

## Project Report on

# ANDROID BASED CAR-POOLING APPLICATION

**NEERAJ AJEET PAL**

(2106849)

*Project Partner*

**NEEL JAYANTIBHAI PATEL**

(2106857)

*Guided By*

**Mrs. PRAJAKTA KAMBLE**



**K.J. SOMAIYA COLLEGE OF SCIENCE AND COMMERCE**



**K.J.Somaiya College of Science and Commerce**  
**Vidyavihar, Mumbai-400 077**  
**Autonomous-Affiliated to University of Mumbai**  
**Department of information technology**



## CERTIFICATE

*Certified that the experimental work as entered in this journal is as per syllabus in **B.Sc. Information Technology for Software Project Management** as prescribed by University of Mumbai and was done in the Information Technology laboratory of **K.J.Somaiya College of Science and Commerce** by the student MR. NEERAJ AJEET PAL having Seat No. 2106849 Of class **T.Y.B.Sc. Information Technology** during the academic year 2021-22.*

Sign of in charge:

Date:

Sign of Examiner:

Date:

Course Co-Ordinator

## **Acknowledgment**

In the present world of competition there is a race of existence in which those are having will come forward succeed. Project is like a bridge between theoretical and practical working.

We express our deep gratitude and appreciation to those who agreed to participate in this project, for their time expended and courage in sharing their insights with the fledging students. It is to them that we are most indebted, and we can only hope that the product of our collaboration benefits each one as much as we are benefitted from process.

We have been immeasurably enriched by working under the supervision of Mrs. Prajakta Kamble, the subject teacher, who has a great knowledge and who has an art of encouraging, correcting and directing us in every situation possible, which has enabled us to complete our project.

At times, our studies carried out at great cost to those closest to us. We thank our family and our fellow students for their best understanding and support. We acknowledge to all the people who have involved and supported us in making this project.

# TABLE OF CONTENT

## INTRODUCTION

1.1	Abstract.....	07
1.2	Objectives.....	08
1.3	• Purposes.....	09
	• Scope.....	09
	• Applicability (Feasibility Study) .....	09
1.4	Achievements (If any) .....	10

## SURVEY OF TECHNOLOGIES

2.1	JavaScript.....	12
	Node.JS.....	12

## REQUIREMENTS AND ANALYSIS

3.1	Problem Definition.....	14
3.2	Requirements Specification.....	15
	▪ Functional Requirements.....	15
	▪ Non-Functional Requirements.....	16
3.3	Software and Hardware Requirements.....	17
3.4	Preliminary Products Description.....	21

## SYSTEM STUDY

4.1	Existing System.....	23
4.2	Proposed System.....	24
4.3	Feasibility Study .....	25
4.4	Economic Feasibility.....	25
4.5	Technical Feasibility.....	26
4.6	Operational Feasibility.....	27

## SYSTEM ANALYSIS

5.1	Preliminary Investigation.....	29
▪	Project Clarification.....	29
▪	Feasibility Study.....	29
▪	Technical Feasibility.....	30
5.2	Planning & Scheduling .....	30
5.3	Gathering Information.....	31

## SYSTEM DESIGN

6.1	Introduction to System Design.....	33
6.2	Basics Modules.....	35
6.3	System Diagrams.....	36
▪	E-R Diagram.....	36
▪	Use Case Diagram.....	37
▪	Class Diagram.....	38
▪	Activity Diagram.....	39

## Project SDLC Model

7.1	Spiral Model.....	42
-----	-------------------	----

## IMPLEMENTATION & TESTING

8.1	Implementation Approaches.....	50
8.2	Code Efficiency.....	51
8.3	Testing Approaches .....	55
8.4	Modification and improvements.....	58

## RESULTS & DISCUSSION

9.1	Test Case.....	65
9.2	User Documentation.....	71

## CONCLUSION

10.1	Conclusion.....	74
10.2	Advantages & Limitation.....	75
10.3	Future Scope of the Project.....	76
10.4	Links.....	77

## REFERENCES

# CHAPTER 1



## **1.1. Abstract:**

The aim of this project is to provide such user friendly and interactive service via Application User interface. This system provides a simple and quicker interaction for the users who are in need of transportation help. The system will overcome most of the drawbacks of existing system problem and will provide simple way to get the convey. The main objective is to improve the data communication within the Driver and the traveler to reduce convey time for each trip and maintain the efficiency. The proposed system will help enhancing the communication among Travel partners within the network. This Application is specially designed for Car Pooling/Ride Share. For e.g.: If a person/student is not having a public transport or a personal vehicle to travel then the user can search in their respective locality, if someone is going to the same destination and share a trip/ride with the person having a private vehicle. Same way the person with the vehicle, willing to share his/her ride can search for a ride partner in Drivers app.

## **Project Overview**

The person will be creating his account on the Rider's App so that he/she can request a ride anytime he/she need. The same way the Driver can register themselves on Driver's app so that they can share the ride with those whom they feel comfortable & friendly. Passenger will be asked some information about them so that the proper safety & support can be provided and there should not be any misinterpretation.

## **1.2 Objective:**

- Carpooling is the concept of sharing your car to accommodate more than one person at a time, eliminating the need for riders to drive themselves in separate vehicles.
- Drivers and passengers offer and search for journeys through one of the several mediums available. After finding a match they contact each other to arrange any details for the journey(s). Costs, meeting points and other details like space for luggage are agreed on. They then meet and carry out their shared car journey(s) as planned.
- Carpooling is commonly implemented for commuting but is increasingly popular for longer one-off journeys, with the formality and regularity of arrangements varying between schemes and journeys.
- Carpooling is not always arranged for the whole length of a journey. Especially on long journeys, it is common for passengers to only join for parts of the journey, and give a contribution based on the distance that they travel. This gives carpooling extra flexibility and enables more people to share journeys and save money.
- Carpooling for Students is not implemented still, but our objective is to help students staying in same locality to travel together and save fuel as well as money and ensure complete safety to an individual. This application can also be used by teachers/ staff members.
- Carpooling usually means to divide the travel expenses equally between all the occupants of the vehicle (driver or passenger). The driver doesn't try to earn money, but to share with several people the cost of a trip he/she would do anyway.



## **1.3 Purpose, Scope and Applicability:**

### **1.3.1 Purpose:**

Carpooling (also car-sharing, ride-sharing and lift sharing) is the sharing of car journeys so that more than one person travels in a car, and prevents the need for others to have to drive to a location themselves. The main purpose of this Application is to provide needy people with convenient way of transport.

### **1.3.2 Scope:**

Carpooling system is very effective means to reduce pollution and the congestion of vehicles in cities. It also provides an eco-friendly way to travel. It also provides an opportunity to meet new people. As today most people prefer private vehicle to travel due to delay caused in public transport system and luxuries provided by private vehicles. Pre-registration ensures security, as only identified people get into the vehicle so that trust can be established. The people registered can be allotted specific days on which they should take their private vehicle, so that no inconvenience is caused to its registered passengers for daily commute. Thus the proposed carpooling system will be effective in reducing environment pollution.

### **1.3.3 Applicability:**

Carpooling application can be used by any user or passenger who have or does not have his own car and wants to travel only by a private vehicle. The user will be able to filter in accordance with his destination and middle point (if there are two different routes to reach some destination middle point will help us to filter it precisely). The user will select for a trip or can manually search his source and destination and will upload a trip to our application.

## 1.4 Achievements:

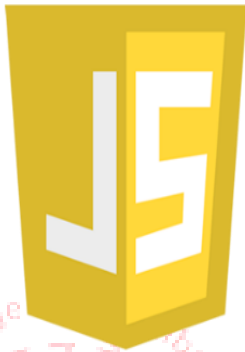
- The main advantage of rideshare solutions is cost saving, of course. Depending on the agreement, a ride can be twice as cheap than traveling by conventional way.
- The car is not a gas cost only. There are some cost items for the maintenance, repair, parts replacement in your vehicle. If you reduce the time of car utilization, you reduce these costs.
- A fewer number of cars on the road can reduce the CO2 emission in the roads and make the air we breathe cleaner.
- Saving time. Fewer cars - fewer traffic jams. That is to say, it is possible to reach a destination point faster and find a parking place.
- Meeting new people, traveling together allows you to find good friends.

# CHAPTER 2



Survey Of different technologies was done for the development of this project. Technology such JavaScript & Node.JS as were referred in order to develop this project.

## JAVASCRIPT



JavaScript is a dynamic programming language that is mainly used for client-side scripts and web development. Lately, programmers have been able to implement standalone application servers using this technology. A famous JavaScript solution is Node.JS

## NODE.JS



Node.js, which is event driven, can delegate processes (especially I/O operations) to other components which gives it time to process other requests. It is also single threaded so it doesn't take advantage of high-performance CPUs that's why scaling-out is preferred when thinking about scaling.

# CHAPTER 3



### **3.1 Problem Definition:**

Today increase of environmental concerns and the congestion of roads, carpooling has gained a lot of popularity when it comes to environment friendly and cheap ways of travelling. Carpooling is when two or more persons share a ride in one of their personal cars. Carpooling reduces pollution since we have less cars on the road. It's also economic since the travel expenses are shared among the riders. Travelling alone may be stressful, so having other persons with you on a trip reduces the stress and is also the occasion to socialize and make the trip funnier. As said, the expenses, both environmental and fiscal, of single occupancy vehicles could be reduced by utilizing the empty seats in personal transportation vehicles. Carpooling and ridesharing target those empty seats: taking additional vehicles off the road reducing traffic and pollution, whilst providing opportunities for social interaction. However, historically carpool scheduling often limited users to consistent schedules and fixed rider groups—carpooling to the same place at the same time with a set person or a group of people. To make that problem worse, the leading problem concerns, given in a 2009 survey about why people don't carpool, were difficulty to organize carpools and inconvenience of organization. We feel both of those can be addressed by employing some novel web technologies and modern-day available data stores which hold social and location based individual user's data. Besides having to solve the aforementioned problems for making a carpooling and ride-sharing solution that users will want to use.

## **3.2: Requirements Specification:**

### **• Functional Requirements:**

#### **-All users can:**

- Create an account.
- Login.
- View Profiles.
- Modify their profiles info (phone, email, etc.).

#### **-After identification drivers can:**

- Submit a ride with specifications.
- Cancel a ride while notifying passengers.
- Modify a ride (date, number of seats, target, start point, etc.).
- Accept or decline a ride request from a passenger.

#### **-In addition, passengers can:**

- Search for a ride.
- View available rides on the map.
- Reserve a ride from a driver.
- Request a ride.
- Rate/Comment on a driver after a trip.
- Report drivers.

- **Non-Functional Requirements:**

- Software extensibility
- Ability to add or modify some features
- Multi-Platform Support (Smartphones, Tablets).
- Implementing an online backend for easier management (AWS).
- Small Application size.
- Fast response from server.
- Secure system.
- Simplicity of interface.
- Development using Android Studio's Java/Kotlin.
- Clear Information and Communications.
- Google Maps API + Geo-localization.
- Clear Privacy Policy.



### **3.3: Hardware & Software Requirements:**

- **Hardware Specifications: (Server Side)**

For the development of this application, we have used a Desktop-PC, it fairly powerful and could handle the usage of multiple emulators at once.

#### **Full Specifications:**

- Processor: Intel Core i3-10100F @ 3.60GHz
- RAM: 8.00GB 2400 MHz DDR4
- Graphics: NVIDIA GeForce GT 710
- HDD Drive: 1TB
- SSD Drive: 128GB

For the testing of the application on a real device we have used an Redmi Note 7S & Realme X android devices.

## ■ Hardware Specifications: (Client Side)

To run the Application, the user must have these software specifications.

### Minimum Requirements:

- Processor: 1.25 GHz
- RAM: 2GB
- Storage: 500MB

### Recommended Requirements:

- Processor: 2.00 GHz
- RAM: 4GB
- Storage: 1GB

## ● Software Specifications: (Server Side)

For the development of this application, we have used a Desktop-PC that can handle the usage of multiple emulators at once.

