A PROJECT REPORT

ON

"AUTORESCUER"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

**BACHELOR'S DEGREE IN
COMPUTER ENGINEERING**

BY

Sayyad Ayesha Harun Samina	17CO02
Qureshi Ahana Mujib Nazarina	17CO07
Sayyad Rahemat Lazim Reshma	17CO09
Shaikh Iram Fatima Abdul Matin Nasreen	17CO12

UNDER THE GUIDANCE OF
Prof. Kalpana R. Bodke

**DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY**
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2020-2021

AFFILIATED TO

UNIVERSITY OF MUMBAI

A PROJECT II REPORT
ON

"AUTORESCUER"

Submitted to
UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

**BACHELOR'S DEGREE IN
COMPUTER ENGINEERING**

BY

Sayyad Ayesha Harun Samina	17CO02
Qureshi Ahana Mujib Nazarina	17CO07
Sayyad Rahemat Lazim Reshma	17CO09
Shaikh Iram Fatima Abdul Matin Nasreen	17CO12

UNDER THE GUIDANCE OF
Prof. Kalpana R. Bodke

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2020-2021
AFFILIATED TO

UNIVERSITY OF MUMBAI

Anjuman-i-Islam's Kalsekar Technical Campus

Department of Computer Engineering

SCHOOL OF ENGINEERING & TECHNOLOGY

Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

CERTIFICATE

This is certify that the project entitled

"AUTORESCUER"

submitted by

Sayyad Ayesha Harun Samina	**17CO02**
Qureshi Ahana Mujib Nazarina	**17CO07**
Sayyad Rahemat Lazim Reshma	**17CO09**
Shaikh Iram Fatima Abdul Matin Nasreen	**17CO12**

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Engineering) at ***Anjuman-I-Islam's Kalsekar Technical Campus, Navi Mumbai*** under the University of MUMBAI. This work is done during year 2020-2021, under our guidance.

Date: / /

Prof. Kalpana R. Bodke
Project Supervisor

Prof. Kalpana R. Bodke
Project Coordinator

Prof.Tabrez Khan
HOD, Computer Department

Dr.Abdul Razak Honnutagi
Director

External Examiner

Acknowledgements

I would like to take the opportunity to express my sincere thanks to my guide **Prof.Kalpana R. Bodke**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout my project research work. Without his kind guidance & support this was not possible.

I am grateful to him/her for his timely feedback which helped me track and schedule the process effectively. His/her time, ideas and encouragement that he gave is help me to complete my project efficiently.

We would like to express deepest appreciation towards **Dr.Abdul Razak Honnutagi**, Director, AIKTC, Navi Mumbai, **Prof.Tabrez Khan**, Head of Department of Computer Engineering and **Prof. Kalpana R.Bodke**, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

Sayyad Ayesha Harun
Qureshi Ahana Mujib Nazarina
Sayyad Rahemat Lazim Reshma
Shaikh Iram Fatima Abdul Matin Nasreen

Project I Approval for Bachelor of Engineering

This project entitled *P̈roject Title¨* by ***Students Name*** is approved for the degree of ***Bachelor of Engineering in Department of Computer Engineering.***

Examiners

1. ....
2. ....

Supervisors

1. ....
2. ....

Chairman

....

Declaration

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Sayyad Ayesha Harun Samina
17CO02

Qureshi Ahana Mujib Nazarina
17CO07

Sayyad Rahemat Lazim Reshma
17CO09

Shaikh Iram Fatima Abdul Matin Nasreen
17CO12

ABSTRACT

Title: AutoRescuer

In the 21st century everyone is in a rush to reach their destination on time, nothing should be a hindrance to their way of achieving their goal. This is an android application which can be used to track down the nearby mechanic shops (automobile service centers) in case vehicle breaks down or doorstep services are needed. Near to the registered service shops, notifications will be given when the customer comes into some area, if someone from that area is added their shop details to the database then the user will notify those shops. The location of the device is continuously transmitted to the server using GPS. Admin Panel offers the admin login to add shops and uninstall shops and use shop logins. After user registering to this application user can find several shop details which are associated with this application owner.

Keywords:
GPS, API, Multilingual system, service centers(Garage), Mechanic service, workshop, User(vehicle Owner), Automobiles, Geo-Location, Firebase, Notification,OTP, NOC, PUC, OS.

Abbreviations:
GPS(Global Positioning System)
API(Application Programming Interface)
OTP(One Time Password)
NOC(No Objection Certificate)
PUC(Pollution Under Control Certificate)
OS(operating System)

Glossary :
A:
Account: account is a location on a network server used to store a computer username, password, and other information.

Actors:An actor in use case modeling specifies a role played by a user or any other system that interacts with the subject.

Admin:Admin is the role with the highest level of access to your website.

Android:Android is a mobile operating system based on a modified version of the

Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets.

Android Studio:Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

API: a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.

Automobile:The definition of an automobile is a means of transportation that usually has wheels and an engine.

AutoRescuer: An Online Automotive Repair Service System

C:
CSS3:Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

D:
Deployment:Make system available to use.

DoorStep Services:Provide vehicle services at home

DataBase:A database is a collection of information that is organized so that it can be easily accessed, managed and updated.

E:
Economy:the operation of a country's money supply, commercial activities and industry

F:
FireBase:Firebase is a platform developed by Google for creating mobile and web applications.

Function:Function is the most useful part for any programming language because with the help of function developer can define various methods, tasks into a single set of instructions and by calling this function you can perform simple defined task.

G:
Garage:a place where vehicles are repaired and/or petrol is sold

H:
Handset: a mobile phone.

HTML5:HTML5 is a markup language used for structuring and presenting content on the World Wide Web.

I:
infrastructure:the basic systems and services that are necessary for a system

J:
Java:Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.

JavaScript:JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm.

K:
Kitkat:Android 4.4 KitKat is a version of Google's operating system (OS) for smartphones and tablets.

L:
Linux:Linux is a family of open-source Unix-like operating systems based on the Linux kernel.

Live Tracking:Live tracking is a feature that allows other users to track your progress while working out in real-time using a combination of GPS signal and cellular data.

M:
Mechanic:A mechanic is an artisan, skilled trade-person, or technician who uses tools to build, maintain, or repair machinery.

Mobile application:A mobile application, also referred to as a mobile app or simply an app, is a computer program or software application designed to run on a mobile device such as a phone, tablet,or watch.

N:
NoSQL:A NoSQL database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases.

O:
OS:An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs.

V:
Visual Studio:Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps

W:
web Design:Web design encompasses many different skills and disciplines in the production and maintenance of websites.

Windows:Microsoft Windows, commonly referred to as Windows, is a group of several proprietary graphical operating system families, all of which are developed and marketed by Microsoft. Each family caters to a certain sector of the computing industry

Contents

Acknowledgement . . . iii
Project I Approval for Bachelor of Engineering . . . iv
Declaration . . . v
Abstract . . . vi
Table of Contents . . . xii

1 Introduction **2**
1.1 Purpose . . . 2
1.1.1 Problem Statement: . . . 2
1.1.2 Objective: . . . 3
1.2 Project Scope . . . 3
1.3 Project Goals and Objectives . . . 4
1.3.1 Goals . . . 4
1.3.2 Objectives . . . 4
1.4 Organization of Report . . . 4

2 Literature Survey **5**
2.1 Paper Title 1: Tracking of Automobile Service Centers Using Android Application (Visit Mechanic) . . . 5
2.1.1 Advantages of Paper . . . 5
2.1.2 Disadvantages of Paper . . . 5
2.1.3 How to overcome the problems mentioned in Paper . . . 5
2.2 Paper Title 2:Mobile Mechanic – An innovative step towards Digital Automobile Service(WheelCare) . . . 6
2.2.1 Advantages of Paper . . . 6
2.2.2 Disadvantages of Paper . . . 6
2.2.3 How to overcome the problems mentioned in Paper . . . 6
2.3 Paper Title 3: Reduction of servicing and maintenance time of car– a future need . . . 7
2.3.1 Advantages of Paper . . . 7
2.3.2 Disadvantages of Paper . . . 7
2.3.3 How to overcome the problems mentioned in Paper . . . 7
2.4 Technical Review . . . 8
2.4.1 Advantages of Technology . . . 8

2.4.2 Reasons to use this Technology 9

3 Project Planning 10
3.1 Members and Capabilities 10
3.2 Roles and Responsibilities 10
3.3 Assumptions and Constraints 10
3.4 Project Management Approach 11
3.5 Ground Rules for the Project 11
3.6 Project Budget 11
3.7 Project Timeline 12

4 Software Requirements Specification 14
4.1 Overall Description 14
4.1.1 Product Perspective 14
4.1.2 Product Features 14
4.1.3 User Classes and Characteristics 15
4.1.4 Operating Environment 15
4.1.5 Design and Implementation Constraints 15
4.2 System Features 15
4.2.1 System Feature 16
4.3 External Interface Requirements 17
4.3.1 User Interfaces 17
4.3.2 Hardware Interfaces 17
4.3.3 Software Interfaces 18
4.3.4 Communications Interfaces 18
4.4 Nonfunctional Requirements 18
4.4.1 Performance Requirements 18
4.4.2 Safety Requirements 18
4.4.3 Security Requirements 18

