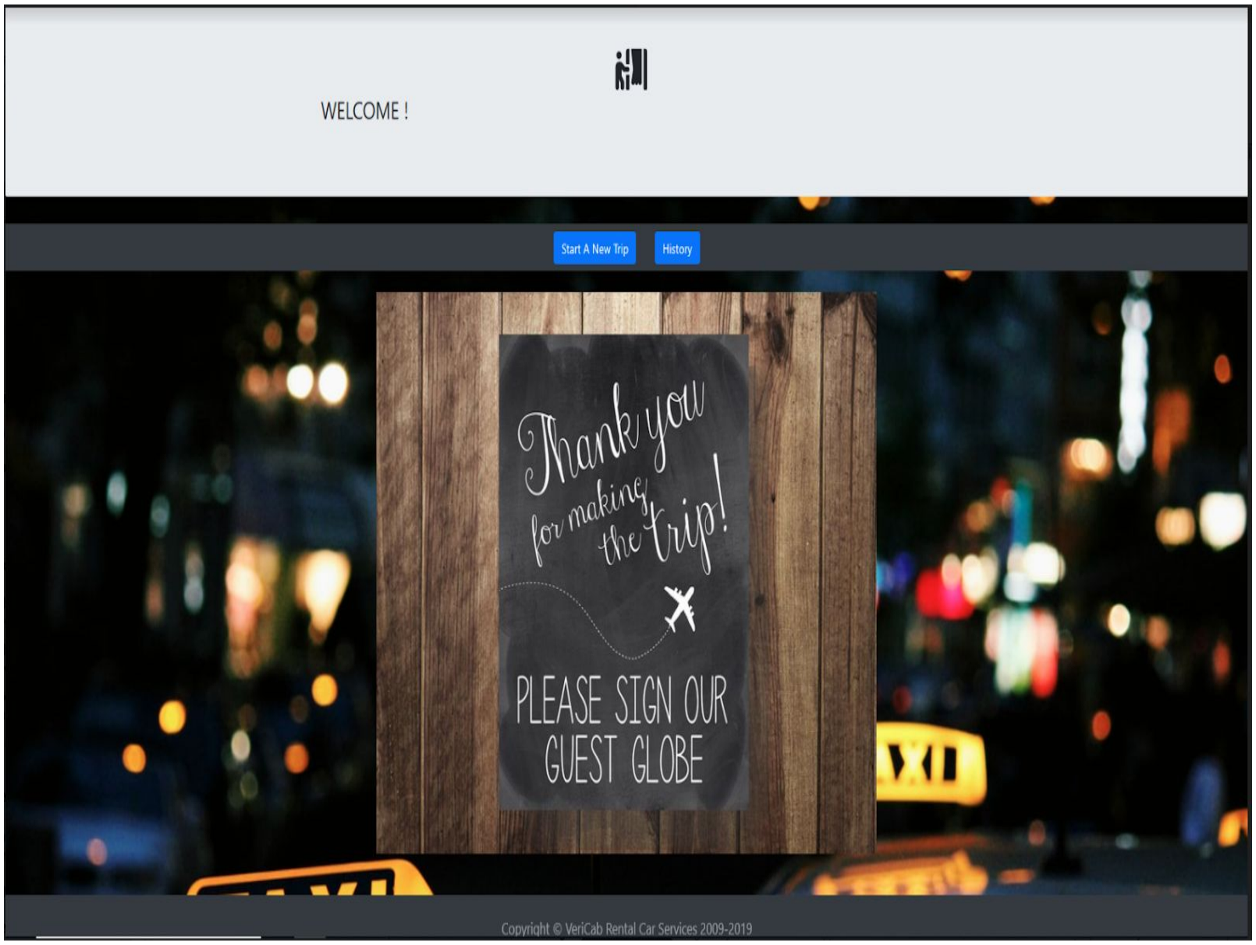

VeriCab

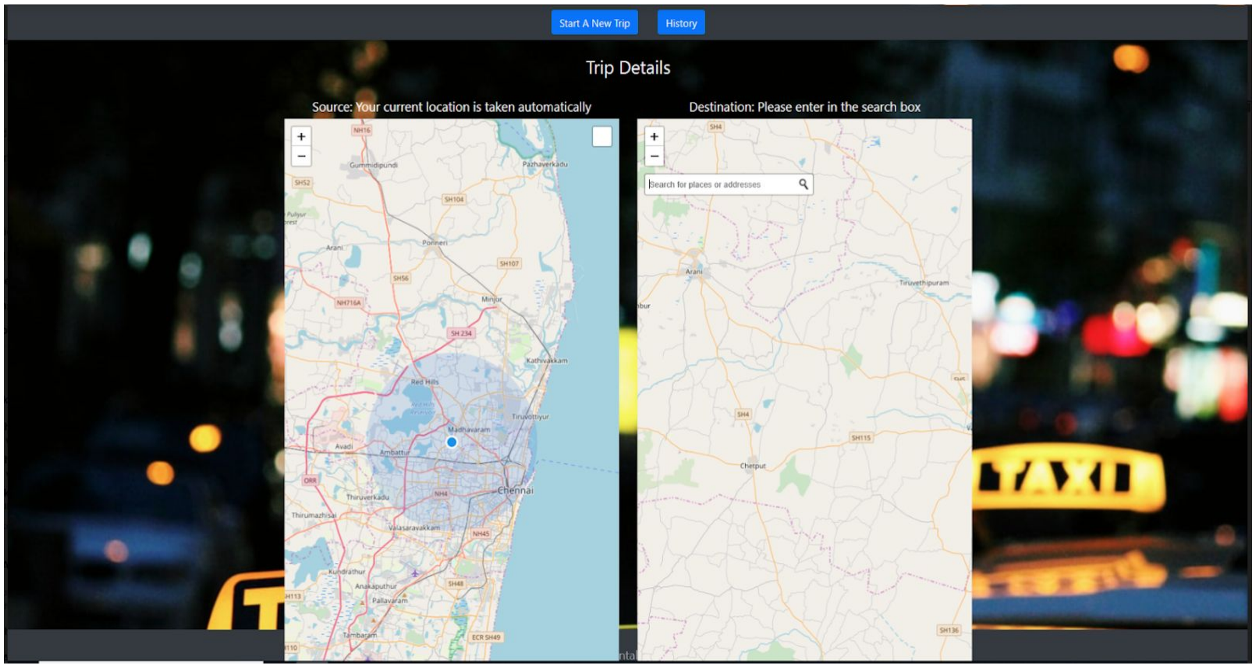
Cab Rental Service

Info!! Select 'Mini' / 'Sedan' / 'Micro' / 'SUV'

4.7 DRIVER PAGE

In Driver Page you will get to know the assignment to the passenger and the location of that passenger so that you can reach him/her out. Once you pickup the passenger you will get option to start the trip. Once the trip has started you will get the option to end the trip. If you click the end trip button then it will go into the history of all trips.