### Full Specifications:

- Operating System: Windows 11 Pro
- Runtime Environment: React Native
- Web Server: JavaScript Server
- IDE: Android Studio 4.2.2 & WebStorm 2021.1.3
- Emulators: Google pixel devices
- Backend Management: AWS Services
- Places and Map Tools: Google Cloud Platform

## ■ Software Specifications: (Client Side)

### **Minimum Requirements:**

- Operating System: Android
- Android Version: Ice-Cream Sandwich v4.0

### **Recommended Requirements:**

- Operating System: Android
- Android Version: Android Q v10.0:

### **3.4: Preliminary Product Description:**

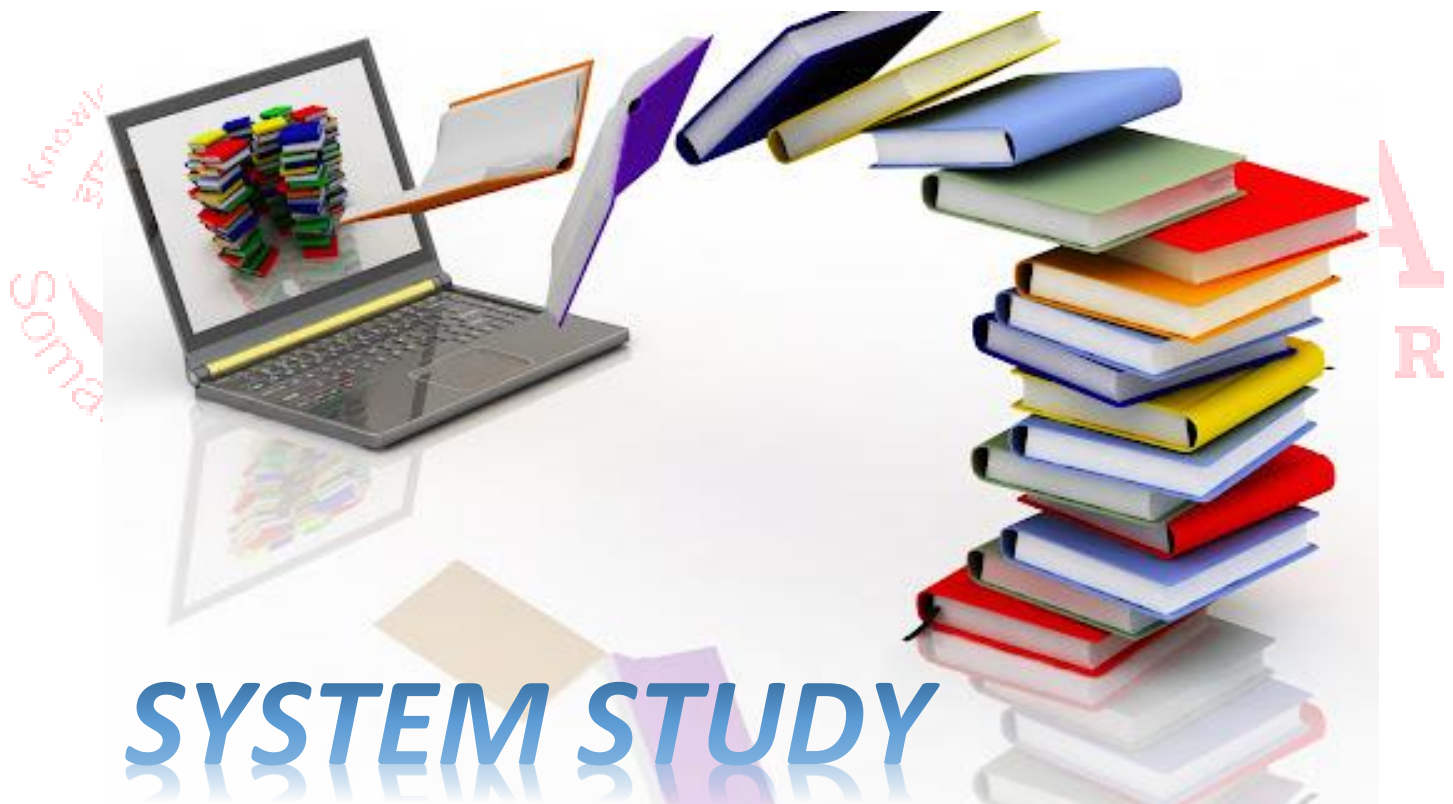
This project (RideIT) aims to develop an Android based application for carpooling, this application allows nonprofessional drivers to submit rides for specific targets and allows passengers to reserve/request rides from drivers all while being secure and having a simple interface.

This application will help people save money and also reduce the pollution of the environment and effects of vehicles; this application focuses on serving needs of students that may have difficulties while travelling to the university/school/colleges.

RideIT will support Android phones and Tablets, Users will need internet connection to use the application to offer or find a common route to travel to.

The application will have a simple and easy interface, Users must register at first before using the application, after that they must choose between a driver or a passenger, a driver can offer a drive to a specific location while a passenger can find or request a ride to a location.

# CHAPTER 4



## *SYSTEM STUDY*

## **4.1 Existing System:**

Many carpool service systems have been proposed which can be divided into two broad categories based on their features. The first of these comprises systems which are web-based and which transmit carpool information to an online community platform. One such system is Carpool Global which supplies an interfacing service for willing drivers and passengers. These Systems do not include Location Display of user (GIS) & not real-time. Many carpool systems have been developed to decrease lessen traffic congestion. Many systems supply simple carpooling functions including the option to send requests for a specified date and time, and search for applicable users. In addition, several systems feature a digital GIS mapping ability by which to provide a visual tool with accurate location information to users. Unfortunately, these systems are neither efficient nor convenient for users who need real-time carpool matches. Our proposed system incorporates mobile communication technology with GIS to create a carpool service which is operable in real time. Subsequently, users can instantly submit carpool requests to the intelligent carpool system which reflect their current locations via the use of smart, handheld, communication devices which feature GPS capabilities. The system will use the carpool matching algorithm to generate and return match results within a short amount of time.

## **4.2 Proposed System:**

Carpooling system is dynamic system which based on two underlying sources of information which includes routes announcement by the user, route selection and registration by customers. The user who is going to travel by his/her car will mention source and destination along with the route which is selected his/her. He will also mention the capacity of vehicle user who find the path as per his request can register for the trip. Carpooling system has detail phased registration system for displaying routes and user positions we used GOOGLE maps. Addition thing we are using payment mode system for the digital payment. We will send you car details and estimate cost of fair. The two kinds of trips are single, which are trip between two cities and frequent trips which are given commuters do every day the application is design to be scalable, extensible, high available and good performance.



### **4.3: Feasibility Study:**

The system Preliminary investigation examines project feasibility, the likelihood the system will be useful to the organization. The main objective is to make sure the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system can be run effectively and smoothly with required resources and time. Some aspects listed down

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

### **4.4: Economic Feasibility:**

The system must be good investment for organization but also must not cause problem to its users. The system should handle the organization economically and maintain the relation with users.

#### **4.5: Technical Feasibility:**

The technical issue can be occurred during the development.

- What technology to be used for smooth operations?
- Do the proposed equipment's are capable of maintaining the data and run the operations.?
- Will the proposed system be able to solve the queries raised by users?
- What challenges are there in System upgradation?
- Data security challenges?

The current system developed is technically strong and easy. Thus, it provides an easy access to the users with simple GUI.

The database's purpose is to create, establish and maintain a workflow among various entities and maintain the role of every user. Permissions are granted based on the roles specified providing technical guarantee of accuracy, reliability and security.

The software and hardware requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast

feedback to the users irrespective of the number of users using the system.

#### **4.6: Operational Feasibility:**

Proposed projects are turned out into information system meeting the organization's operating requirements.

Operational feasibility aspects of the project are important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- Is there sufficient support available for management from the users?
- Will the system work as expected and provide the output that we need?
- Will there be any negative feedback from the user and complaints about the system?

# CHAPTER 5



## **5.1 Preliminary Investigation:**

Preliminary Investigation is conducted in the following phases.

- Project clarification
- Feasibility study
- Project appraisal

### **Project Clarification:**

When a system development or modification request is made, the first systems activity or the preliminary investigation has three parts: Request clarification, feasibility study and project appraisal. Many requests from employees and users in organization are not clearly stated. Therefore, before any systems investigation can be considered, the project request must be examined to determine precisely what the originator wants. This is called Project clarification.

The important outcome of the preliminary investigation is the determination that the system request is feasible.

### **Feasibility Study:**

The feasibility study is performed to determine whether the proposed system is viable considering the Technical, Operational and Economical factors. After feasibility study we can have a clear-cut view of the system's benefits and drawbacks.

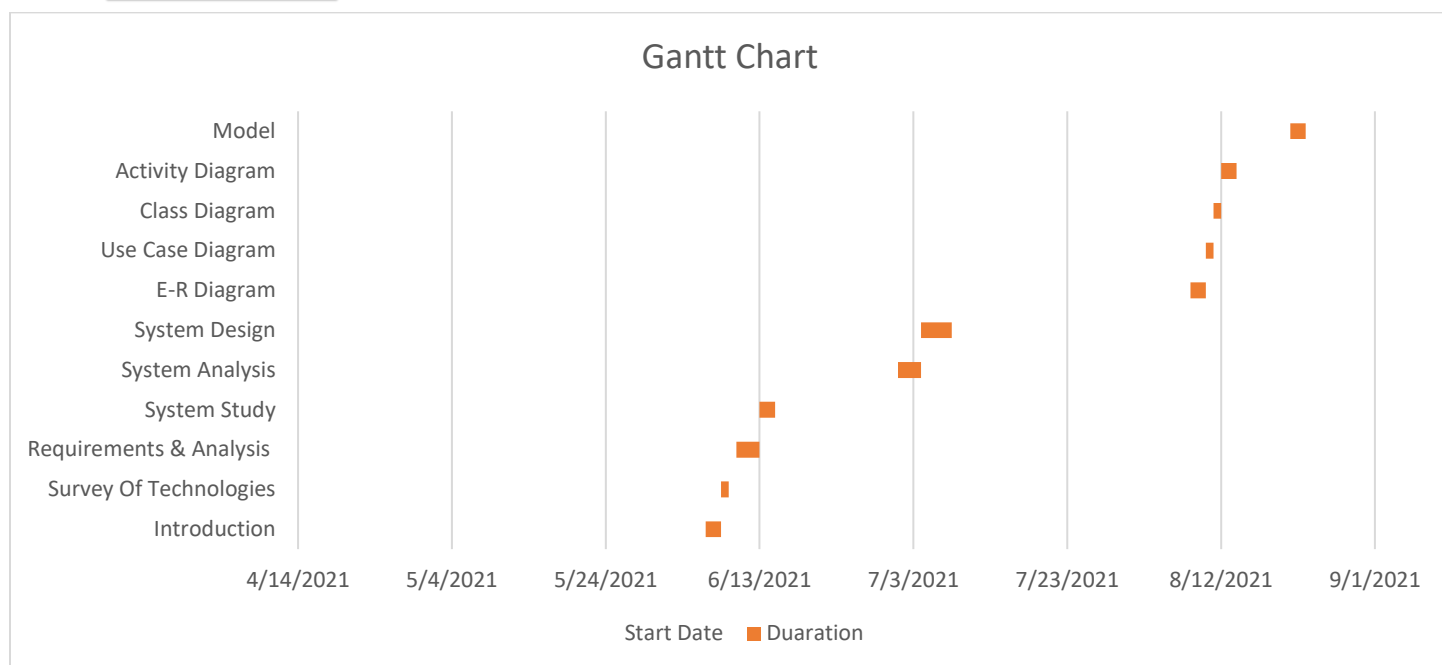
## Technical Feasibility:

The proposed system is developed using JavaScript & Node.JS as front-end tool and AWS as the back-end service. The data of the user is stored in a server named AWS. Amazon Web Services (AWS) is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis

## 5.2 Planning & Scheduling:

<u>Tasks</u>	<u>Assigned To</u>	<u>Progress</u>	<u>Date</u>	
			<u>Start</u>	<u>End</u>
<u>Introduction</u>	Neel	100%	06/06/21	08/06/21
<u>Survey Of Technologies</u>	Neeraj	100%	08/06/21	09/06/21
<u>Requirements &amp; Analysis</u>	Neeraj	100%	10/06/21	13/06/21
<u>System Study</u>	Neel	100%	13/06/21	15/06/21
<u>System Analysis</u>	Neeraj	100%	01/07/21	04/07/21
<u>System Design</u>	Neel	100%	04/07/21	08/07/21
<u>E-R Diagram</u>	Neel	100%	08/08/21	10/08/21
<u>Use Case Diagram</u>	Neeraj	100%	10/08/21	11/08/21
<u>Class Diagram</u>	Neel	100%	11/08/21	12/08/21
<u>Activity Diagram</u>	Neeraj	100%	12/08/21	14/08/21
<u>Model</u>	Neel	100%	21/08/21	23/08/21

## ❖ Gantt Chart



## 5.3 Gathering Information:

The analysis done through collection of data plays the wider and big role in the analysis of the system. So, the data collected at different levels of management needed to keep track of full information of the system.