5 System Design 19
5.1 System Requirements Definition 19
5.1.1 Functional requirements 19
5.1.2 System requirements (non-functional requirements) 22
5.2 System Architecture Design 22
5.3 Sub-system Development 24
5.3.1 User (Vehicle Owner): 25
5.3.2 Garage Side User 26
5.3.3 Admin: 28
5.4 Systems Integration 28
5.4.1 Class Diagram 29
5.4.2 Sequence Diagram 29

5.4.3 Component Diagram 31
5.4.4 Deployment Diagram 32

6 Implementation 35
6.1 User side function module 35
6.2 Updating text 35
6.3 Location module 45
6.4 Database module 46
6.5 Admin side function module 48

7 System Testing 51
7.1 Test Cases and Test Results 51
7.2 Sample of a Test Case 52
7.2.1 Software Quality Attributes 59

8 Screenshots of Project 60
8.1 Vehicle owner side: 60
8.2 Live tracking: 68
8.3 Garage owner side: 69
8.4 Admin Side: 74

9 Conclusion and Future Scope 76
9.1 Conclusion 76
9.2 Future Scope 77

References 77

List of Figures

3.1 Project Task Assigned to members 12
3.2 Project Timeline 12
3.3 Project Timeline continuation 13

5.1 Use Case Diagram 20
5.2 Level Zero DFT 21
5.3 Level ONE DFT 21
5.4 Level Two DFT 22
5.5 System Architecture of User 23
5.6 System Architecture of Garage Side User 23
5.7 System Architecture of Admin 24
5.8 User Profile Module 25
5.9 User Function Module 25
5.10 User Request Management Module 26
5.11 Garage Profile Module 26
5.12 Garage Function Module 27
5.13 Garage Request Management Module 27
5.14 Admin Profile Module 28
5.15 Admin Function Module 28
5.16 Class Diagram 29
5.17 Sequence Diagram of User 30
5.18 Sequence Diagram of Garage Side User 30
5.19 Sequence Diagram of Admin 31
5.20 Component diagram of user side 31
5.21 Component diagram of garage side side 32
5.22 Component diagram of admin side 32
5.23 deployment diagram of user side 33
5.24 deployment diagram of garage side user 33
5.25 deployment diagram of admin side 34

6.1 IMAGE CAPTION 35
6.2 Map with marker 45
6.3 updating data module 46

8.1 Application information and user login 60

8.2 registration of a user 61
8.3 Vehicle user functions 61
8.4 Locating on map 62
8.5 Vehicle details selection 62
8.6 Garage selection 63
8.7 Service selection 63
8.8 Service request status 64
8.9 Notification status 64
8.10 Garage selection for spare parts 65
8.11 spare parts information 65
8.12 order details and order status 66
8.13 Enable live tracking 66
8.14 Live tracking status 67
8.15 Live tracking module 68
8.16 Garage registration process 69
8.17 Garage information 69
8.18 Garage function module 70
8.19 Garage services with cost 71
8.20 User's service request status 71
8.21 Booking history and spare parts with cost 72
8.22 spare parts with cost 73
8.23 Booking history and spare parts 73
8.24 Sign-up page for Admin 74
8.25 Login page for Admin 74
8.26 Login via Gmail 75
8.27 Firebase database 75

List of Tables

3.1 Table of Capabilities 10
3.2 Table of Responsibilities 10

Chapter 1

Introduction

The title of our project is AUTORESCUER, it is an online mechanic booking system for Androids. This project concerned about developing an online mechanic booking system that will be used for booking mechanic from nearby garages. As we are moving towards the technology dominant age and the financial status of the person is also increasing by time, the power of both (technology + financial status) are giving wings to the people not only to realize but also to fulfill their dreams. AutoRescuer, an online mechanic booking system is essential for booking mechanic from nearby garage which will track of customer location as well as garage location, booking details, payment details, booking history, etc.
The main objective of the Android Project on Online mechanic booking system is to manage the details of garage owner and garage, vehicle owner, booking details. Admin will manage all the information about garage and garage owner.

The purpose of the project is to build an application program to reduce the time required to provide vehicle services and mechanic will get a job opportunity. AutoRescuer Aims to out rule traditional ineffective and inefficient ways to request services which is mostly offline. Now user can easily book and get a mechanic in a short time period and get a perfect one. This will reduce the time of the customer and they will have a worthy mechanic.

1.1 Purpose

1.1.1 Problem Statement:

Considering a real-life problem where a person faces troubles when their car breaks down in the middle of the road, can really become a major problem.This can be solved by using our application AutoRescuer which can help the users to find nearby location of service or a repairing center. Here the proposed system provides a mobile application for Online Automobile services . It will enable any vehicle user to search and communicate with any garage in the vicinity. The system uses Firebase for storage which is a technology provided by Google. Firebase uses NoSQL for storing the database of the application which includes users detail. Whenever the user accesses the System, users location is derived from their device and then the user is provided with the locations of the nearby garages. The user is then free to select a garage by his/her choice. After a garage is selected by the user, users information like location is sent to the chosen garage provided that mechanic is not busy and they can accept the users request to provide the service and payment will be COD(Cash On Delivery).The user can book a car and bike service from the proposed system in just few steps. Vehicle breakdown? Tyre puncture?The system has got it covered! Book mechanics Anytime! Anywhere!

1.1.2 Objective:

The identified challenges motivate to bring up of a solution to all the problems stated in the above problem statement section. This project aims to implement an android application with the help of Android studio and Firebase.

The core objectives which have been designated as fundamental to the project are:
•Identify and understand Infrastructure and working of Mechanics.
•How Mechanic approach to the vehicle user to solve problem.
•Try to connect two different Users (vehicle user and garage side user) on the same platform.
• Help user to take care of their vehicles at critical times.
•Design a website to Admin to handle all the users data.
•Design an implement User Interface which will be easy for user to understand.

1.2 Project Scope

The goal of this project is to produce an interactive and entertaining application for the Android marketplace. Auto Rescuer is composed of two main components: a client-side application which will run on Android handsets, and a server-side application which will support and interact with various client-side features. The system is designed to provide features of all the vehicles, services provided by the service centers, locations of all the service centers in the vicinity etc. The above proposed model is easy to implement considering the available technology infrastructure. The model is simple, secure and scalable. The proposed model is based on serial communication.

1.3 Project Goals and Objectives

1.3.1 Goals

Given below are the project goals to achieve the desired outcome:
- Do the best we can with what we have been given.
- Ensure a high quality of design and design documentation.
- To finish project within the scheduled time and budget.
- To meet the requirements the goal that were set for the project at the start.

1.3.2 Objectives

Following are the objectives of the proposed approach and this thesis work:
- To find out the different factors responsible for effecting servicing and maintenance process of a vehicle.
- To make an online mechanic booking system.
- To offer easy, simple and quick booking experiences for vehicle owner.
- To offer a responsive booking engine that offers a great booking experience across android devices.
- Easy maintenance of garage, user record, user login, etc.
- To give a job opportunity to a local but worthy mechanic.
- Google Map navigation to customer location and Live tracking.
- To grab real time information of garages.

1.4 Organization of Report

The report is organized as follows : The introduction is given in Chapter 1. It describes the fundamental terms used in this project. It describes the Problem Statement, Goal, Objectives and scope of this project. The Chapter 2 describes the review of the relevant various techniques in the literature systems. It describes the pros and cons of each technique with how to overcome those cons using new technology. The project planning includes members and capabilities of this project ,roles and responsibilities of each member,Budget of Project and Project timeline is describe in Chapter 3. The Chapter 4 describes System Requirements specification such as Functional and Nonfunctional Requirements of project. Along with this it also explain features of system and constraints of system. The Chapter 5 includes Design Information with Class Diagram, Sequence Diagram , Component Diagram and System Architecture. Implementation of each module is explained in Chapter 6. Chapter 7 shows final Test Cases and Test Results. Chapter 8 includes Screenshot of outputs and Conclusion and Future Scope of Project is described in Chapter 9

Chapter 2

Literature Survey

2.1 Paper Title 1: Tracking of Automobile Service Centers Using Android Application (Visit Mechanic)

This is an android application which can be used to track down the mechanic shops (automobile service centers). After getting the location of the service center, the shop keeper will give the details so that the location of the shop and details of the shop will be sent to the database. Near to the registered service shops, notifications will be given when the customer comes into some area, if someone from that area is added their shop details to the database then the user will notify those shops. The location of the device is continuously transmitted to the server using GPS and LBS. Admin Panel offers the admin login to add shops and uninstall shops and use shop logins. After user registering to this application user can find several shop details which are associated with this application owner. From the details, the user can get directions to the services center.

2.1.1 Advantages of Paper

a. Easy to track mechanic shop using GPS

b. User can get direction to the service center

2.1.2 Disadvantages of Paper

a. This system logic recommends the near by services center's address to the user even if the center is not opened.

b. The liberty to check booking history of themselves is absent in this particular system.

2.1.3 How to overcome the problems mentioned in Paper

a. The system should be able to provide information related to time to the user, so that they can check whether the garage is opened or closed

b. The system should allow users to view the history of their bookings. The booking history feature also acts as a reference.