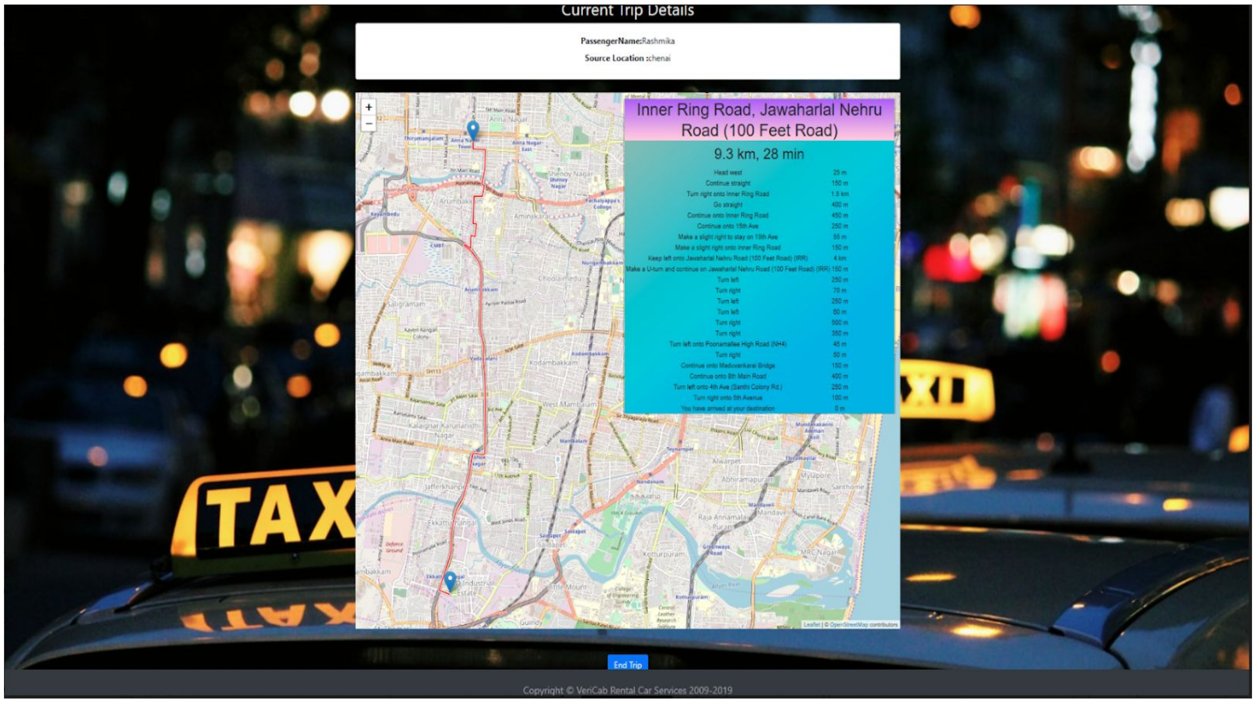

WELCOME !
Start A New Trip
History
Thank you for making the trip!
PLEASE SIGN OUR GUEST GLOBE
TAXI
Copyright © VeriCab Rental Car Services 2009-2019



Trip Details

Source: Your current location is taken automatically

Destination: Please enter in the search box



Current Trip Details

PassengerName: Rashmika

Source Location :chenai

Inner Ring Road, Jawaharlal Nehru Road (100 Feet Road)

9.3 km, 28 min

Head west	25 m
Continue straight	150 m
Turn right onto Inner Ring Road	1.5 km
Go straight	400 m
Continue onto Inner Ring Road	450 m
Continue onto 15th Ave	250 m
Make a slight right to stay on 15th Ave	55 m
Make a slight right onto Inner Ring Road	150 m
Keep left onto Jawaharlal Nehru Road (100 Feet Road) (IRR)	4 km
Make a U-turn and continue on Jawaharlal Nehru Road (100 Feet Road) (IRR)	150 m
Turn left	250 m
Turn right	70 m
Turn left	250 m
Turn left	80 m
Turn right	500 m
Turn right	350 m
Turn left onto Poonamallee High Road (NH4)	45 m
Turn right	50 m
Continue onto Maduvankarai Bridge	150 m
Continue onto 8th Main Road	400 m
Turn left onto 4th Ave (Santhi Colony Rd.)	250 m
Turn right onto 5th Avenue	100 m
You have arrived at your destination	0 m

Copyright © VeriCab Rental Car Services 2009-2019

4.8 PAYMENT PAGE

Payment Page comes when the driver clicks on the end trip but at the user end user will receive Payment Page using Payment Service to calculate the fair which has to be reflected on the screen of the user. But before getting Payment Amount it will verify using checkIfTripHasCompleted, if is returned as true then it will show the amount otherwise it will show the payment has failed.

5.0 DATA MANAGEMENT

5.1 DATA DESCRIPTION

The Database consists of :

- **Offers**: In offers admin can add and remove offers.
- **Passenger**: In Passenger table we can add, view, delete and update passenger details.
- **Ongoing Trip**: In this we can see the details of ongoing trips.
- **History**: In this we can see the details of history of trips of passengers and drivers.
- **Authentication**:In this table you we will have the user-name and passwords.
- **Driver**: In this table we will have the details of all drivers.

5.2 DATA OBJECTS

➔ **Offers**: offerName, offerCode, validity, offerMessage.

➔ **Passenger**: passengerId, passengerName, passengerGender, phoneNumber, passengerEmail.