The collection of data is done from the interviews on two levels.

**Formal Online Session:** One to One Session with peoples who were travelling during this pandemic were conducted.

**Informal Group Session:** Information were collected from social media like Facebook, Instagram, WhatsApp, etc.

# CHAPTER 6

## SYSTEM DESIGN





## 6.1 Introduction to System Design

System design defines the process of planning a new system or to replace the disadvantages of existing system. Simply, system design is like the blueprint for building, it specifies all the features that are to be in the finished product.

System design phase follows system analysis phase. Design encompasses identifying functions, data streams among those functions, maintaining a record of the design decisions and providing a blueprint the implementation phase.

Some essential fundamental concepts involved in the design of application software are:

- Abstraction
- Modularity
- Verification

**Abstraction** means details of the various component are encrypted. Abstraction allows system designer to make step-wise refinement, which at each stage of the design may hide, unnecessary details associated with representation or implementation from the surrounding environment.

**Modularity** involves of main module into well-defined manageable units with well-defined interfaces among the units. This enhances design clarity, which in turn cases implementation, Verification, fundamental concept in software design.

Some of the important factors of quality that are need to be considered in the design of application software are:

**Reliability:**

The software should give expected results.

**Extensibility:**

The software should be capable of adapting new changes.

**Reusability:**

The software should be developed using a modular approach, which permits modules to be reused by other application, if possible.

**Preliminary Design:**

Preliminary design is basically concerned with deriving an overall picture of the system. Deriving entire system into modules and sub-modules while keeping Cohesion and Coupling factors in mind. Tools, like STAR UML which helps designing ER, Data Flow Diagrams, etc.

## **6.2 Basics Modules**

**There are two basic modules:**

### **1. User's Application Module:**

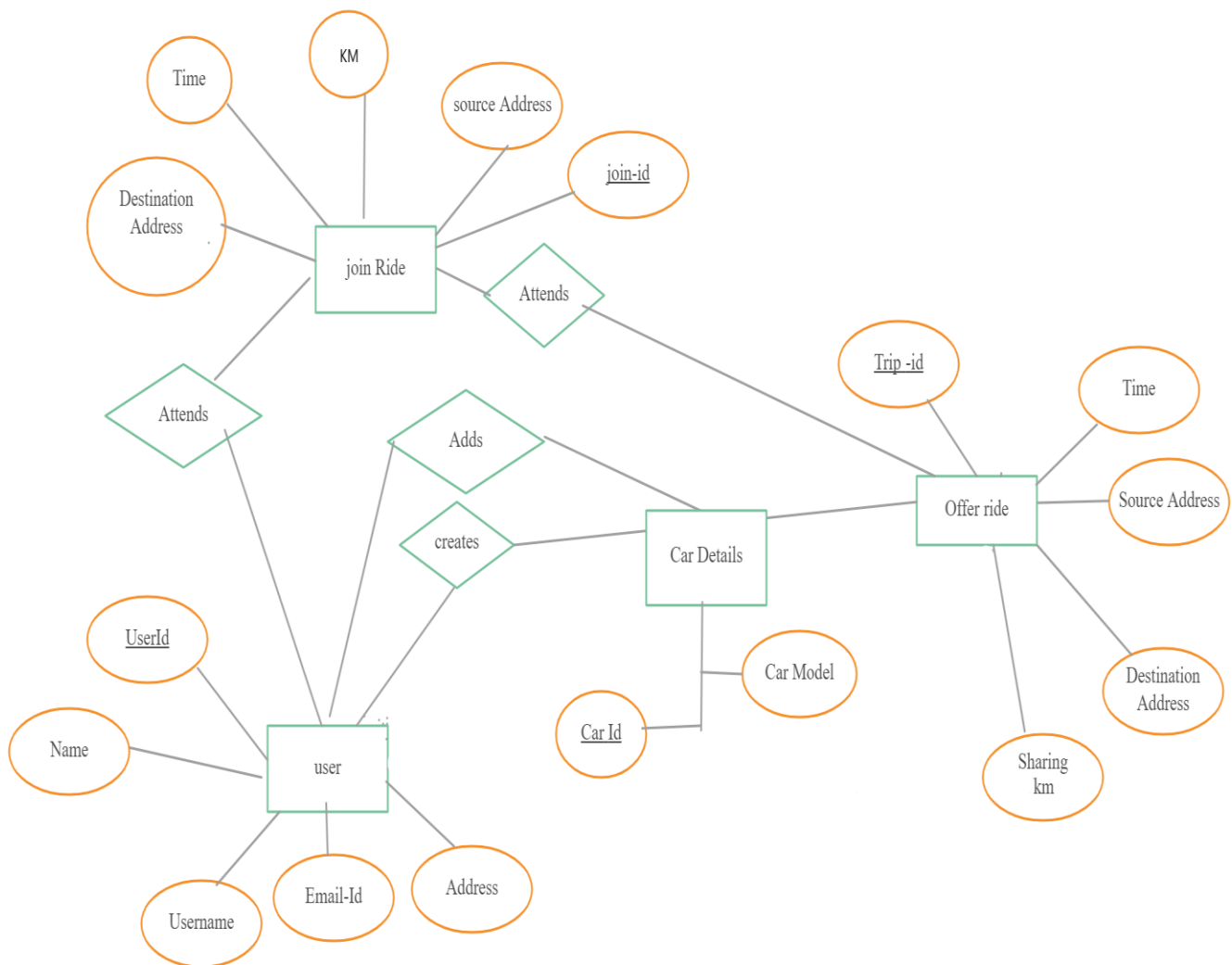
- This is an Android based module.
- It will work as an intermediate between the rider and the driver, in order to meet the travelling requirements.
- The main function of this module is to help the user/rider for booking/cancelling the ride and payments.
- This module also manages locations and details of the rider.

### **2. Rider's Application Module:**

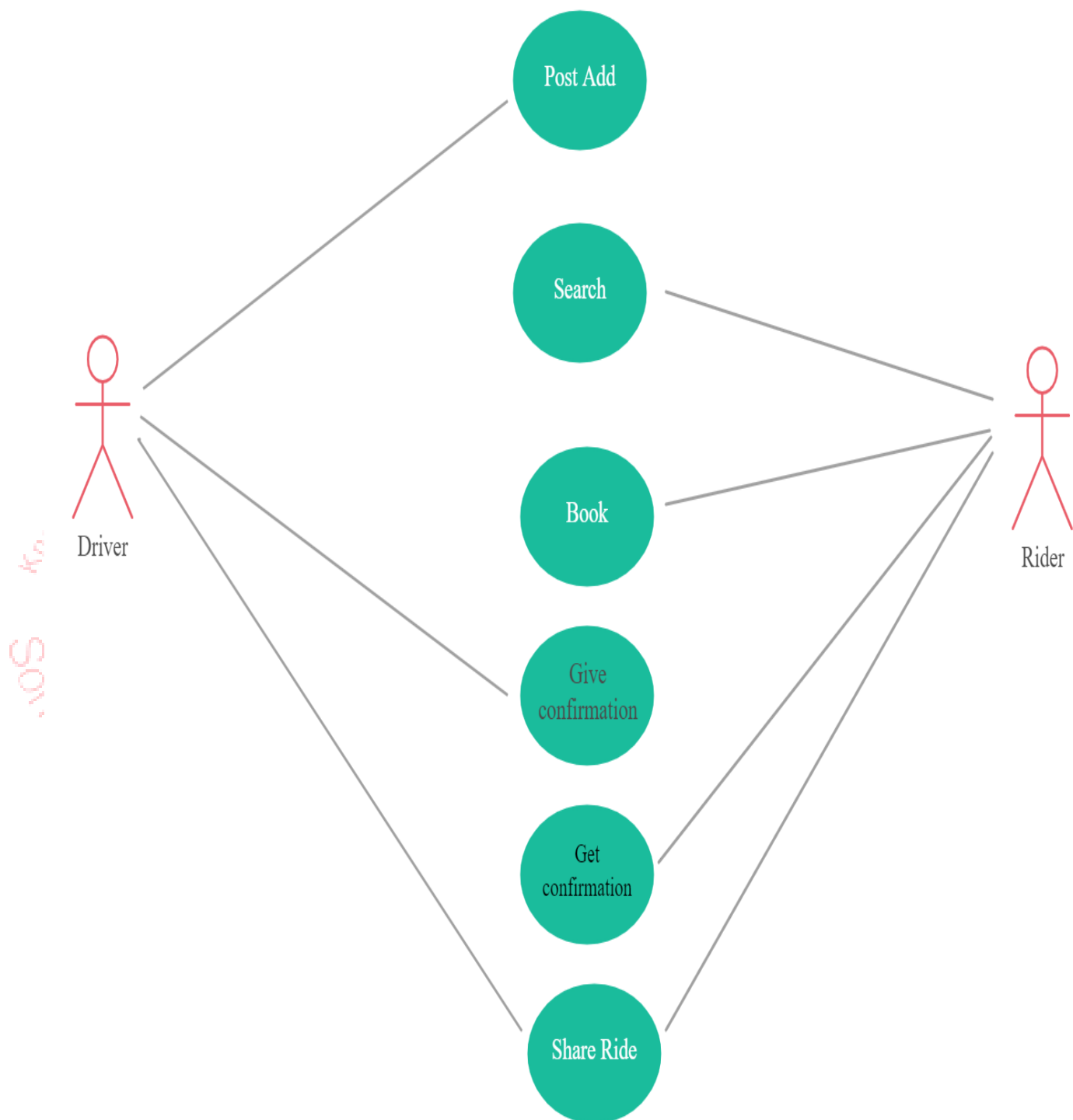
- It is also an android based module.
- This module will mainly have functions related to the drivers who are willing to share a ride.
- It manages the details of the driver's vehicle and share these details to the rider for safety purposes.