2.2 Paper Title 2:Mobile Mechanic – An innovative step towards Digital Automobile Service(WheelCare)

The application is subdivided into two main sections namely the Mechanic's portal and the User's portal. Both these portals would require the respective user to either login or register themselves up before using the application. The location can be provided or obtained using the GPS available in mobile phones. The User's portal would display a map with the current location of themselves along with all the available repair shops around them, which shall be consistently updated using the plethora of location services available on Microsoft Azure Cloud. User can get detailed information about these shops along with a route to reach the desired mechanic shop, using Google Map API.

2.2.1 Advantages of Paper

a. Goggle Map API is used to reach out mechanic shops.

b. Repair shops are consistently updated using the plethora of location services available on Microsoft Azure Cloud.

2.2.2 Disadvantages of Paper

a. Mobile Mechanic was designed for Cars only.

b. User verification process is absent in this system.

c. In case User wants to update there personal information this system does not allowed to do so.

2.2.3 How to overcome the problems mentioned in Paper

a. The system should provide services to all types of vehicle.

b. User verification process can be done trough OTP and Garage verification can done via uploading document such as NOC of garage.

c. The system should allow user to update the information.

2.3 Paper Title 3: Reduction of servicing and maintenance time of car– a future need

The objective of this study was to find the different factors responsible for effecting the servicing and maintenance process of a car and to look for the opportunities to reduce time required for it. This study also attempts to suggest some probable solution based upon the research findings. To gather the data we took up a research in few areas of Mumbai (India) with some user groups like mechanics, car owners, drivers, managers etc. For this research, methodology adopted was to study how mechanics approach to find the problems, to understand how the cars are increased on the Indian roads and to understand the car owner's behavior.

2.3.1 Advantages of Paper

This paper is very helpful to understand factors which are responsible for effecting the servicing and maintenance process of a car

a. Different methodology is explain to overcome problems faced by vehicle owner.

b. It also help us to understand how the cars are increased on the Indian roads and to understand the car owner's behavior.

2.3.2 Disadvantages of Paper

The people have more disposable income as the economy is growing. Car Finance options available from financial body at reasonable rate of interest. Improvement in highway infrastructure. Changing lifestyle. People started preferring small cars as their family vehicle. Due to all those points mentioned above, the growth of four wheeler vehicles will be rapid, because of which the new challenges may arise. Few are listed below:

a. Maintaining traffic on the road will be a huge challenge.

b. Pollution will increase, which may create health issues.

c. Long queues at a service station for maintaining servicing of the car.

2.3.3 How to overcome the problems mentioned in Paper

With respect to the research and finding some probable solutions are listed below by which servicing and maintenance time of cars at mechanic shop can be reduced.

a. By Interactive manual

b. Conduct workshop for the car owner

c. Mobile application

2.4 Technical Review

Android is a software package and linux based operating system for mobile devices such as tablet computers and smartphones. It is developed by Google and later the OHA (Open Handset Alliance). Java language is mainly used to write the android code even though other languages can be used.
We have used Java Language to develop this application. The goal of android project is to create a successful real-world product that improves the mobile experience for end users. There are many code names of android such as Lollipop, Kitkat, Jelly Bean, Ice cream Sandwich, Froyo, Ecliar, Donut.
To build an android application we have used android studio.Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

Firebase is a Backend-as-a-Service(BaaS). Google Firebase is Google-backed application development software which allows developers to develop Android, IOS, and Web apps. For reporting and fixing app crashes, tracking analytics, creating marketing and product experiments, firebase provides several tools.

To build an website for admin we have used HTML,Css and Ajax technologies.HTML is an acronym which stands for Hyper Text Markup Language which is used for creating web pages and web applications.It is platform-independent because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.
CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces.
AJAX is an acronym for Asynchronous JavaScript and XML. It is a group of inter-related technologies like JavaScript, DOM, XML, HTML/XHTML, CSS, XML-HttpRequest etc.
JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive.

2.4.1 Advantages of Technology

a. Android Studio:
It is open source software.
No need to pay, free of charge product.
It supports big community of other android developers.
Provide Java to Kotlin code auto translation
Debug mode is excellent.

b. Java: The java has excellent multi-functional IDE. Java is an object-oriented program and it is easy to reuse the code.

c. FireBase:
Reliable Extensive Databases
Free Multi-Platform Firebase Authentication
Firebase Testing Services to Improve App Quality

d. HTML and Css: It is a markup language, so it provides a flexible way to design web pages along with the text.It facilitates programmers to add a link on the web pages (by html anchor tag), so it enhances the interest of browsing of the user. With the help of css we can add new looks to your old HTML documents.

e. Ajax and JavaScript: AJAX allows you to send and receive data asynchronously without reloading the web page. So it is fast.
Speed, simplicity, popularity and Interoperability are the advantages of javascript.

2.4.2 Reasons to use this Technology

a. Reason we are making android application is because the number of mobile users today greater than the number of desktop users. As user spend more times on mobile applications. Mobile application provides better personalization.

b. Java: Java is robust. Java is statistically built language and it is easy to track the error in Java.

c. Database we are using here is Firebase because, Firebase is a cloud service (NoSql) suitable for real time applications. It is quit fast in response in comperison to MySql. Data structure used by NoSql databases are more flexible and scalable than RDBMS.

d. HTML and Css: HTML is a very easy and simple language. It can be easily understood and modified.It is very easy to make an effective presentation with HTML because it has a lot of formatting tags.
We can completely change the look of your website with only a few changes in CSS code.

e. Ajax and JavaScript: Ajax adds a lot of value to UX. If done right, the user gets a great feel when using the site, and better data usage because it doesn't load the whole page everytime.
JavaScript gives web pages interactive elements that engage a user.

Chapter 3

Project Planning

3.1 Members and Capabilities

Table 3.1: Table of Capabilities

SR. No	Name of Member	Capabilities
1	Sayyad Ayesha Harun	UI Design
2	Qureshi Ahana Mujib	Back-End, UI Design
3	Sayyad Rahemat Lazim	Website Front-End
4	Shaikh Iram Fatima Abdul Matin	Back-End,Front-End

Work Breakdown Structure

a. All of the members are equally important in developing the project.

b. We work on a different part of the project based on one's capability.

c. Firstly we came up with documentation, And based on the documentation we set our goal and created a blueprint.

d. We then started going hands-on with the project to develop it according to the flow as decided earlier.

3.2 Roles and Responsibilities

Table 3.2: Table of Responsibilities

SR. No	Name of Member	Role	Responsibilities
1	Qureshi Ahana Mujib	Team Leader	DataBase, UI Design
2	Sayyad Ayesha Harun	Team Member	UI Design
3	Sayyad Rahemat Lazim	Team Member	Website Front-End
4	Shaikh Iram Fatima Abdul Matin	Team Member	Back-End, Front-End

3.3 Assumptions and Constraints

a. User of the app Should know how to use browser and Internet.

b. User should know how to deal with live tracking.

c. User should know working of application.

d. Garage owner needs to register the garage in order to provide services.

3.4 Project Management Approach

a. Planning of project.

b. Defining the scope of the project.

c. Estimation of time and It's management.

d. Creating Gantt Charts and properly assigning tasks to members.

e. Reporting the progress of project with the guide.

3.5 Ground Rules for the Project

a. Properly planning and gathering relevant information is very important.

b. Developing a Blueprint of the project and work accordingly.

c. All the members should report to the guide whenever required.

d. Setting up small goals every week.

e. Achieving the small goal within that span of time.

f. Keeping tracks of the progress towards project.

g. Participate in meeting.

h. Inform the leader about unavailability.

3.6 Project Budget

a. It is a light project.

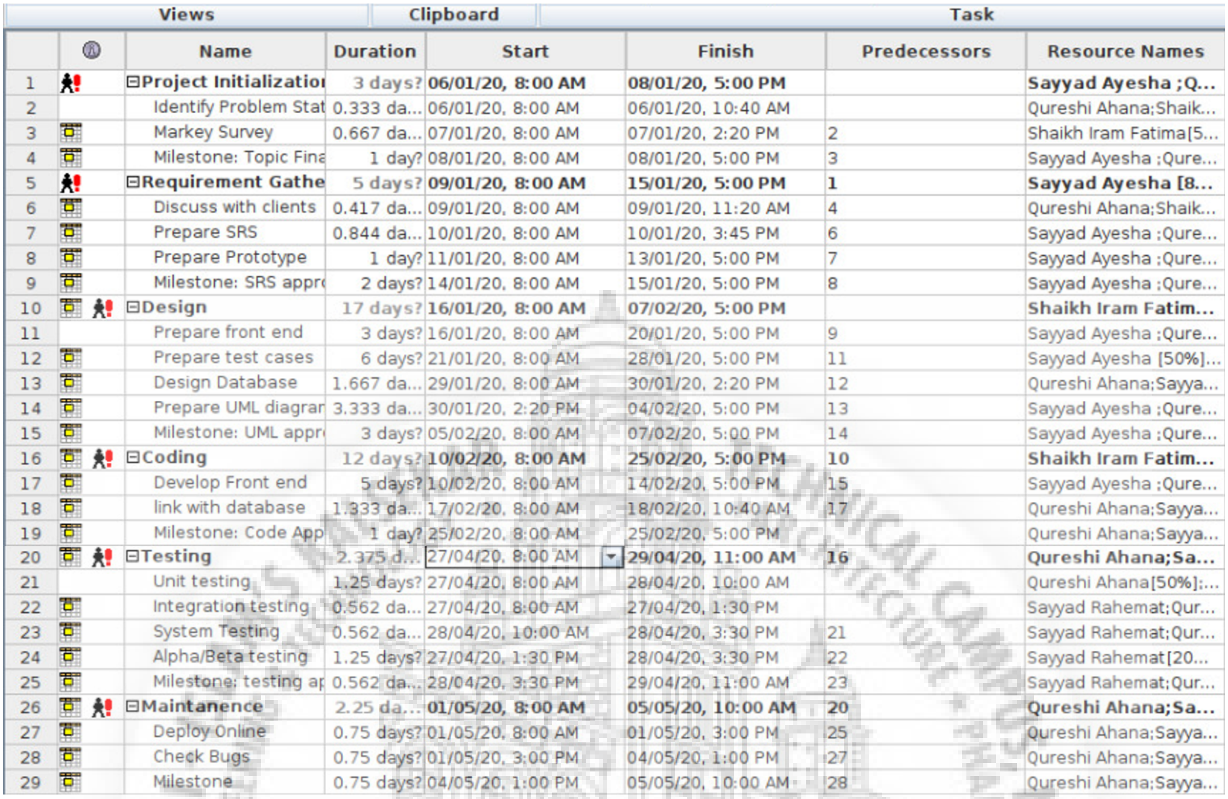
b. Cost of the project is very low and efficient.