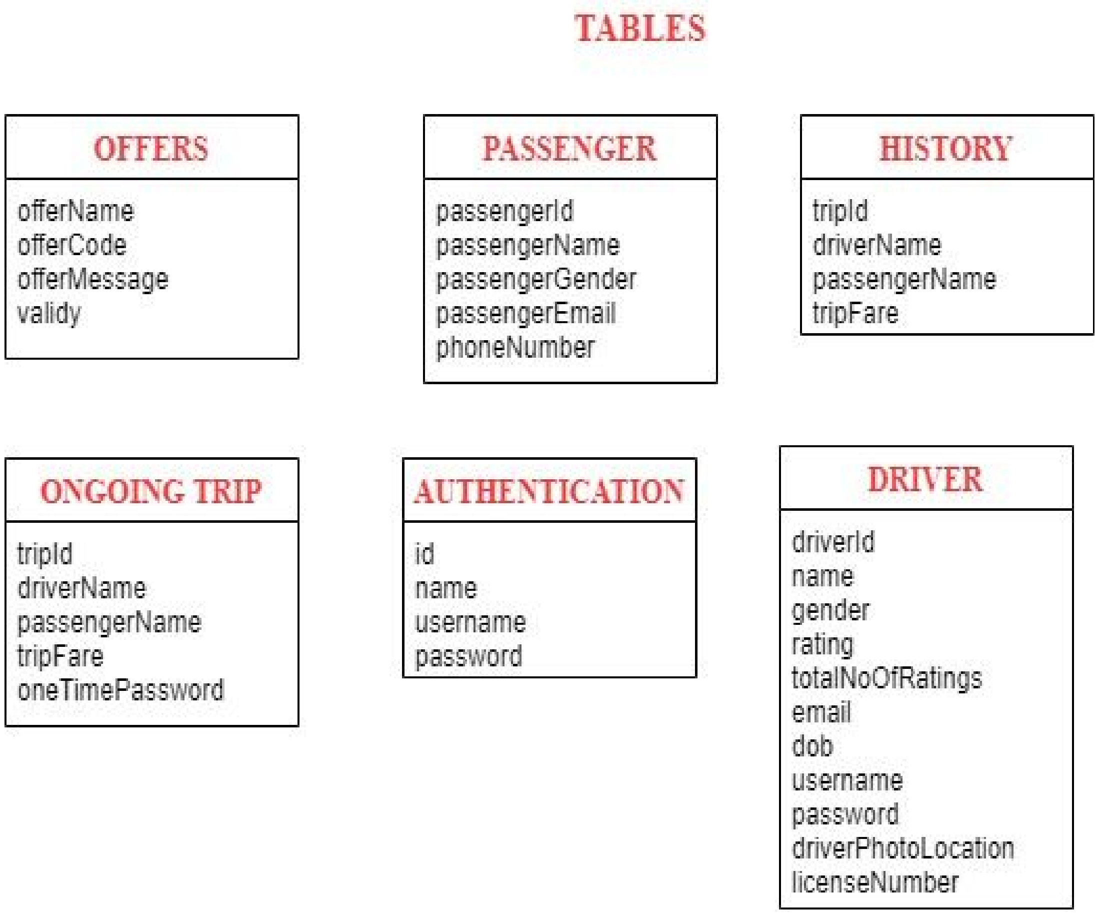
➔ **Ongoing Trip**: tripId, driverName, passengerName, tripFare, oneTimePassword

➔ **History**: tripId, driverName, passengerName, tripFare.