## 6.3 System Diagrams

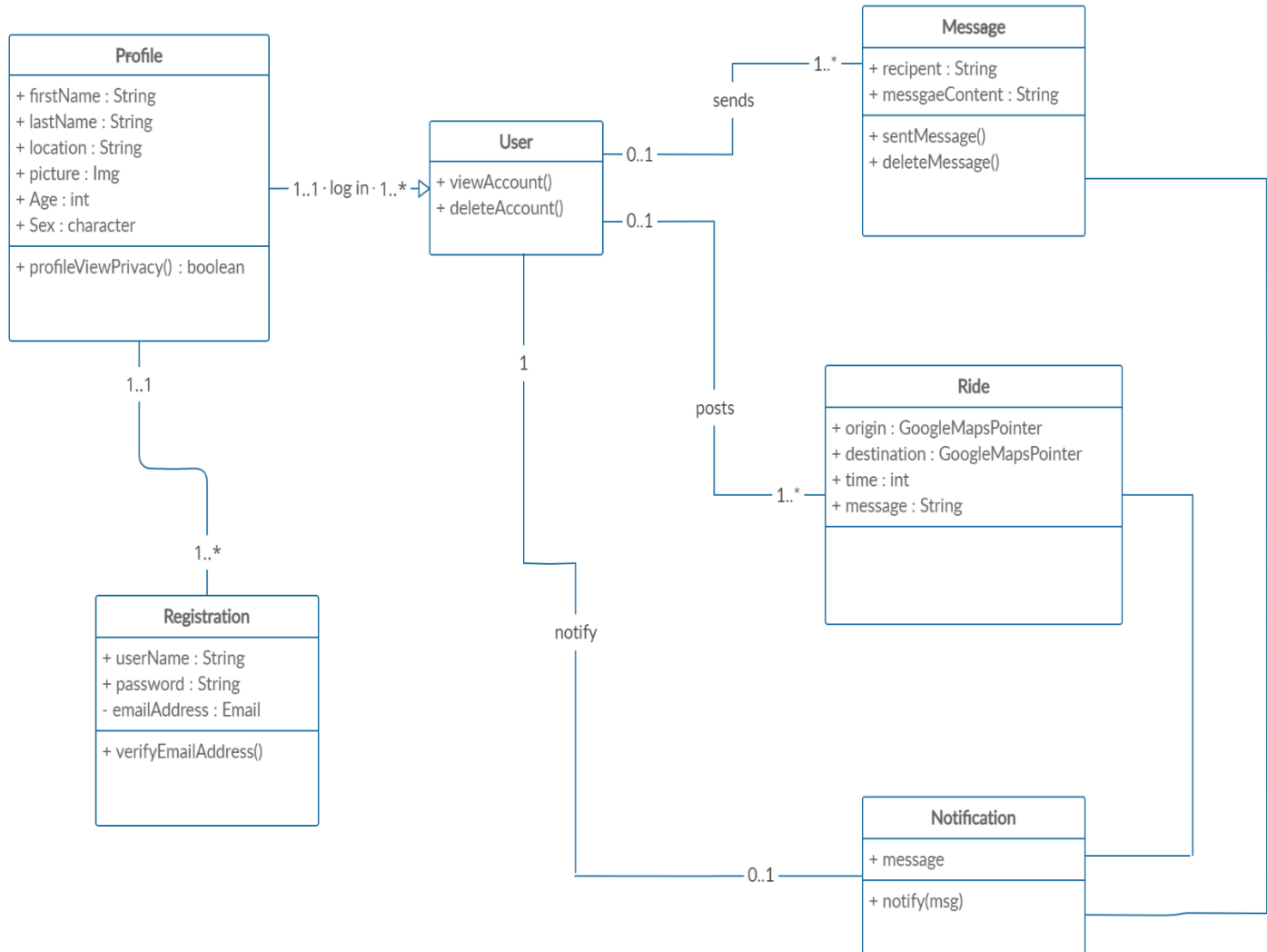
### ❖ Entity-Relationship Diagram:



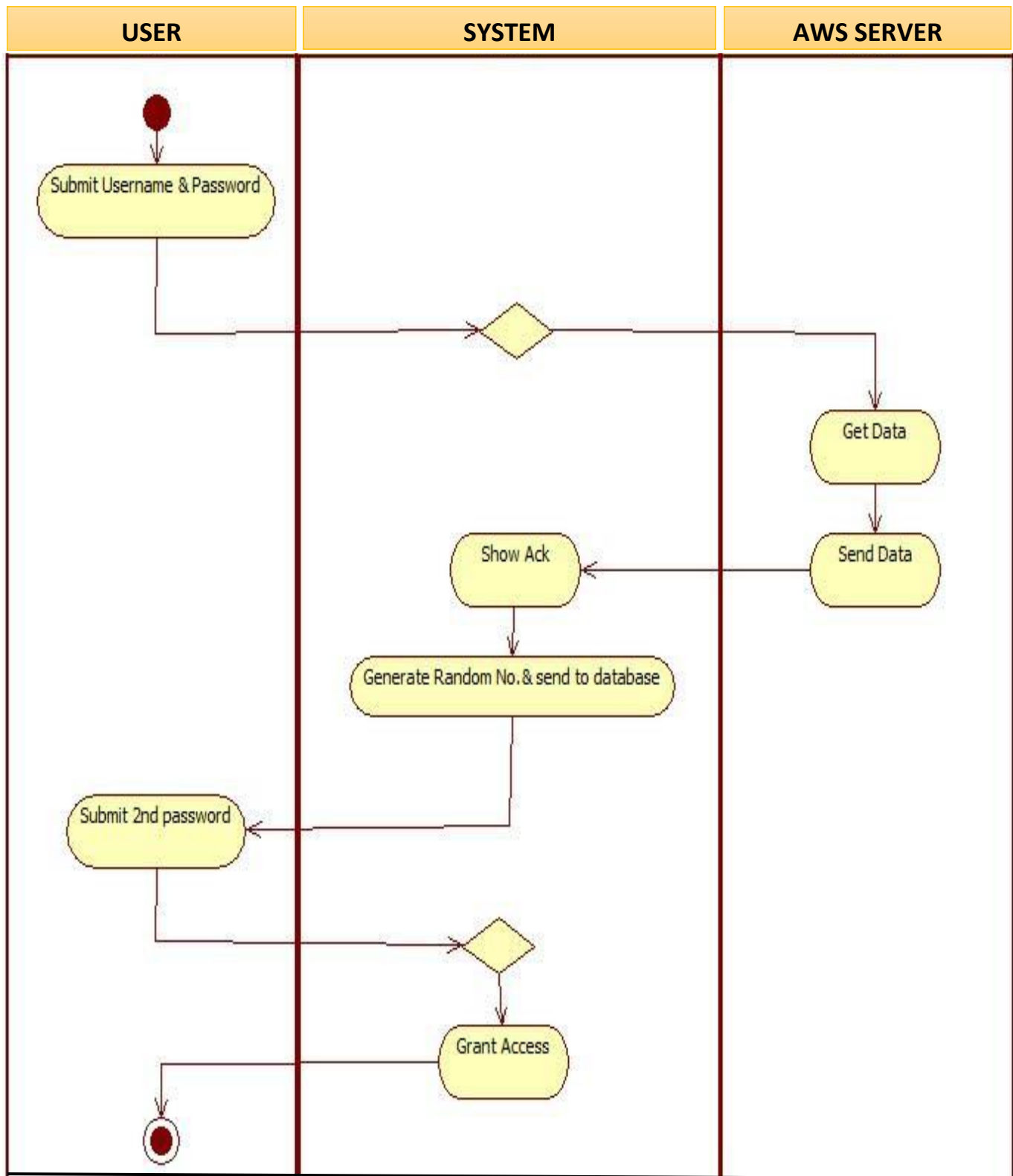
## ❖ Use Case Diagram:



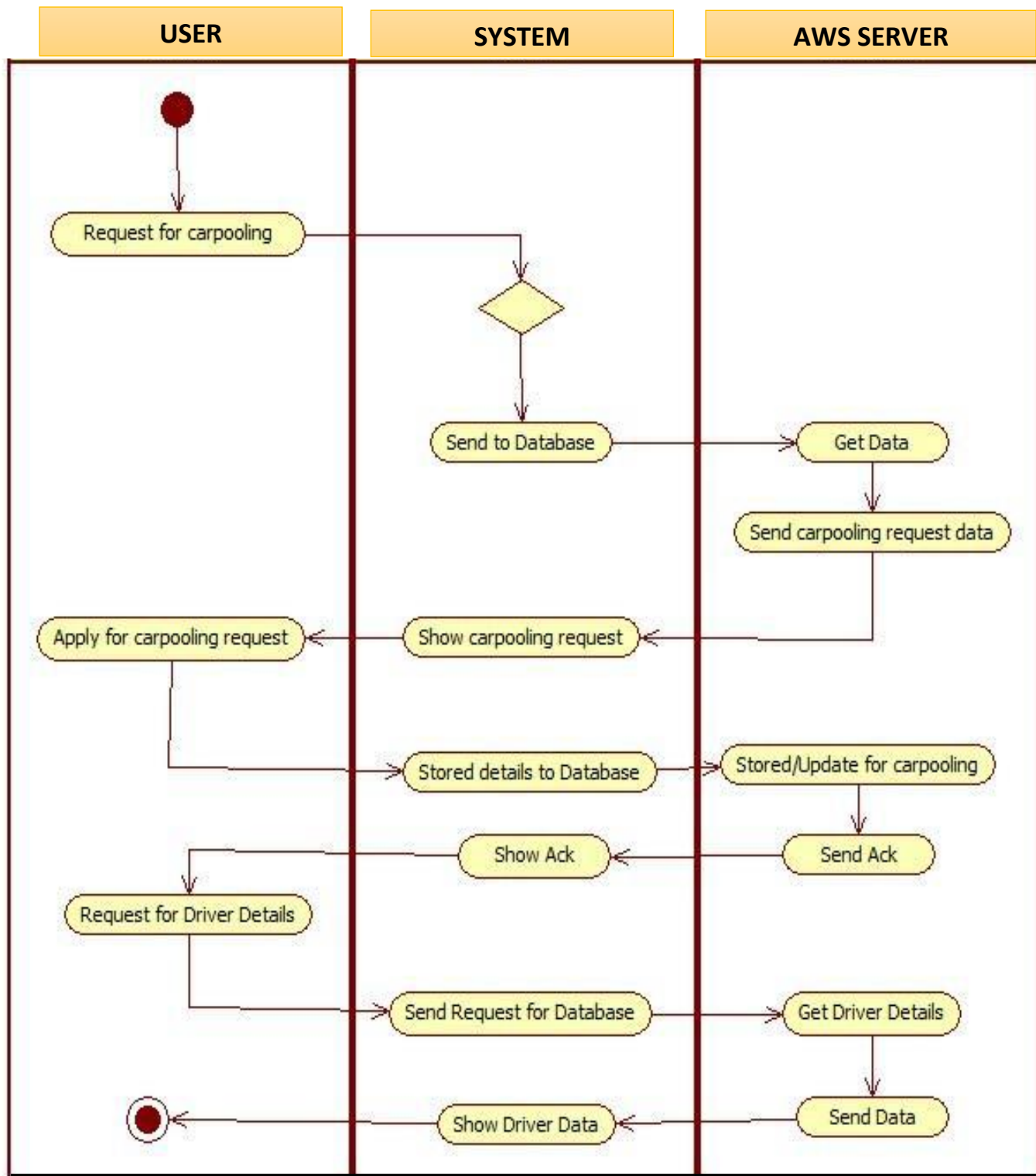
## ❖ Class Diagram:



## ❖ Activity Diagrams: (LOGIN)

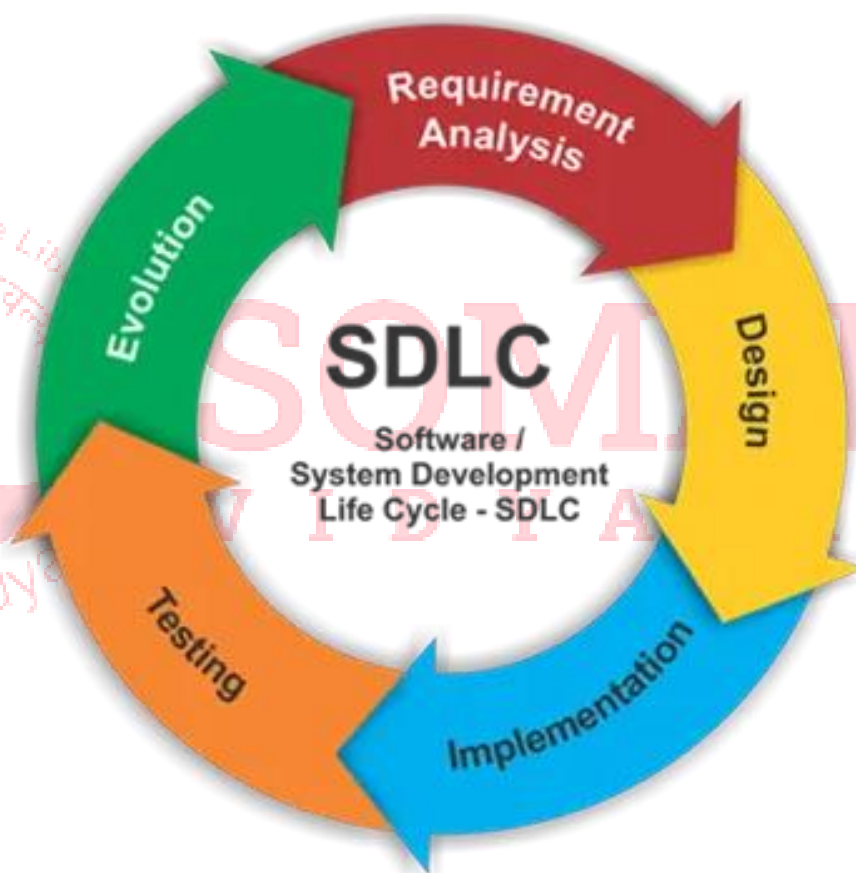


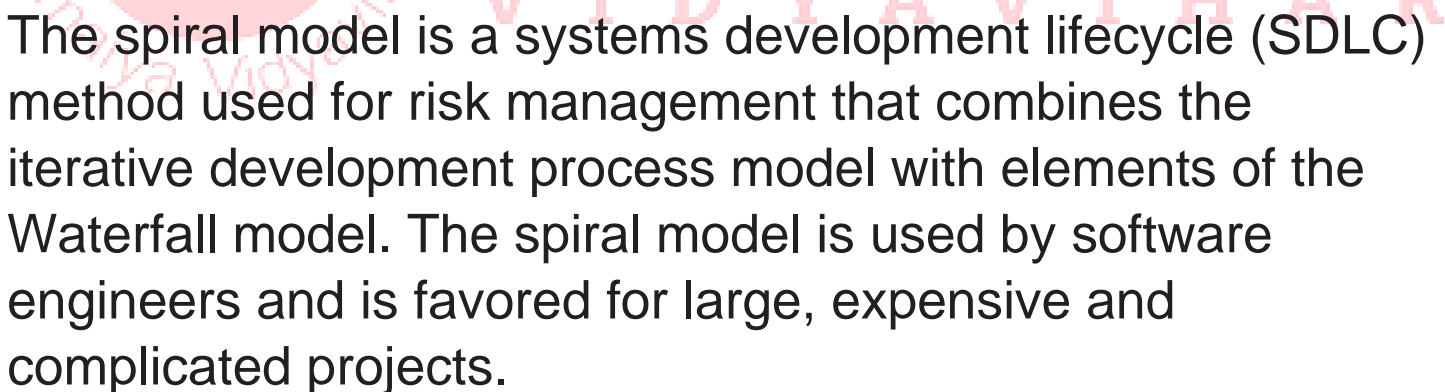
## ➤ Activity Diagram: (Workflow)