c. Free version of Android studio is available.

d. Firebase is open source.

e. HTML, Css are also free to use.

3.7 Project Timeline



		Name	Duration	Start	Finish	Predecessors	Resource Names
1		⊟Project Initializatioı	3 days?	06/01/20, 8:00 AM	08/01/20, 5:00 PM		Sayyad Ayesha ;Q...
2		Identify Problem Stat	0.333 da...	06/01/20, 8:00 AM	06/01/20, 10:40 AM		Qureshi Ahana;Shaik...
3		Markey Survey	0.667 da...	07/01/20, 8:00 AM	07/01/20, 2:20 PM	2	Shaikh Iram Fatima[5...
4		Milestone: Topic Fina	1 day?	08/01/20, 8:00 AM	08/01/20, 5:00 PM	3	Sayyad Ayesha ;Qure...
5		⊟Requirement Gathe	5 days?	09/01/20, 8:00 AM	15/01/20, 5:00 PM	1	Sayyad Ayesha [8...
6		Discuss with clients	0.417 da...	09/01/20, 8:00 AM	09/01/20, 11:20 AM	4	Qureshi Ahana;Shaik...
7		Prepare SRS	0.844 da...	10/01/20, 8:00 AM	10/01/20, 3:45 PM	6	Sayyad Ayesha ;Qure...
8		Prepare Prototype	1 day?	11/01/20, 8:00 AM	13/01/20, 5:00 PM	7	Sayyad Ayesha ;Qure...
9		Milestone: SRS appr	2 days?	14/01/20, 8:00 AM	15/01/20, 5:00 PM	8	Sayyad Ayesha ;Qure...
10		⊟Design	17 days?	16/01/20, 8:00 AM	07/02/20, 5:00 PM		Shaikh Iram Fatim...
11		Prepare front end	3 days?	16/01/20, 8:00 AM	20/01/20, 5:00 PM	9	Sayyad Ayesha ;Qure...
12		Prepare test cases	6 days?	21/01/20, 8:00 AM	28/01/20, 5:00 PM	11	Sayyad Ayesha [50%]...
13		Design Database	1.667 da...	29/01/20, 8:00 AM	30/01/20, 2:20 PM	12	Qureshi Ahana;Sayya...
14		Prepare UML diagrar	3.333 da...	30/01/20, 2:20 PM	04/02/20, 5:00 PM	13	Sayyad Ayesha ;Qure...
15		Milestone: UML appr	3 days?	05/02/20, 8:00 AM	07/02/20, 5:00 PM	14	Sayyad Ayesha ;Qure...
16		⊟Coding	12 days?	10/02/20, 8:00 AM	25/02/20, 5:00 PM	10	Shaikh Iram Fatim...
17		Develop Front end	5 days?	10/02/20, 8:00 AM	14/02/20, 5:00 PM	15	Sayyad Ayesha ;Qure...
18		link with database	1.333 da...	17/02/20, 8:00 AM	18/02/20, 10:40 AM	17	Qureshi Ahana;Sayya...
19		Milestone: Code App	1 day?	25/02/20, 8:00 AM	25/02/20, 5:00 PM		Qureshi Ahana;Sayya...
20		⊟Testing	2.375 d...	27/04/20, 8:00 AM	29/04/20, 11:00 AM	16	Qureshi Ahana;Sa...
21		Unit testing	1.25 days?	27/04/20, 8:00 AM	28/04/20, 10:00 AM		Qureshi Ahana[50%];...
22		Integration testing	0.562 da...	27/04/20, 8:00 AM	27/04/20, 1:30 PM		Sayyad Rahemat;Qur...
23		System Testing	0.562 da...	28/04/20, 10:00 AM	28/04/20, 3:30 PM	21	Sayyad Rahemat;Qur...
24		Alpha/Beta testing	1.25 days?	27/04/20, 1:30 PM	28/04/20, 3:30 PM	22	Sayyad Rahemat[20...
25		Milestone: testing ap	0.562 da...	28/04/20, 3:30 PM	29/04/20, 11:00 AM	23	Sayyad Rahemat;Qur...
26		⊟Maintanence	2.25 da...	01/05/20, 8:00 AM	05/05/20, 10:00 AM	20	Qureshi Ahana;Sa...
27		Deploy Online	0.75 days?	01/05/20, 8:00 AM	01/05/20, 3:00 PM	25	Qureshi Ahana;Sayya...
28		Check Bugs	0.75 days?	01/05/20, 3:00 PM	04/05/20, 1:00 PM	27	Qureshi Ahana;Sayya...
29		Milestone	0.75 days?	04/05/20, 1:00 PM	05/05/20, 10:00 AM	28	Qureshi Ahana;Sayya...