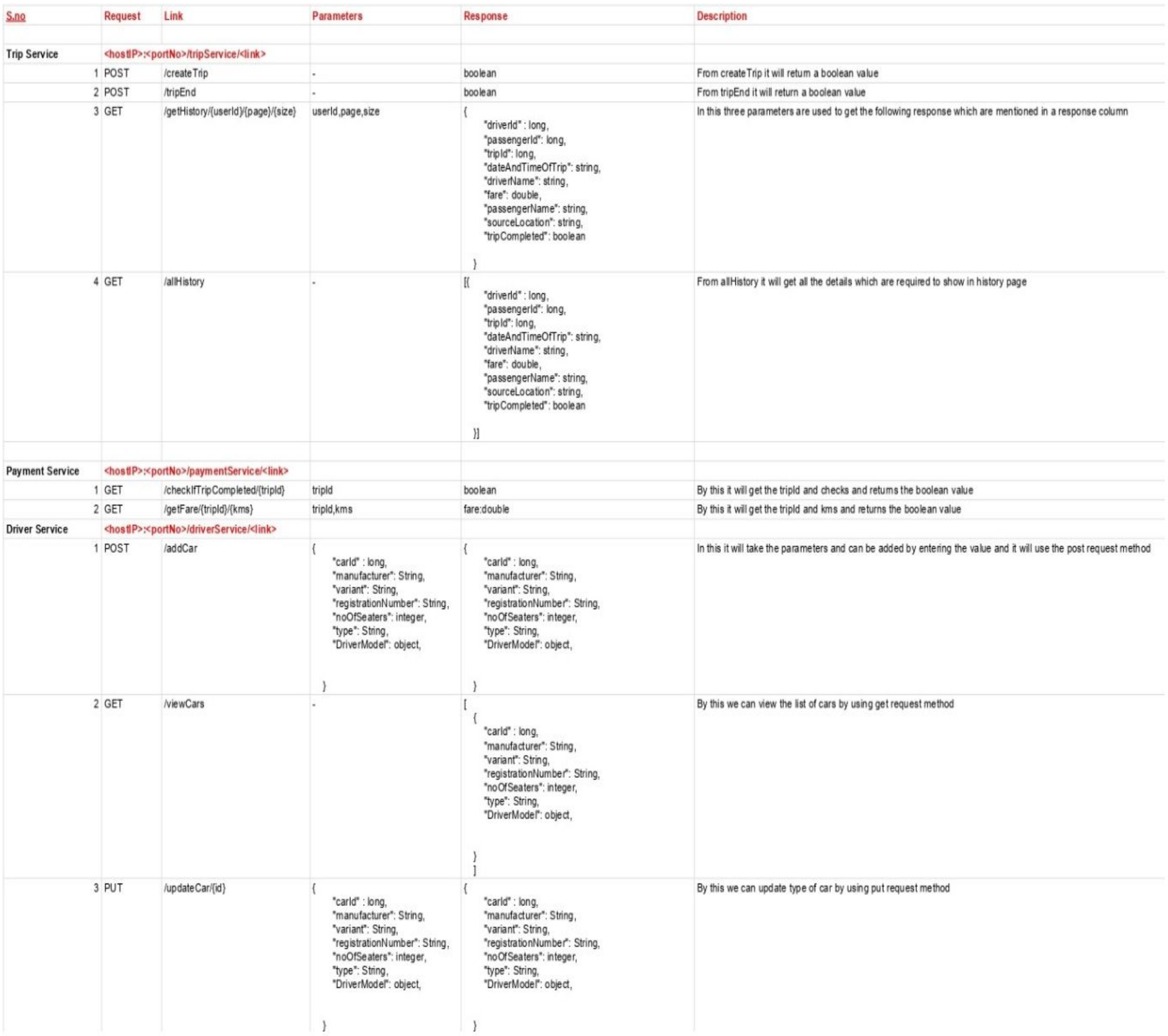
➔ **Authentication**: id, name, username, password.

➔ **Driver:** driverId, name, gender, rating, totalNoOfRatings, email, dob, username, password, driverPhotoLocation, licenseNumber.

5.3 DATABASE TABLE DIAGRAM



6.0 REST API CALLS AND END POINTS



S.no	Request	Link	Parameters	Response	Description
Trip Service	<hostIP>:<portNo>/tripService/<link>				
1	POST	/createTrip	-	boolean	From createTrip it will return a boolean value
2	POST	/tripEnd	-	boolean	From tripEnd it will return a boolean value
3	GET	/getHistory/{userId}/{page}/{size}	userId,page,size	{ "driverId" : long, "passengerId": long, "tripId": long, "dateAndTimeOfTrip": string, "driverName": string, "fare": double, "passengerName": string, "sourceLocation": string, "tripCompleted": boolean }	In this three parameters are used to get the following response which are mentioned in a response column
4	GET	/allHistory	-	[{ "driverId" : long, "passengerId": long, "tripId": long, "dateAndTimeOfTrip": string, "driverName": string, "fare": double, "passengerName": string, "sourceLocation": string, "tripCompleted": boolean }]	From allHistory it will get all the details which are required to show in history page
Payment Service	<hostIP>:<portNo>/paymentService/<link>				
1	GET	/checkIfTripCompleted/{tripId}	tripId	boolean	By this it will get the tripId and checks and returns the boolean value
2	GET	/getFare/{tripId}/{kms}	tripId,kms	fare:double	By this it will get the tripId and kms and returns the boolean value
Driver Service	<hostIP>:<portNo>/driverService/<link>				
1	POST	/addCar	{ "carId" : long, "manufacturer": String, "variant": String, "registrationNumber": String, "noOfSeaters": integer, "type": String, "DriverModel": object, }	{ "carId" : long, "manufacturer": String, "variant": String, "registrationNumber": String, "noOfSeaters": integer, "type": String, "DriverModel": object, }	In this it will take the parameters and can be added by entering the value and it will use the post request method
2	GET	/viewCars	-	[{ "carId" : long, "manufacturer": String, "variant": String, "registrationNumber": String, "noOfSeaters": integer, "type": String, "DriverModel": object, }]	By this we can view the list of cars by using get request method
3	PUT	/updateCar/{id}	{ "carId" : long, "manufacturer": String, "variant": String, "registrationNumber": String, "noOfSeaters": integer, "type": String, "DriverModel": object, }	{ "carId" : long, "manufacturer": String, "variant": String, "registrationNumber": String, "noOfSeaters": integer, "type": String, "DriverModel": object, }	By this we can update type of car by using put request method