# CHAPTER 7





When viewed as a diagram, the spiral model looks like a coil with many loops. The number of loops varies based on each project and is often designated by the project manager. Each loop of the spiral is a phase in the software development process.

The spiral model enables gradual releases and refinement of a product through each phase of the spiral as well as the ability to build prototypes at each phase. The most important feature of the model is its ability to manage unknown risks after the project has commenced; creating a prototype makes this feasible.

## **Uses of the spiral model**

As mentioned before, the spiral model is best used in large, expensive and complicated projects.

Other uses include:

- Projects in which frequent releases are necessary.
- Projects in which changes may be required at any time.
- Long term projects that are not feasible due to altered economic priorities.
- Medium to high risk projects.
- Projects in which cost and risk analysis is important.
- Projects that would benefit from the creation of a prototype.
- Projects with unclear or complex requirements.

## **Spiral model phases**

When looking at a diagram of a spiral model, the radius of the spiral represents the cost of the project and the angular degree represents the progress made in the current phase. Each phase begins with a goal for the design and ends when the developer or client reviews the progress.

Every phase can be broken into four quadrants: identifying and understanding requirements, performing risk analysis, building the prototype and evaluation of the software's performance.

Phases begin in the quadrant dedicated to the identification and understanding of requirements. The overall goal of the phase should be determined and all objectives should be elaborated and analyzed. It is important to also identify alternative solutions in case the attempted version fails to perform.

Next, risk analysis should be performed on all possible solutions in order to find any faults or vulnerabilities -- such as running over the budget or areas within the software that could be open to cyber attacks. Each risk should then be resolved using the most efficient strategy.

In the next quadrant, the prototype is built and tested. This step includes: architectural design, design of modules, physical product design and the final design. It takes the proposal that has been created in the first two quadrants and turns it into software that can be utilized.

Finally, in the fourth quadrant, the test results of the newest version are evaluated. This analysis allows programmers to stop and understand what worked and didn't work before progressing with a new build. At the end of this quadrant, planning for the next phase begins and the cycle repeats. At the end of the whole spiral, the software is finally deployed in its respective market.

### **Steps of the spiral model**

While the phases are broken down into quadrants, each quadrant can be further broken down into the steps that occur within each one. The steps in the spiral model can be generalized as follows:

- The new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.

- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
- A second prototype is evolved by a fourfold procedure: (1) evaluating the first prototype in terms of its strengths, weaknesses, and risks; (2) defining the requirements of the second prototype; (3) planning and designing the second prototype; (4) constructing and testing the second prototype.
- The entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation and other factors that could result in a less-than-satisfactory final product.
- The existing prototype is evaluated in the same manner as was the previous prototype, and, if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
- The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
- The final system is constructed, based on the refined prototype.
- The final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing basis to prevent large-scale failures and to minimize downtime.



## **Benefits of the spiral model**

As mentioned before, the spiral model is a great option for large, complex projects. The progressive nature of the model allows developers to break a big project into smaller pieces and tackle one feature at a time, ensuring nothing is missed. Furthermore, since the prototype building is done progressively, the cost estimation of the whole project can sometimes be easier.

Other benefits of the spiral model include:

- **Flexibility** - Changes made to the requirements after development has started can be easily adopted and incorporated.
- **Risk handling** - The spiral model involves risk analysis and handling in every phase, improving security and the chances of avoiding attacks and breakages. The iterative development process also facilitates risk management.
- **Customer satisfaction** - The spiral model facilitates customer feedback. If the software is being designed for a customer, then the customer will be able to see and evaluate their product in every phase. This allows them to voice dissatisfactions or make changes before the product is fully built, saving the development team time and money.

## Limitations of the spiral model

Limitations of the spiral model include:

- **High cost** - The spiral model is expensive and, therefore, is not suitable for small projects.
- **Dependence on risk analysis** - Since successful completion of the project depends on effective risk handling, then it is necessary for involved personnel to have expertise in risk assessment.
- **Complexity** - The spiral model is more complex than other SDLC options. For it to operate efficiently, protocols must be followed closely. Furthermore, there is increased documentation since the model involves intermediate phases.
- **Hard to manage time** - Going into the project, the number of required phases is often unknown, making time management almost impossible. Therefore, there is always a risk for falling behind schedule or going over budget.



# CHAPTER 8



## Implementation & Testing

## 8.1 Implementation Approaches

Sr no.	Implementation Plan	Action
1.	Modules	i. User Module ii. Rider Module
2.	Percentage Completed	i. User Module-100% ii. Rider Module-100%
3.	Status	i. User Module-Completed ii. Rider Module-Completed
4.	Day Started	15/08/2020
5.	Day to be completed	
6.	Actual completion date	20/02/2022
7.	Module Assignment	Neel Patel & Neeraj Pal
8.	Importance of Module	i. User Module: High ii. Rider Module: High

## 8.2 Code Efficiency

### ✓ React Native:

- **Cross-platform/ time efficient**

React Native's advantages stem mostly from the fact that it is a cross-platform technology. Meaning, the source code is processed and rendered into native Android and iOS components. Although this carries consequences in both quality and performance, it allows the same source code to be used to make apps for both platforms. For developments where cost and time are priorities, React Native offers increasingly more attractive options.

From a technical perspective, React Native development allows the use of one framework and one source code to build apps for Android & iOS apps. Apps produced are real native software. Simply put, whether building an app for Android, iOS or both, this framework can be used to create the same source code, which is then rendered (translated) into corresponding native components.

This advantage suits other cross-platform frameworks, as well. Flutter is based on the same concept as RN – building an app with one source code for multiple operating systems.

- **Hot Reloading**

The results of a developer's work in React Native can be monitored in pretty much real-time, without the need of compiling the code. Hot Reloading allows code changes to be seen straight away.

This is one of the most liked features in cross-platform frameworks like React Native and Flutter.

- **Cost-efficient solution**

Currently, the cost associated with creating an app can be reduced by 25-30% when choosing React Native. The savings come from having just one development team instead of two. This means easier project management and more control over the uniformity of the production.

✓ **Node JS:**

Node.js is known to produce super-fast-performing and scalable apps as it uses event-driven architecture and non-blocking (asynchronous) tasks that run on a single thread.

Fast and scalable network applications can be built as the JavaScript execution is fast. The single threaded model with event loop is also highly helpful and allows handling multiple client requests. It is one of the major advantages of using Node.

## ✓ AWS Amplify:

AWS Amplify provides the capability of building a variety of apps like iOS, Android, React Native, Flutter, React, and Vue that can easily communicate with the AWS services via the AWS Amplify Framework.

- **Store and Sync Data Securely**

AWS Amplify lets you securely sync and store data seamlessly between applications with the help of Amazon AppSync and Amazon S3. It also allows for easy offline synchronization.

- **Quick Backend Updates**

Due to the serverless nature of AWS Amplify, making changes to back-end related functions has gotten much easier. This saves time that used to be previously spent on configuring and maintaining back-end features.

- **DataStore**

AWS Amplify also provides DataStore capability, which offers a programming model for leveraging shared and distributed data without having to write additional code both for offline and online use cases. The result is a distributed and cross-user data, which is as simple as working with local-only data.

- **Deployment and App Development**

AWS Amplify promotes faster app development and is also very useful for continuous deployments.

- ✓ **Google Cloud Platform**

Google Cloud protects your data, applications, infrastructure, and customers from fraudulent activity, spam, and abuse with the same infrastructure and security services Google uses. Google Cloud's networking, data storage, and compute services provide data encryption at rest, in transit, and in use.

Google Cloud Allows Quick Collaboration: Many users can contribute to and access projects at the same time as data is stored in the cloud instead of their computers. Google's Investments in Security Protect Customers: Customers benefit from process-based and physical security investments made by Google.

## 8.3 Testing Approaches

### Functional vs. Non-functional Testing

The goal of utilizing numerous testing methodologies in your development process is to make sure your software can successfully operate in multiple environments and across different platforms. These can typically be broken down between functional and non-functional testing. Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using uses cases provided by the design team or business analyst. These testing methods are usually conducted in order and include:

#### ➤ Unit testing

##### i. User Module:

##### **Testing of Validation Controls in User Registration**

- Name text box can accept only upto 24 characters.
- Username text box can accept minimum of 5 and maximum of 24 characters.
- Email input box can only be validated with an .e
- du TLD.
- Password should be minimum 6 characters long & must contain one of 0-1, A-Z, a-z and special characters.

## Testing of User Verification:

- Username must be same as entered before.
- OTP entered must match which is sent in the given email by user.

## ii. Rider Module:

### Testing of Validation Controls in Rider Registration

- Name text box can accept only upto 24 characters.
- Username text box can accept minimum of 5 and maximum of 24 characters.
- Email input box can only be validated with an .edu TLD.
- Password should be minimum 6 characters long & must contain one of 0-1, A-Z, a-z and special characters.

## Testing of Rider Verification:

- Username must be same as entered before.
- OTP entered must match which is sent in the given email by user.



## ➤ Integration testing

- After the unit testing for each module is done, both the modules are tested simultaneously.
- Once the user/rider logs in after entering the correct credentials, he/she gets complete access of the respective applications.
- Both the user and rider can change his/her password if they want or they might forget it.

**Non-functional testing methods incorporate all test types focused on the operational aspects of a piece of software. These include:**

### ● Performance testing

- Whenever the rider goes online with any of his/her vehicle, the user can easily see their vehicle on the map.
- When user creates any order, the data is stored in AWS and is fetched by the rider.

### ● Security testing

- For security we have added a shield button in both the applications so that user can see the rider details and vice versa.
- As well as only users with .edu mail which can be only provided by university ,can login in our applications.