Figure 3.1: Project Task Assigned to members

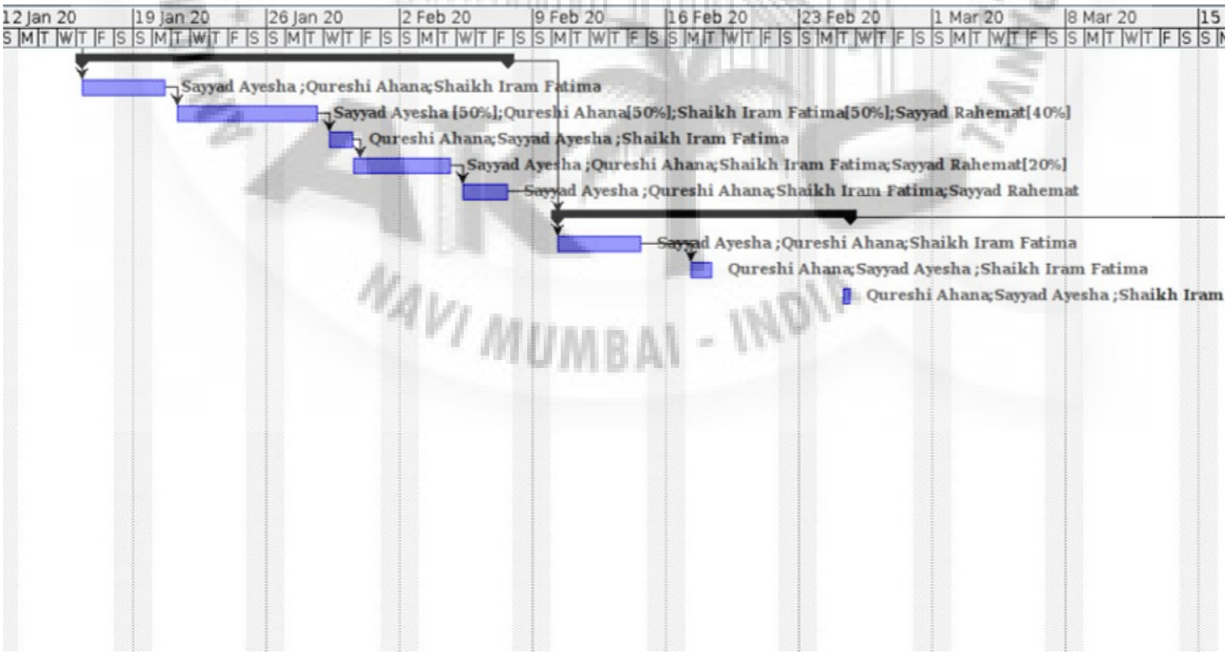


Figure 3.2: Project Timeline

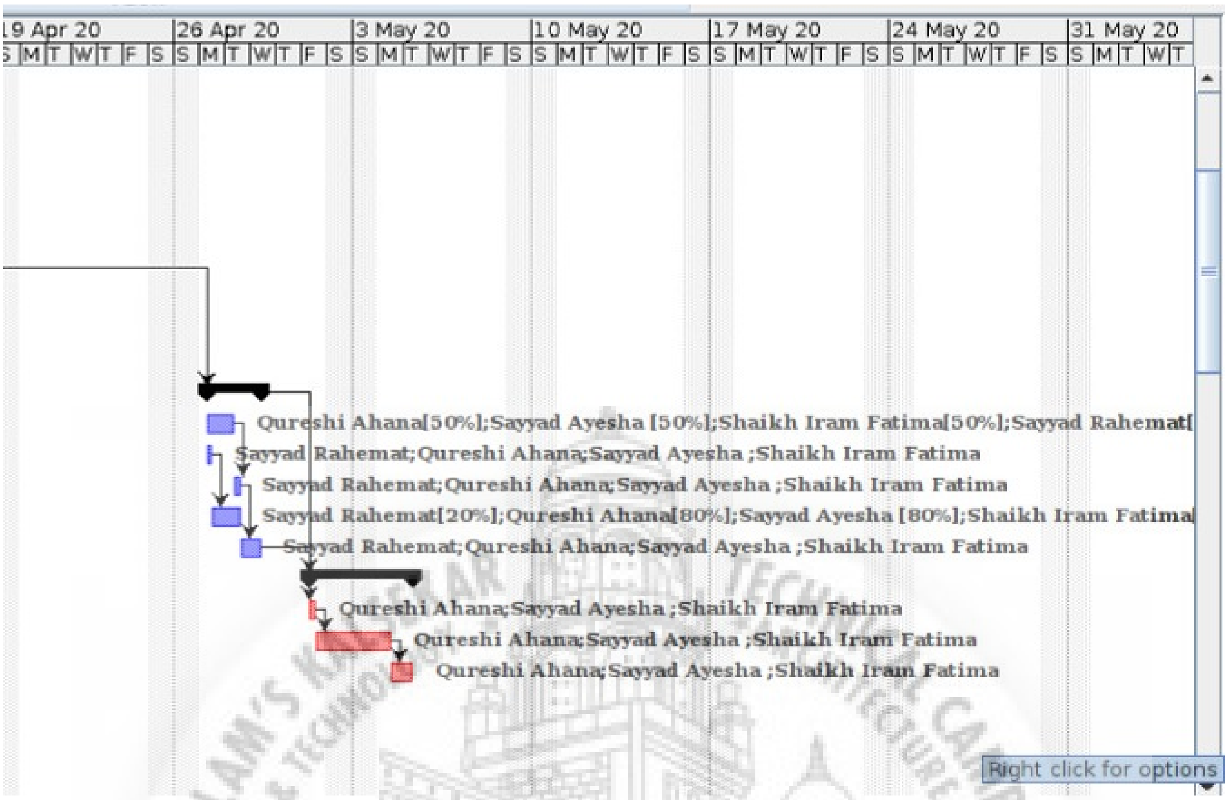

19 Apr 20
26 Apr 20
3 May 20
10 May 20
17 May 20
24 May 20
31 May 20
S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T
Qureshi Ahana[50%];Sayyad Ayesha [50%];Shaikh Iram Fatima[50%];Sayyad Rahemat[
Sayyad Rahemat;Qureshi Ahana;Sayyad Ayesha ;Shaikh Iram Fatima
Sayyad Rahemat;Qureshi Ahana;Sayyad Ayesha ;Shaikh Iram Fatima
Sayyad Rahemat[20%];Qureshi Ahana[80%];Sayyad Ayesha [80%];Shaikh Iram Fatima
Sayyad Rahemat;Qureshi Ahana;Sayyad Ayesha ;Shaikh Iram Fatima
Qureshi Ahana;Sayyad Ayesha ;Shaikh Iram Fatima
Qureshi Ahana;Sayyad Ayesha ;Shaikh Iram Fatima
Qureshi Ahana;Sayyad Ayesha ;Shaikh Iram Fatima
Right click for options

Figure 3.3: Project Timeline continuation

Chapter 4

Software Requirements Specification

4.1 Overall Description

This section gives a scope description and overview of everything included in this SRS document. It provides all the detail about SRS including its different functions, role of particular users, requirement of different hardware and software required.

4.1.1 Product Perspective

The Product is open source. It is an android application for vehicle user and Garage side user and website for admin to manage both users data. The goal of this project is to produce an interactive and entertaining application for the Android marketplace.
AutoRescuer is composed of two main components: a client-side application which will run on Android handsets, and a server-side application which will support and interact with various client-side features. The system is designed to provide features of all the vehicles, services provided by the service centers, locations of all the service centers in the vicinity etc. The above proposed model is easy to implement considering the available technology infrastructure. The models is simple, secure and scalable. The proposed model is based on serial communication.

4.1.2 Product Features

Their are four major system in this product:

a. For Vehicle User home Page derived current location of system and get vehicle breakdown related information. Next thing is to select service and book mechanic

b. For Vehicle User live tracking is available. Location can be shared if user wishes to share.

c. For Garage Side User first upload NOC, then create profile to get registered.Home page content services provided by garage.

d. Admin is the super-user who will manage both the users via a website.

4.1.3 User Classes and Characteristics

The project is an Android Application for vehicle user and garage user.User of project includes Vehicle user and Garage side user. A website is created to handle both the users data by super-user (admin).
All the user should have knowledge of Internet and should have knowledge about how to use an android phone. Admin should know how to use computer.

4.1.4 Operating Environment

Software Requirements:
• Operating System: Linux or Windows(7 and above)
• Adroid Studio • FireBase

Hardware Requirements:
• Processor i5
• Memory 8GB
• Hard disk 1TB

Usage Requirements:
• An Android Mobile API 19 and above(kitkat)
• Internet connectivity

4.1.5 Design and Implementation Constraints

The product is made using android studio hence, only android phone users can use this application. User may access the product using any android device by using correct OTP send on registered mobile number.
The information of all the users, booking histories, garage data must be stored in database. Internet connectivity is the main source to use the product.
Admin should use correct username and password to manage databse.

4.2 System Features

Following are the mode of operation provides by the system:
For vehicle user :
Register/Login through OTP, Create and Edit his/her profile, Locate oneself on the map,Choose the required service, Fill in the vehicle details, Select the garage on the basis of service cost and distance.

For Garage side user:
Register/Login through OTP, Validate his/her garage through NOC document,Create

and Edit his/her profile, Locate oneself on the map, Fill in the garage details(can update as well), Receive the user's request,Accept/Reject the request.

For Admin:
Can perform CRUD operation on the data of User and Garage owner.

4.2.1 System Feature

Locating oneself on Map is the main feature of the system. On vehicle user side after locating themselves on map nearby garages will be filtered and shown to the user. Garage user needs to locate themselves so that vehicle user can book services. Rest of the features are easy to understand.

Description and Priority

The highest priority is given to the locating function. As it is very useful for vehicle user to search and select nearby garages.
Second prior feature is status of request service.When vehicle user confirm the booking slot, message is directly send to the registered mobile number of garage user. Garage side user can accept or reject the request and the respective notification will be send to the vehicle user.

Stimulus/Response Sequences

Vehicle User side
Stimulus: Register or Login via OTP.
Response: The User is registered or logged in and all the available features are visible.
Stimulus: User clicks on Map to locate themselves .
Response: User location is stored in database.
Stimulus: User clicks on select services and then select services from the nearby garage.
Response: The services and the garage related data is stored in database.
Stimulus: User clicks on confirm booking
Response: The notification will be pop-up on garage side user.
Stimulus: Receives service accepted or rejected message.
Response: Garage side user can accept or reject the request if service request is accepted then mechanic will be sent to users location.

Garage User side
Stimulus: Register or Login via OTP.
Response: The User is registered or logged in and needs to uplaod NOC .
Stimulus: User clicks on locate on map then Add services and other garage related

data.
Response: Location and Services provided by garage with cost and the timing of the garage id stored in database.
Stimulus: Receive user request for mechanic or service.
Response: Garage user get a direct message on registered mobile with data such as user location, requested services etc.
Stimulus: User clicks on accept services
Response: Notification will be sent to vehicle user with garage details and cost of the services. **Stimulus:** User clicks on reject services
Response: Rejected service notification will be sent to the vehicle user

Admin
Stimulus: Login through website.
Response: CRUD operations can be performed.

Functional Requirements

REQ-1: Users are limited to Android handsets.
REQ-2: Admin is limited to Linux or Windows(7 and above).
REQ-3: Access to the Databases.
REQ-4: Access to Internet.

4.3 External Interface Requirements

4.3.1 User Interfaces

Android provides a variety of pre-built UI components such as structured layout objects and UI controls that allow to build the graphical user interface for app. Android also provides other UI modules for special interfaces such as dialogs, notifications,navigation drawer, and menus.
The navigation drawer consist of user profile details which can be easily updated,booking history of services. The notification pop-up when user request for the services and also messages are sent to the respected users.
For the admin website is build.the GUI is very simple. Home page is nothing but the firebase console which contain all users data

4.3.2 Hardware Interfaces

This application works on android handset with API 19 and above(kitkat) version. And website can be run on any hardware.

4.3.3 Software Interfaces

Since this application is a mobile application, it will only need an Android kitkat version API 19 or higher in order to perform. Database is maintained in Firebase.