4	GET	/viewCar/{carId}	carId	{
				"carId" : long,
				"manufacturer": String,
				"variant": String,
				"registrationNumber": String,
				"noOfSeaters": integer,
				"type": String,
				"DriverModel": object,
				}
					By this we can view the list of cars with carId parameter by using get request method

5	GET	/viewByType/{type}	type	[
				{
				"carId" : long,
				"manufacturer": String,
				"variant": String,
				"registrationNumber": String,
				"noOfSeaters": integer,
				"type": String,
				"DriverModel": object,
				}
				]
					By this we can view the car type with type parameter by using get request method

6	DELETE	/deleteCar/{id}	id	string	It is used to delete car by using id as a parameter

7	POST	/viewDrivers	-	{
				"driverId" : Long,
				"name": String,
				"gender": String,
				"rating": Double,
				"totalNoOfRatings": Long,
				"email": String,
				"phoneNumber": String,
				"Date" : Date,
				"password" : String,
				"driverPhotoLocation":String,
				"LicenseNumber":String
				}
					It is to view all the list of drivers by using post request method

Passenger Service	<hostIP>:<portNo>/passengerService/<link>				

1	POST	/addPassenger	-	{
				"passengerId" : long,
				"passengerName": String,
				"passengerGender": String,
				"passengerPhoneNumber": long,
				"passengerEmail": String,
				}
					It is used to view and add passenger by using post request method

2	PUT	/updatePassenger/{id}	{
			"passengerId" : long,
			"passengerName": String,
			"passengerGender": String,
			"passengerPhoneNumber": lon
			"passengerEmail": String,
			}
				{
				"passengerId" : long,
				"passengerName": String,
				"passengerGender": String,
				"passengerPhoneNumber": long,
				"passengerEmail": String,
				}
					It is used to update passenger with parameter id by using post request method

3	DELETE	/deletePassenger/{id}	id	String	It is used to delete passenger with parameter id using delete request method

4	GET	/viewPassengers	-	{
				"passengerId" : long,
				"passengerName": String,
				"passengerGender": String,
				"passengerPhoneNumber": long,
				"passengerEmail": String,
				}
					It is used to view passengers list by using get request method

7.0 NON-FUNCTIONAL / OPERATIONAL REQUIREMENTS

7.1 SECURITY

- ➢ Pages of the website must be accessed in the way they were intended to be accessed.
- ➢ Included files shall not be accessed outside of their parent file.
- ➢ Administrator can only perform administrative task on pages they are privileged to access.
- ➢ Customers will not be allowed to access the administrator pages.
- ➢ OTP should not be shared with other drivers.
- ➢ Payment method may require your card details.

7.2 EFFICIENCY AND MAINTAINABILITY

- ➢ Page loads should be returned and formatted in a timely fashion depending on the request being made.
- ➢ Administrators will have the ability to edit the aspects of the drivers, driver descriptions, prices, offers and website directly

8.0 REFERENCES

1. *https://redux.js.org/basics/basic-tutorial*
2. *https://react-redux.js.org/introduction/basic-tutorial*
3. *https://www.javatpoint.com/spring-boot-tutorial*
4. *https://www.tutorialspoint.com/spring_boot/index.htm*
5. *https://www.tutorialspoint.com/spring_boot/spring_boot_eureka_server.htm*
6. *https://www.tutorialspoint.com/spring_boot/spring_boot_hystrix.htm*
7. *https://www.tutorialspoint.com/cassandra/index.htm*
8. *https://www.javatpoint.com/junit-tutorial*
9. *https://www.javatpoint.com/selenium-tutorial*
10. *https://www.tutorialspoint.com/uml/index.htm*