- **Compatibility testing**

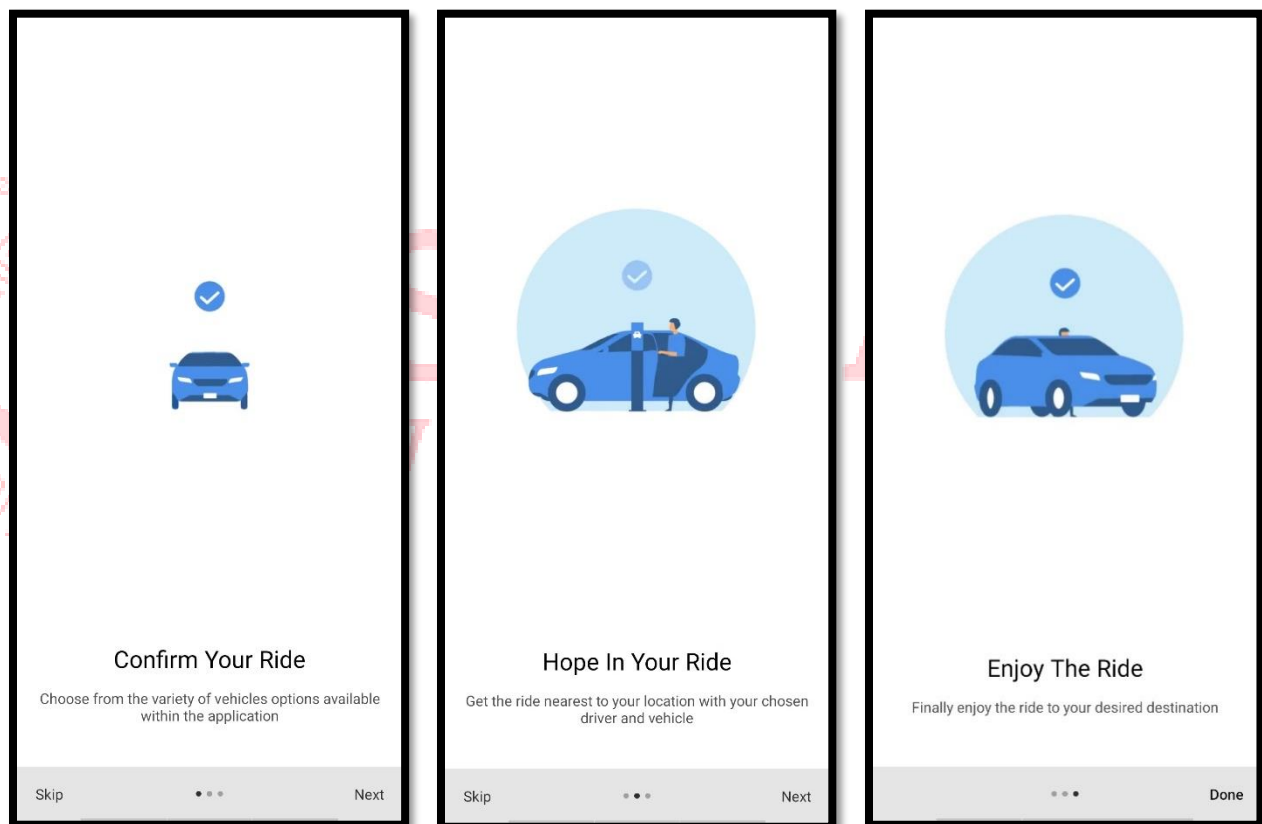
- Our applications are compatible with almost all android devices.

## 8.4 Modification & Improvements

- One of the most attractive modifications done in the applications are:
  - i. **Splash Screen:** A splash screen is a particular screen on a website or piece of software that displays while the application or other item is loading.



- ii. **Onboarding-Screen:** An onboarding screen is like a walkthrough, aimed to introduce what an app does to a user and of course how to use it. That's the simplest way of describing it. Designing it however is a totally different thing.

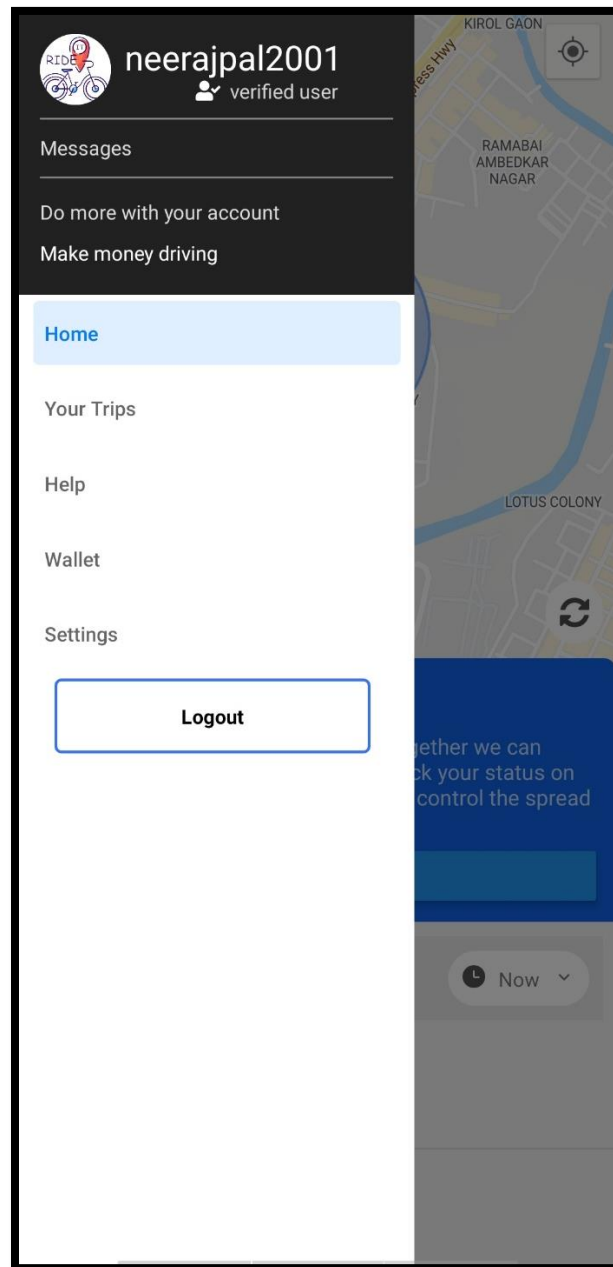


- iii. **Custom Authentication UI:** Authentication screen was design completely different from previous provided by AWS. A fully customizable Screens were added in each and every phase of authentication like registration, verification, sign-up, login, etc.

The image displays three mobile application screens for authentication, arranged side-by-side.

- Sign In Screen:** Features a logo with a bicycle and a location pin labeled 'IT' above the word 'RIDE'. It includes input fields for 'Username' and 'Password' (with a toggle icon), a blue 'Sign In' button, and links for 'Forgot Password?' and 'User not confirmed?'. At the bottom, it says 'Don't have an account? Create one'.
- Create an account Screen:** Has a title 'Create an account'. It contains input fields for 'Name', 'Username', 'Email', 'Password' (with a toggle icon), and 'Confirm Password' (with a toggle icon). A blue 'Register' button is present. Below the button, it states: 'By registering, you confirm that you accept our [Terms of Use](#) and [Privacy Policy](#).' At the bottom, it says 'Have an account? Sign In'.
- Confirm your Username Screen:** Has a title 'Confirm your Username'. It includes an input field for 'Username', another for 'Enter your confirmation code', a blue 'Confirm' button, and a 'Resend Code' button. At the bottom, it says 'Back to Sign In'.

- iv. **Custom Drawer:** Drawer was customized at the time of design phase. New elements were added in it including username and many other things.



- v. **Logo:** Logo for each application was designed in that way that it can be easily recognizable for user and rider respectively.

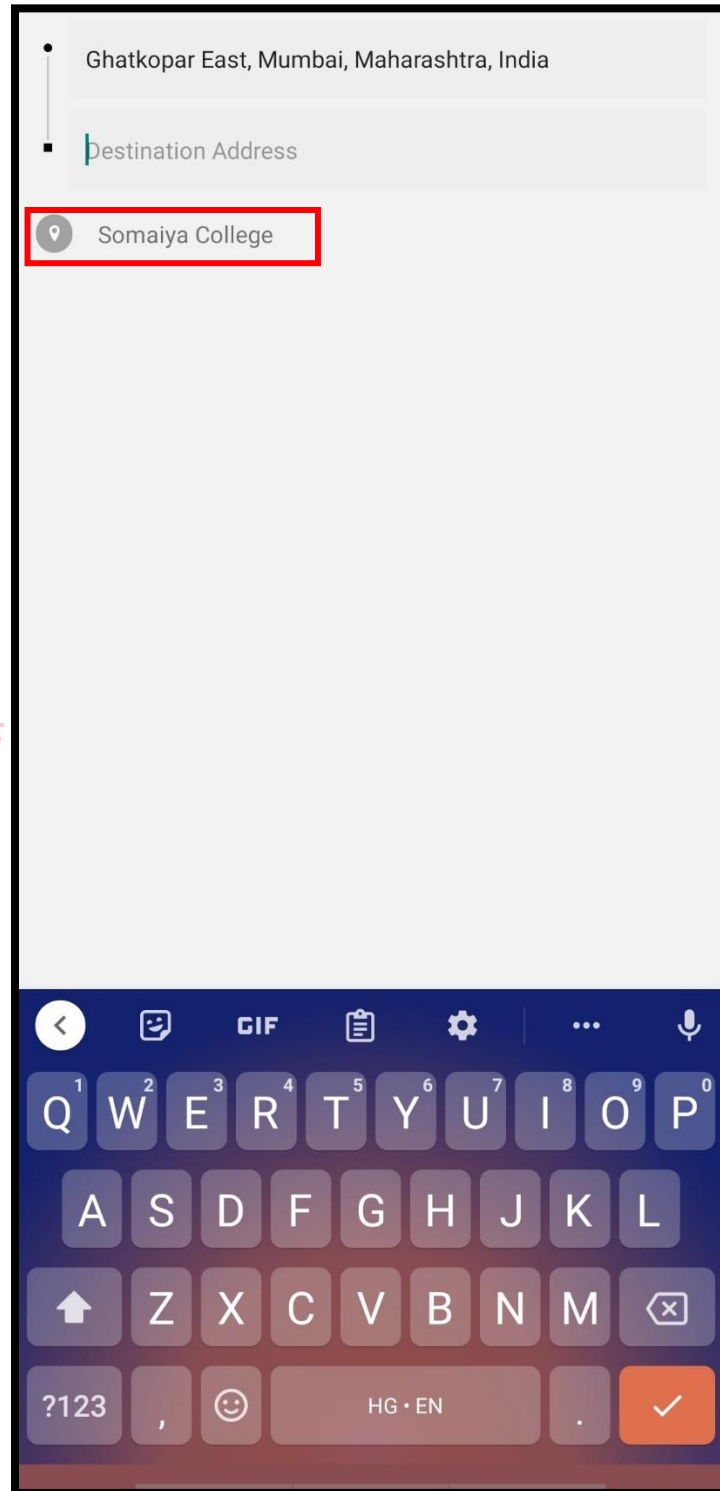


**RIDE-IT USER**



**RIDE-IT RIDER**

- vi. **Pinned Destination:** At the time of modification “Somaiya College” was pinned as the destination for quick access.



# CHAPTER 9



## ***Results & Discussion***



## 9.1 Test Case

Sr no.	Module	Expected Output	Actual Output	Status
<b>Test Case 1</b>	User & Rider Login	With .edu Account Login Successful	Login Successful	Complete
<b>Test Case 2</b>	User & Rider Confirmation	OTP received in .edu Mail	OTP received	Complete
<b>Test Case 3</b>	Book a Ride	Ride is booked after selecting destination and vehicle	Ride is booked	Order status pending
<b>Test Case 4</b>	Rider accepts a ride	Ride is accepted by pressing the popup order	Picking Up (Username) if order is accepted	Picking Up the User
<b>Test Case 5</b>	Rider decline a ride	Ride is declined by tapping on decline	Return to homepage until there is no ride request	Ride is Declined & Return to Homepage

<b>Test Case 6</b>	Journey Completed	User reaching the destination along with the rider	<b><u>User:</u></b> Destination Reached <b><u>Rider:</u></b> The user has been reached to the desired destination	Order Completed
<b>Test Case 7</b>	Log Out	User/Rider can logout from the account using Logout Button	User/Rider Signed out Successfully	Complete



V I D Y A V I H A R

## Test Case Scenario 1:

**Create an account**

Neeraj Pal

NeerajPal2001

neerajg.01@gmail.com

Enter a valid email with .edu TLD

Neeraj@2001

Neeraj@2001

**Register**

By registering, you confirm that you accept our [Terms of Use](#) and [Privacy Policy](#).

Have an account? Sign In

**Create an account**

Neeraj Pal

NeerajPal2001

neeraj.pal@somaiya.edu

Neeraj@2001

Neeraj@2001

**Register**

By registering, you confirm that you accept our [Terms of Use](#) and [Privacy Policy](#).