4.3.4 Communications Interfaces

The application uses internet to communicate with user.Also offline messages related to the service request are sent to the user. In case of any difficulties while using application user can contact through whatsApp or call. In case of emergency vehicle user can send their live location to the known people. On Admin side the product is a light web, there is no such large communication in the system. Only Databases access, that also done locally. Also https standard is used in-order to gain the access to the browser.

4.4 Nonfunctional Requirements

4.4.1 Performance Requirements

Performance of overall system is very efficient and well optimize. The time taken to Show various Garages on map would take 3-5 sec. Process and everything is well organized. The vehicle user tracker also take approx 3-4 sec to show in map. The messages related to services will be delivered to the users in a very short time.

4.4.2 Safety Requirements

Login and sign up must be authenticated for the pre-existing users. Data of every user should maintain.

4.4.3 Security Requirements

Sign In: Only registered user can access his/her account. Sign Up: No duplicate of the data of the user should be there. Sent and received message should be transferred via Reliable Data Transfer connection.

Chapter 5

System Design

5.1 System Requirements Definition

System requirement definitions specify what the system should do, its functionality and its essential and desirable system properties. The techniques applied to elicit and collect information in order to create system specifications and requirement definitions involve consultations, interviews, requirements workshop with customers and end users. The objective of the requirements definition phase is to derive the two types of requirement:

5.1.1 Functional requirements

They define the basic functions that the system must provide and focus on the needs and goals of the end users.

Use-case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

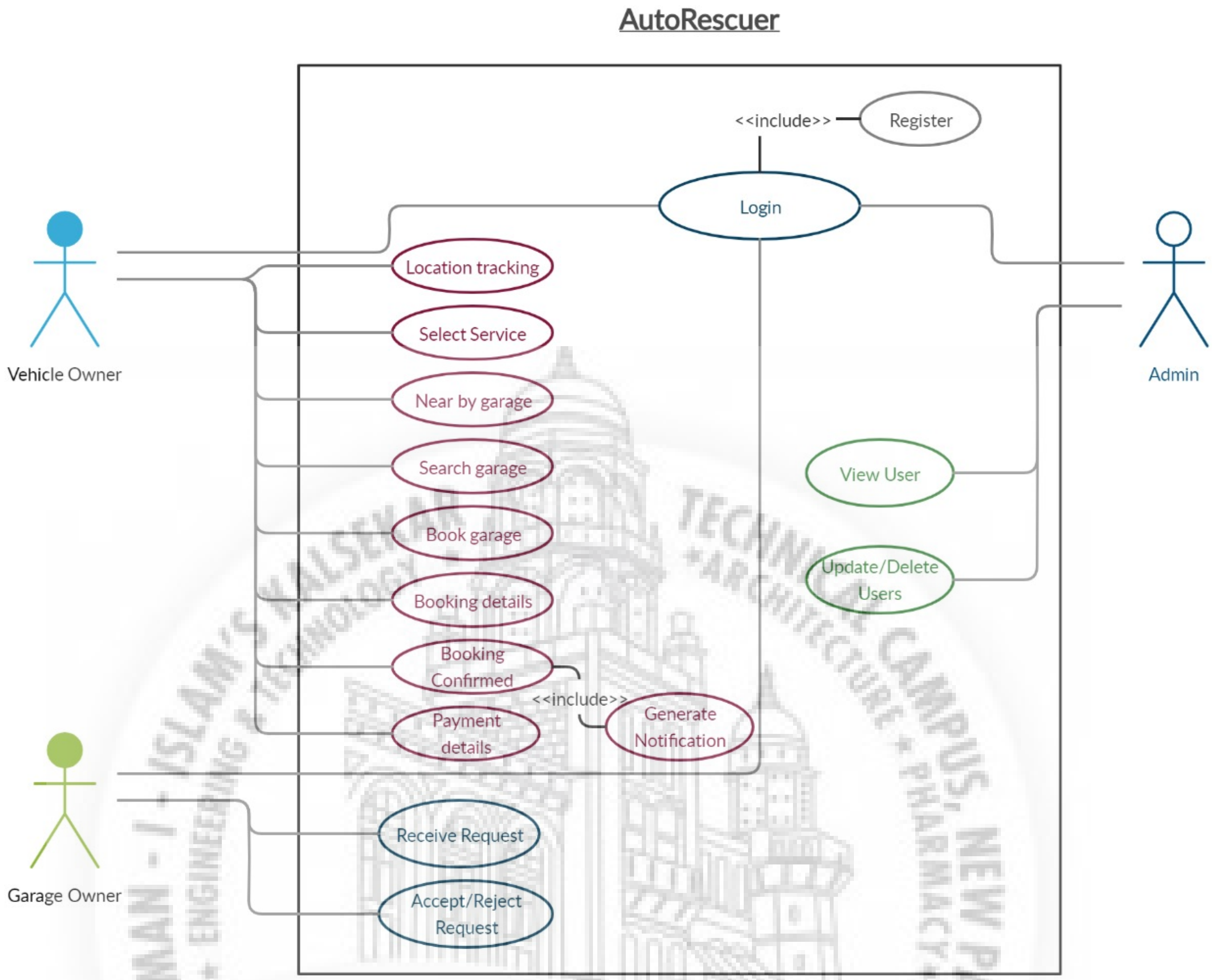


Figure 5.1: Use Case Diagram

Data-flow Diagram

A data-flow diagram is a way of representing a flow of a data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. Given below is Level 0 Level 1 and Level 2 DFD of system.

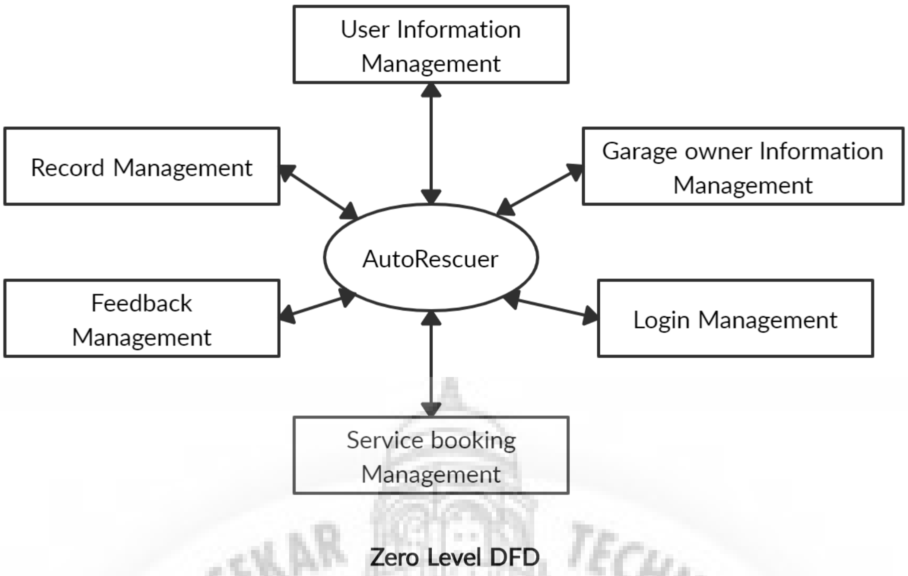

User Information Management
Record Management
Garage owner Information Management
AutoRescuer
Feedback Management
Login Management
Service booking Management
Zero Level DFD

Figure 5.2: Level Zero DFT

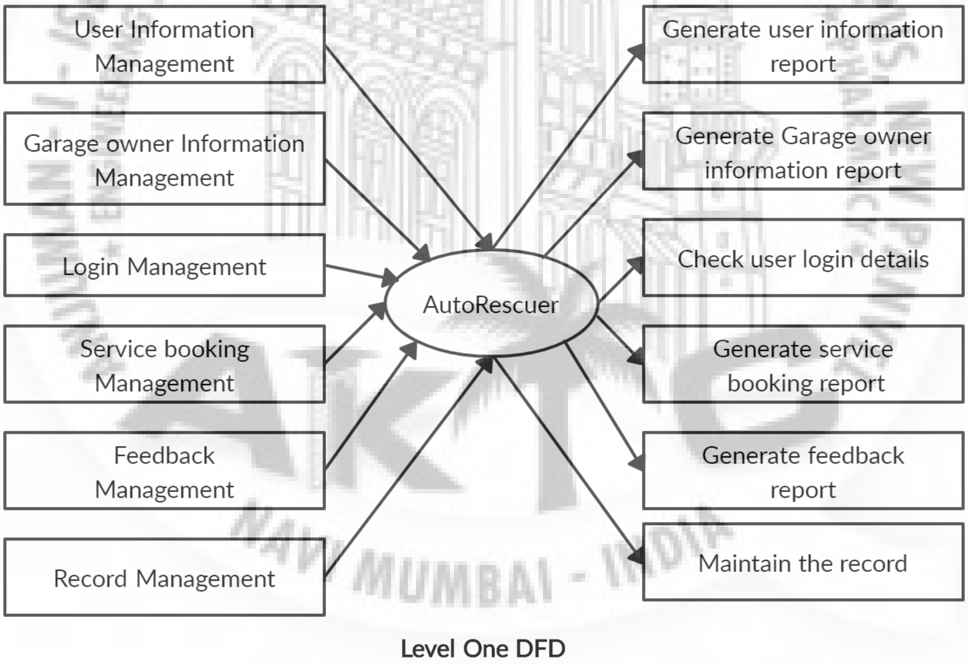

User Information Management
Garage owner Information Management
Login Management
Service booking Management
Feedback Management
Record Management
AutoRescuer
Generate user information report
Generate Garage owner information report
Check user login details
Generate service booking report
Generate feedback report
Maintain the record
Level One DFD

Figure 5.3: Level ONE DFT

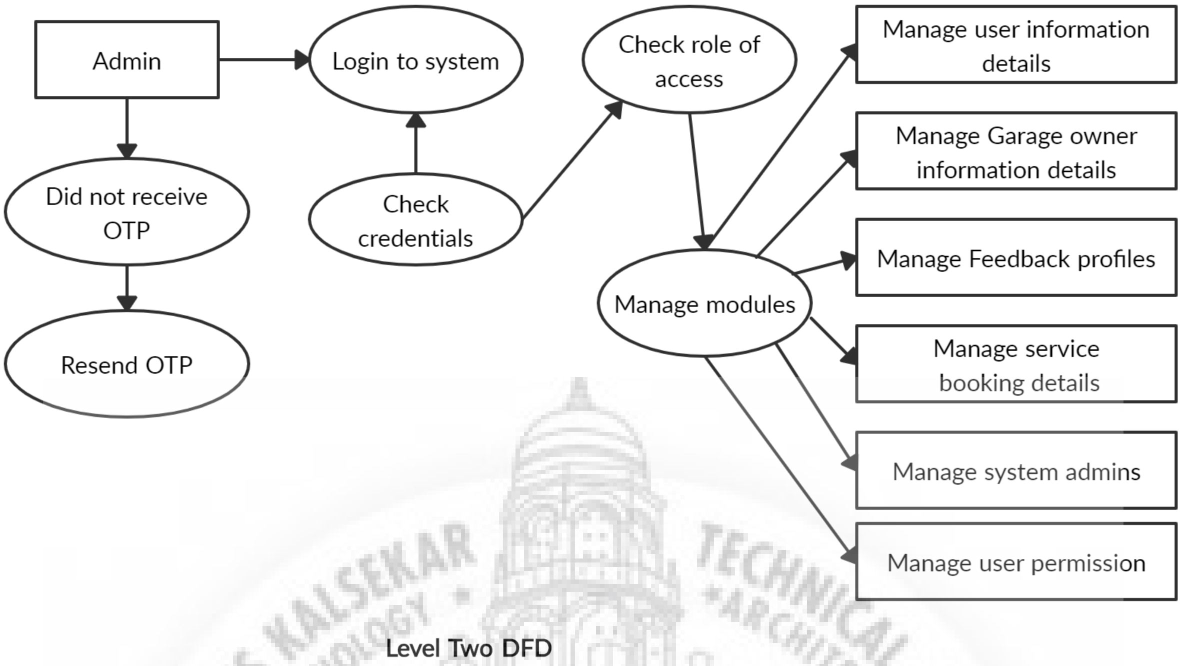


Figure 5.4: Level Two DFT

5.1.2 System requirements (non-functional requirements)

These are non-functional system properties such as availability, performance and safety etc. They define functions of a system, services and operational constraints in detail.

a. Usability - Application implementation is feasible using technologies that are accessible to the end-users.

b. Portability - The interfaces are compatible with Web View and Mobile view.

c. Performance Efficiency -Application is able to perform well in a proper time constraint.

d. Multi User System -Application is able to consider the presence of more than one user in the same environment. All the features of the system operates properly for all users and provides proper transparency.

e. Time Efficiency - Time taken for the executing of system is less.