Have an account? Sign In

## Test Case Scenario 2:

**Confirm your email**

NeerajPal2001

Enter your confirmation code

**Confirm**

[Resend Code](#)

[Back to Sign In](#)

[RideIT] Please Verify Your Account

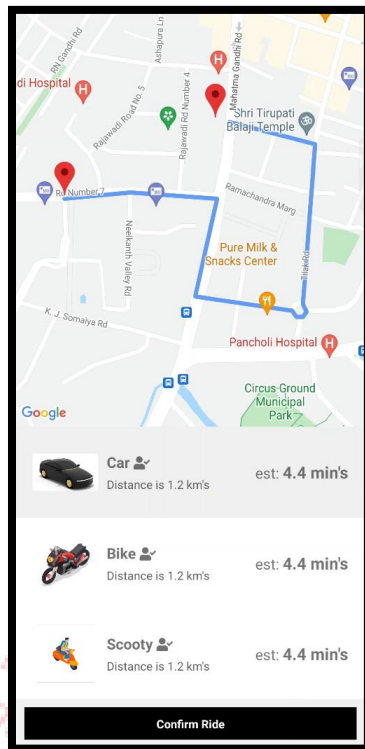
External Inbox

no-reply@verificationemail... 1:13 pm to me

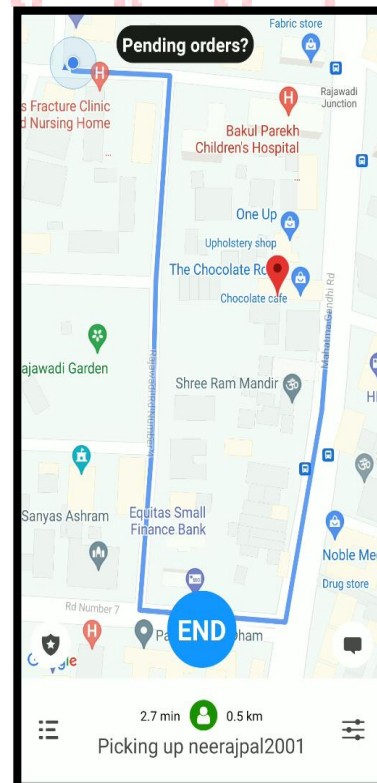
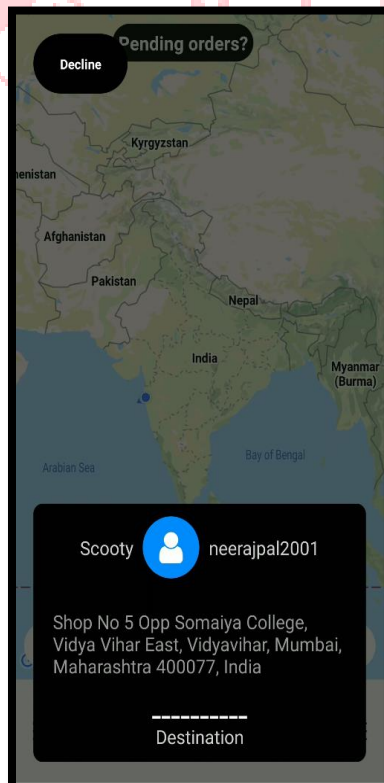
Dear Customer, 766768 is the One Time Password (OTP) for the verification of your account in RideIT Application. Please don't share this OTP with anyone for security purposes, If not requested this OTP then ignore this mail.

[Reply](#) [Reply all](#) [Forward](#)

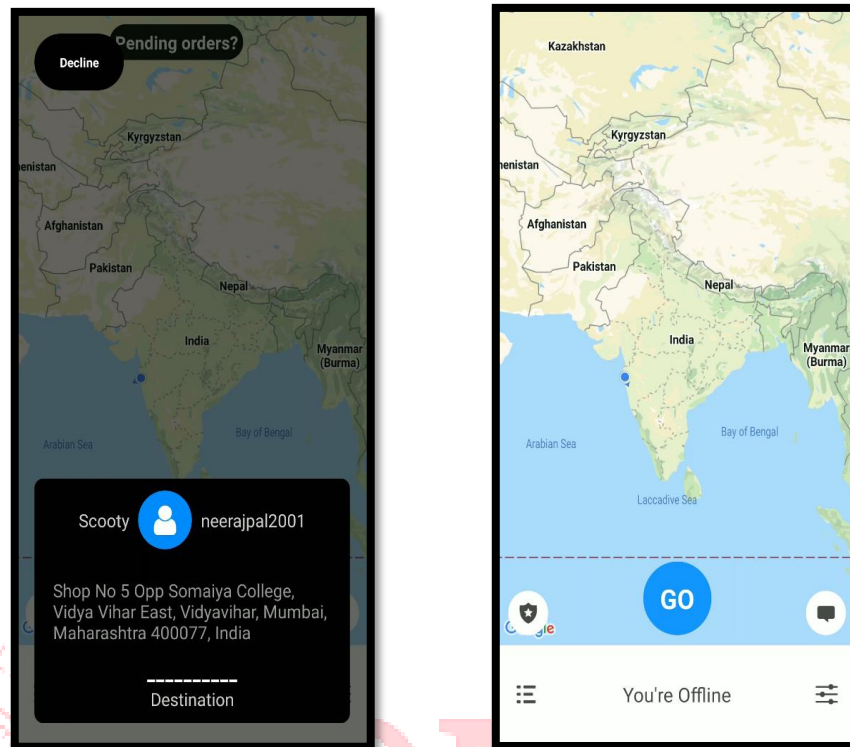
### Test Case Scenario 3:



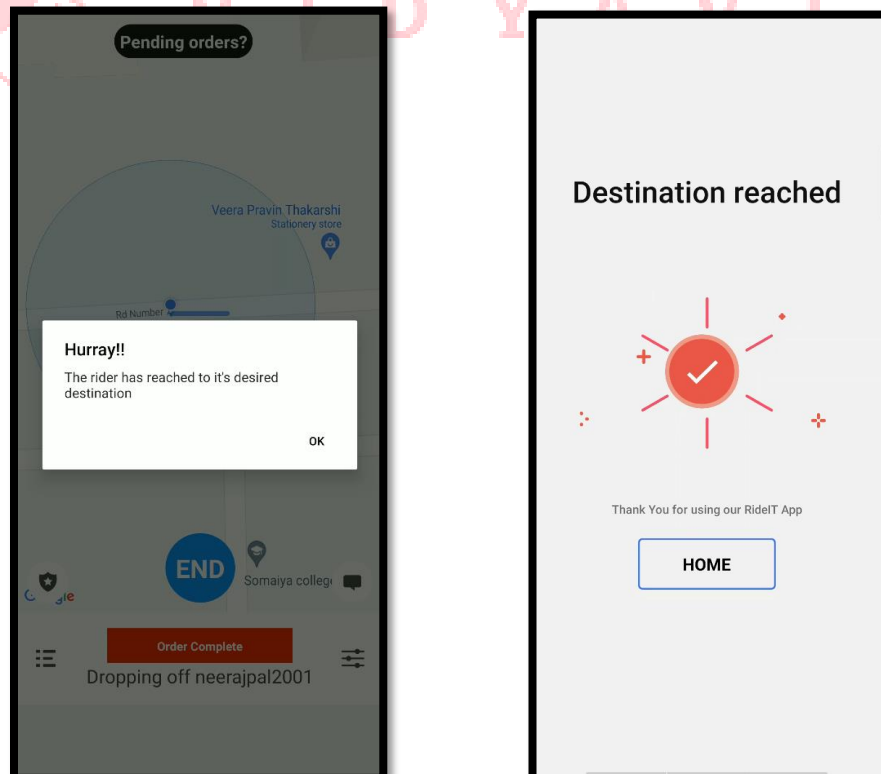
### Test Case Scenario 4:



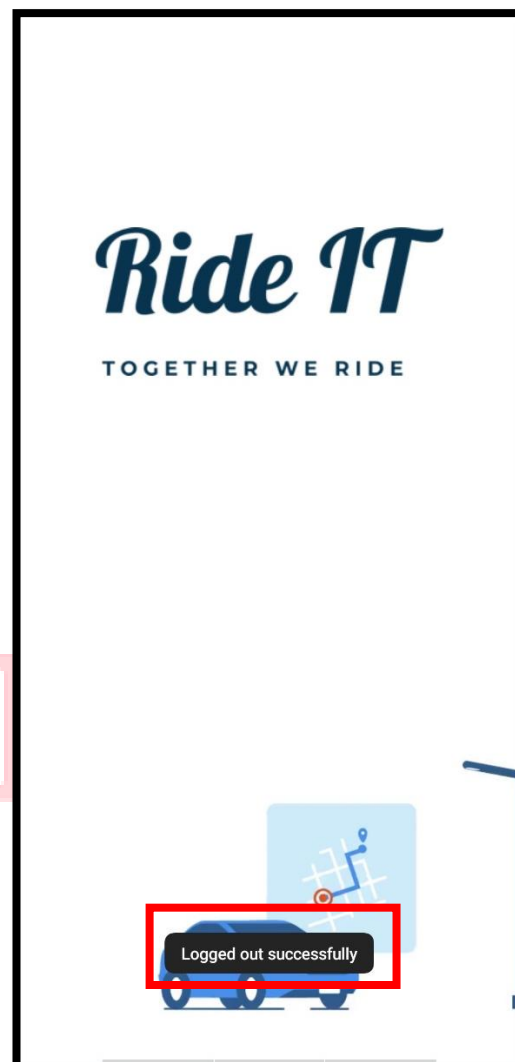
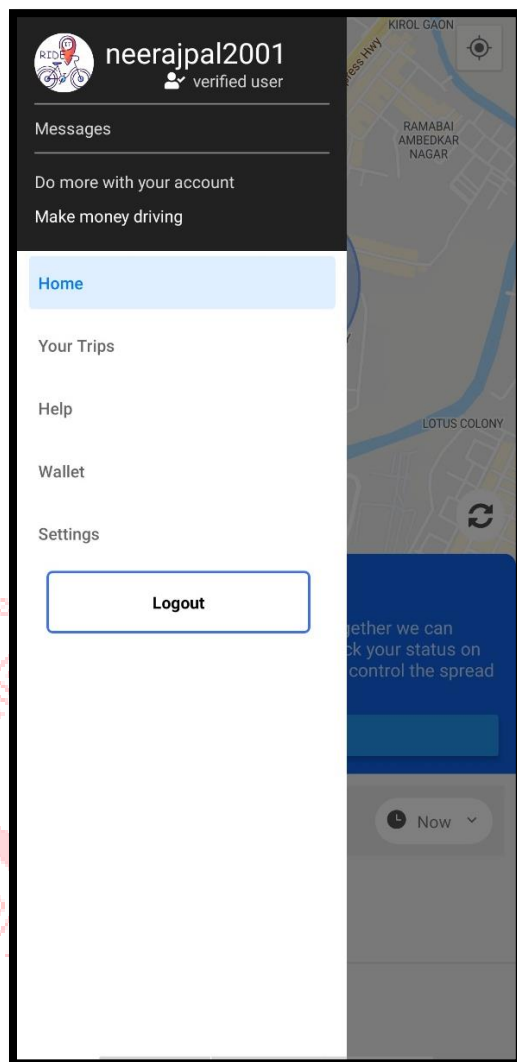
## Test Case Scenario 5:



## Test Case Scenario 6:



## Test Case Scenario 7:




## 9.2 User Documentation

The working of the carpooling service is mainly dependent on two modules:

1. User Module
2. Rider Module

Other than modules, the functionalities provided in the applications are as follows:

- **HOME SCREEN: For User**

**1. Current Location:** The current location is a feature added that helps us locate the device's position on the Map by tapping on that .

**2. Refresh Button:** This button refreshes and fetch the latest location of the rider/car position, whenever pressed.

**3. Car/Rider Location:** User will be able to see the location of the rider/car on the map.

**4. Covid Message:** The home screen has covid precaution message and a link is provides with it which will redirect the user to the government covid website.

**5. Where to?:** Just below the covid message, there is “Where to?” bar for the user which redirects to the screen where he/she can enter the source /destination preferred .

**6. Drawer Button:** This button will slide the drawer from the left in which there are many options provides along with the username and Logout Button.

- **HOME SCREEN: For Rider**

**1. Go Online:** Whenever this button is pressed by the rider, the rider goes online and will start receiving ride requests.

**2. Contact Button:** The following button will popup a dialog box which contain the information to contact the developers.

**3. User Details:** Shield shape Button on the screen pops up when pressed and display the information of the users when rider accepts the ride request.

**4. Drawer Button:** This button will slide the drawer from the left in which there are many options provides along with the username and Logout Button.



# CHAPTER 10



## 10.1 Conclusion

RideIT is an Graduation Project which consist of two Modules/application namely RideIT & RideIT-Driver. RideIT for the user and other one for the riders.

The users and rider will create their respective account in the applications with .edu mail provided by the university.

User can book ride to his/her desire location and if and only If rider accepts the request when he/she is online the ride will take place.

The aim of this project is to provide such user friendly and interactive service via Application User interface. This system provides a simple and quicker interaction for the users who are in need of transportation help. This project mainly focuses for the students and staff members going to the same university who wants to share the ride with someone. If the students with vehicles will share the rides it will help create a bond between strangers and also helps the environment saving fuel. Staff members having vehicle also can share their rides with their students or other staff members.

## 10.2 Advantages & Limitations

### ✓ Advantages of RideIT:

- There is no such service in India which provides a car pool for university staff and students.
- The person registered in the applications will be verified and will be of same university as he/she will be using the mail provided by the university.
- The person can easily track the other person as there is a security option to view the details of users/riders.
- It is fast and accurate because of the backend used is “AWS”. Same goes for security as well.

### ✓ Limitation of RideIT:

- Some parts of the applications are hardcoded so they cannot be changes as functionalities.
- In the Drawer, only username will vary from person to person. Other all functionalities are on “Coming Soon”.
- Application functionalities cannot work properly in case of poor connectivity.

## 10.3 Future Scope of the project

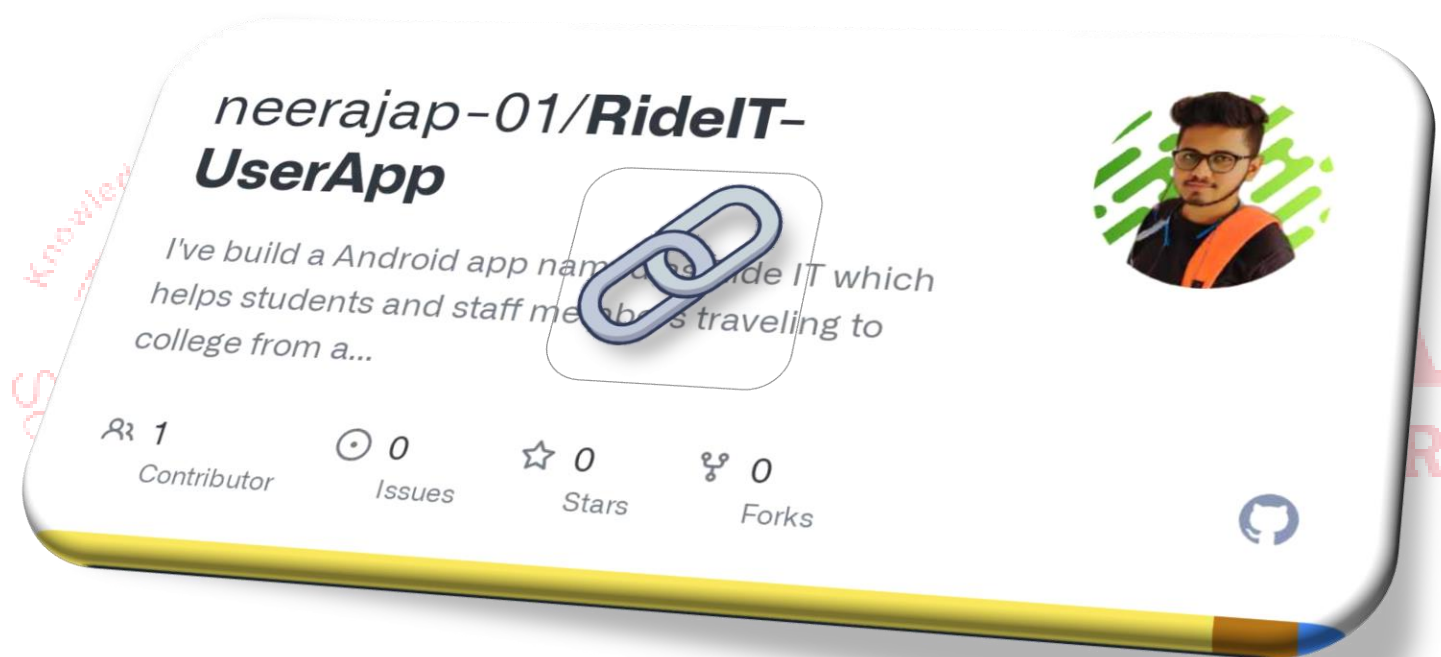
- Currently the Users cannot sign in or register using mobile number. This will be added soon.
- In future we can tie up with other universities as well.
- Soon it can also work under poor connectivity.
- Some hardcoded parts can be converted into fully functional elements.



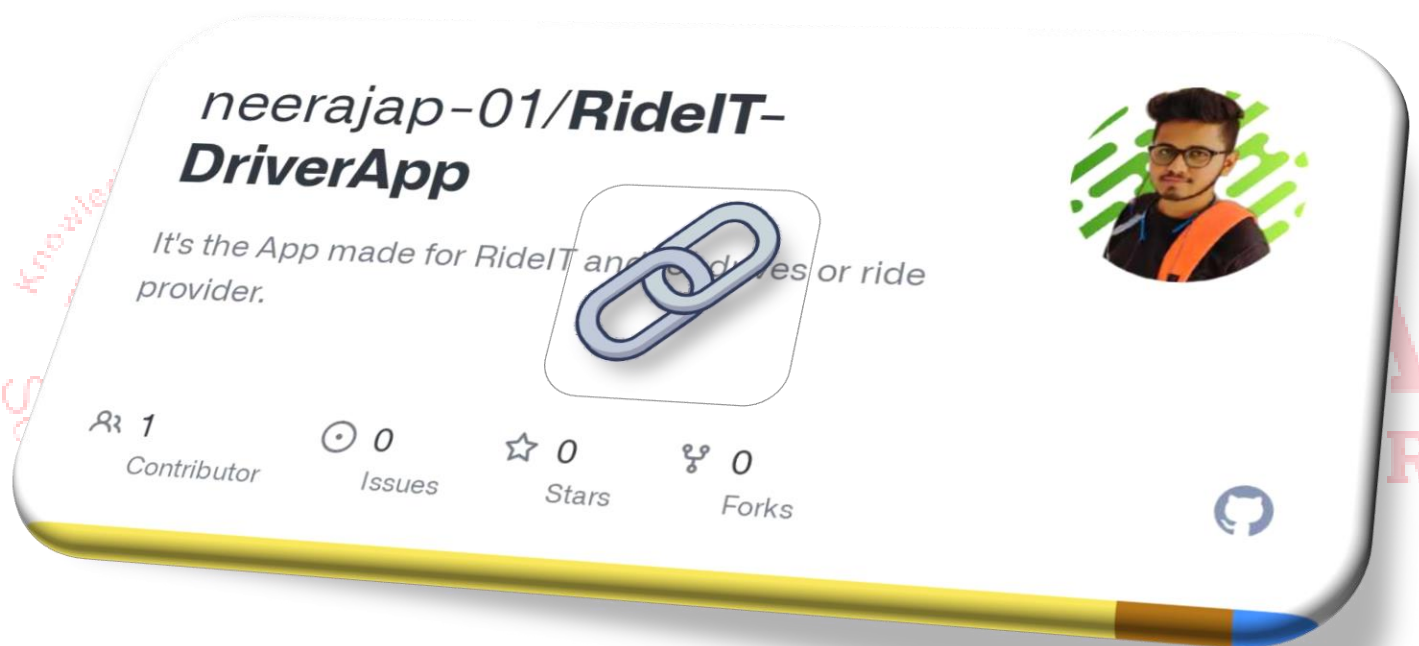
SOMAIYA  
VIDYAVIHAR

## 10.4 Links

### Source code for User App:



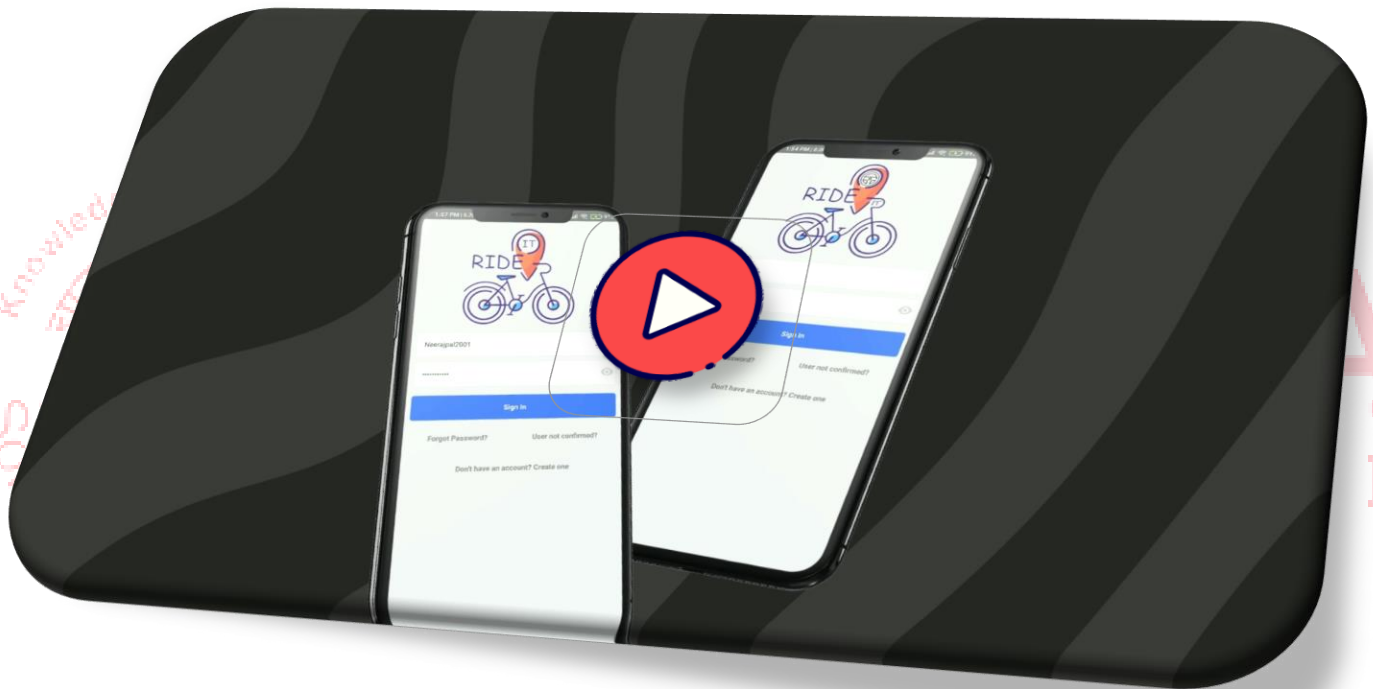
## Source code for Rider App:



## Official website:



# Full Demo of the Project:





## My Portfolio Website:



## REFERENCES

- <https://reactnative.dev/docs/environment-setup>
- <https://oblador.github.io/react-native-vector-icons/>
- [https://www.youtube.com/playlist?list=PLY3ncAV1dSVDkO8RF\\_H6kMykUce9X2jkc](https://www.youtube.com/playlist?list=PLY3ncAV1dSVDkO8RF_H6kMykUce9X2jkc)
- <https://www.npmjs.com/package/react-native-maps>
- <https://www.npmjs.com/package/react-native-google-places-autocomplete>
- <https://www.npmjs.com/package/react-native-maps-directions>
- <https://www.npmjs.com/package/react-native-get-location>
- <https://docs.amplify.aws/start/q/integration/react-native/>
- **Most helpful during errors are:**
  - <https://www.google.com/>
  - <https://stackoverflow.com/>
  - <https://github.com/>
  - <https://www.youtube.com/>