Database Schema/ E-R Diagram

5.2 System Architecture Design

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

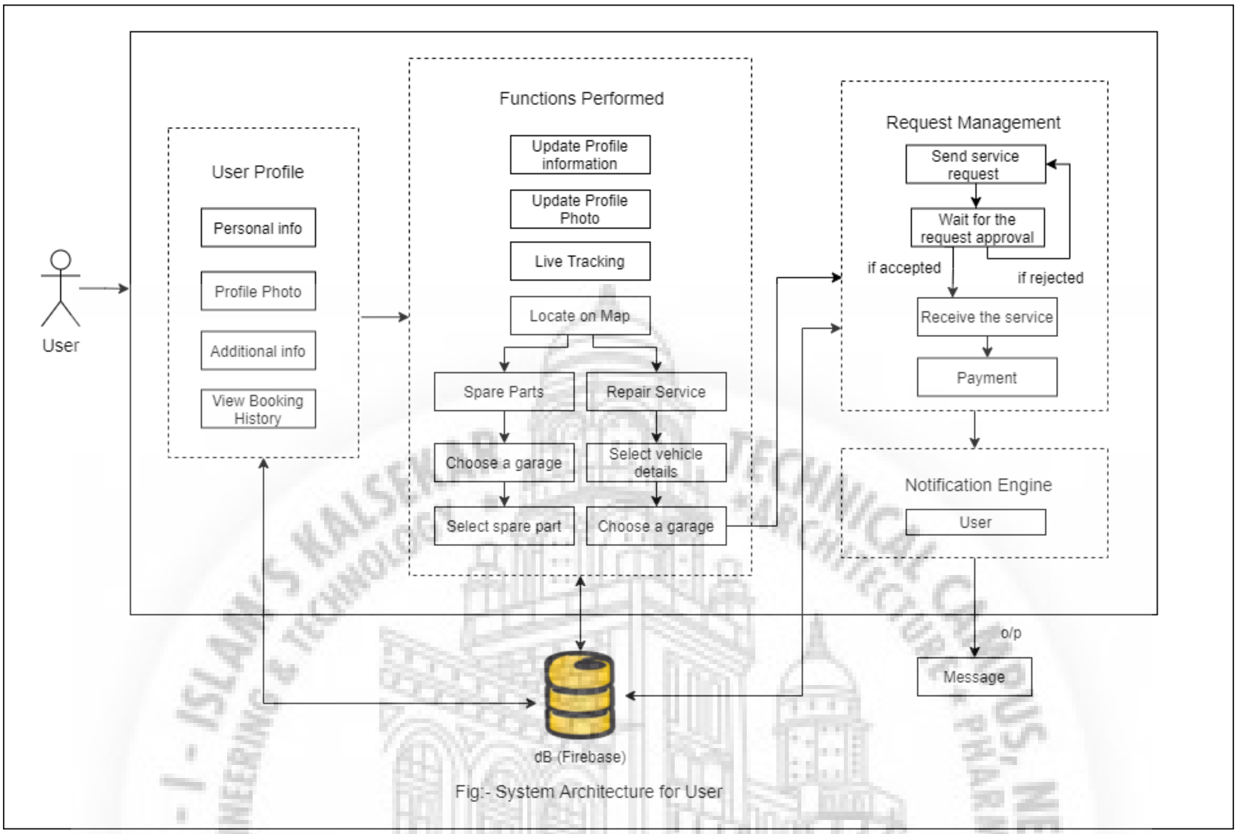


Figure 5.5: System Architecture of User

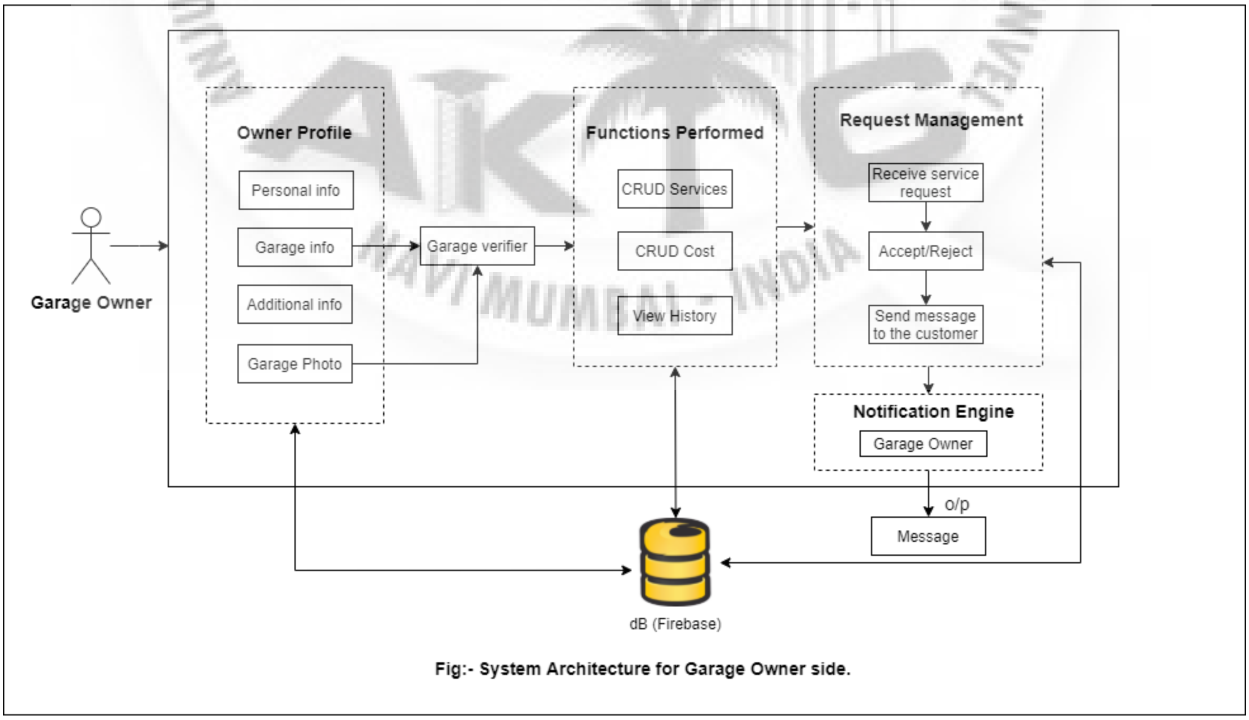


Figure 5.6: System Architecture of Garage Side User

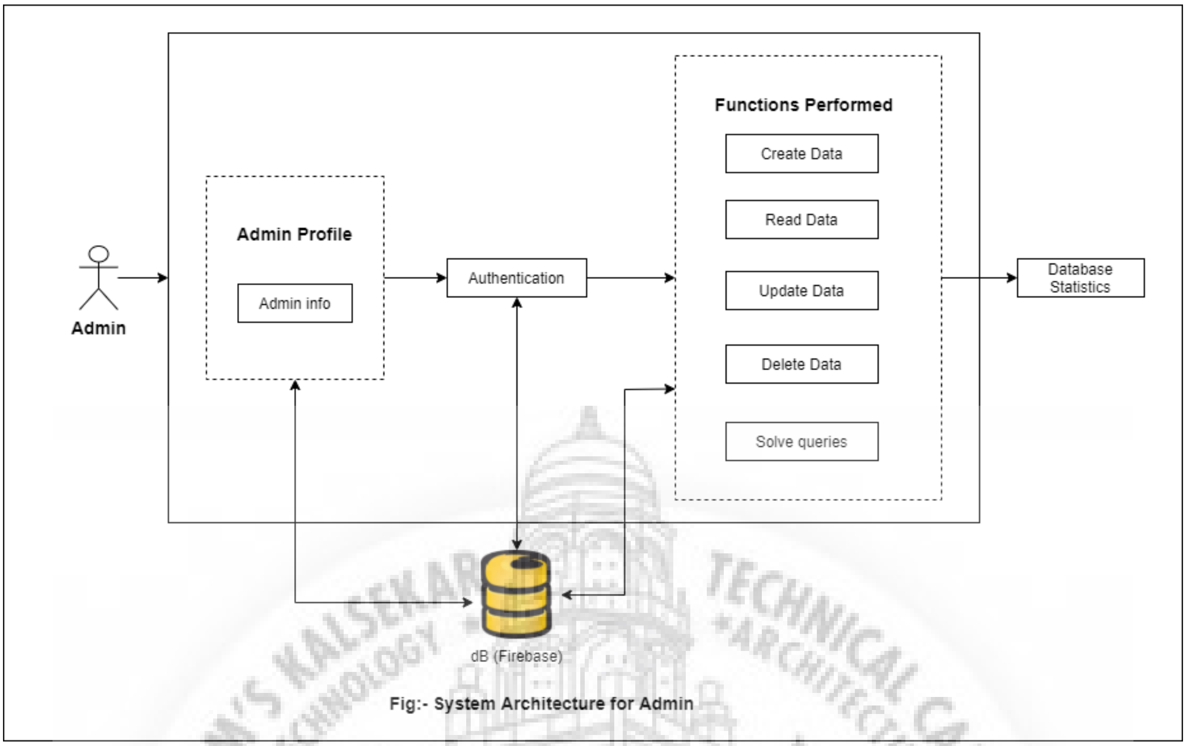


Figure 5.7: System Architecture of Admin

5.3 Sub-system Development

The system consist three user for each user we have user profile module for vehicle side user consist of personal information profile photo and booking history. While on garage side the profile module contains personal information, garage information and NOC.And on Admin side just login ID and Password.
The second common module is function module this tells us what function user can perform. The vehicle user can update or delete or create account, user can locate position on map, user can select services and garages. The garage side user can update garage information and cost, and allowed to see booking history. Admin can perform CRUD in order to solve queries.
Third Module is request management module in this vehicle user send request for services to the garage then garage can accept or reject the request. And the notification will be sent to both the users.

5.3.1 User (Vehicle Owner):

User Profile Module:

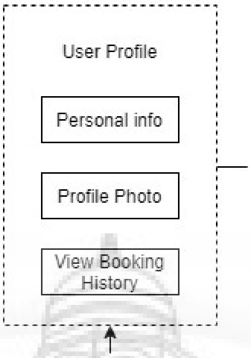


Figure 5.8: User Profile Module

The user will get registered themselves and then he/she will login create profile by filling personal information, profile photo and can view booking history.

User Function Module:

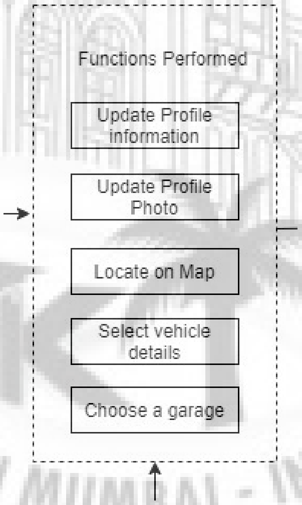


Figure 5.9: User Function Module

User has the full-fledged right to update profile information and photo. User will locate themselves on Map and then can select vehicle details and finally can choose a garage.

User Request Management Module:

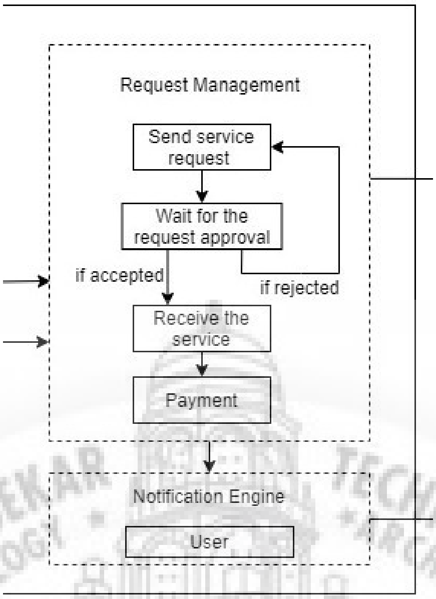


Figure 5.10: User Request Management Module

User will send service request to garage owner and has to wait for the request approval if the request is rejected then user has to search for another garage and has to send the service request again. And if request is accepted user will receive the service and has to pay for it. After receiving the request user will send the service feedback. User will send notification regarding service booking to garage which will be in a message form.

5.3.2 Garage Side User

Garage Profile Module:

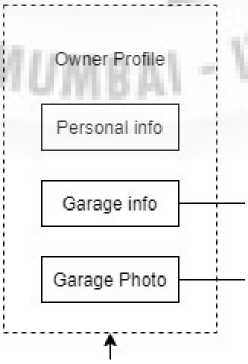


Figure 5.11: Garage Profile Module

The garage user will get registered themselves and then he/she will login create profile by filling personal information, garage information, profile photo and there is a garage verifier which will verify garage. The Garage owner will be verified on the basis of the NOC(No Objection Certificate)

Garage Function Module:

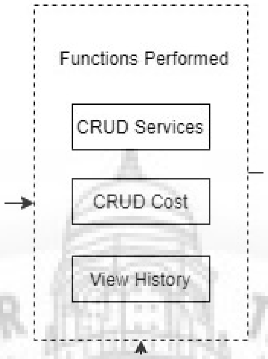


Figure 5.12: Garage Function Module

Garage user can perform CRUD operations on the services which they provide and the cost. Garage user can view the booking history.

Garage Request Management Module:

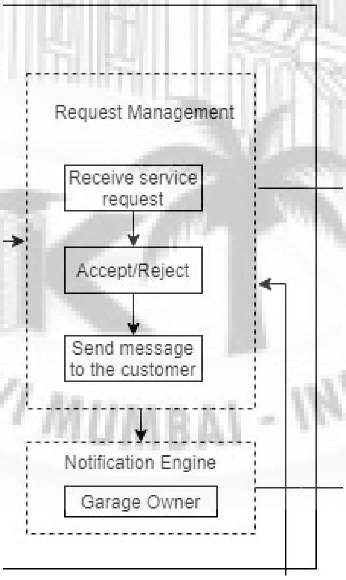


Figure 5.13: Garage Request Management Module

Garage user will receive service request from vehicle user and has the full-fledged right to accept or reject the request. If request is accepted message is send to customer and according to the problem garage owner will assign a suitable mechanic for it. Garage user will send notification regarding service booking to vehicle user which will be in a message form.

5.3.3 Admin:

Admin Profile Module:

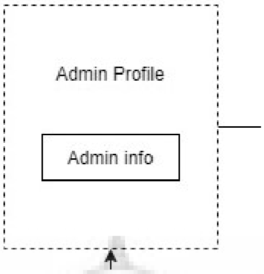


Figure 5.14: Admin Profile Module

In Admin Profile module there will be the information of admin. Authentication process will take place and once the user is authenticated then only he/she will get the full access of admin.

Admin Function Module:

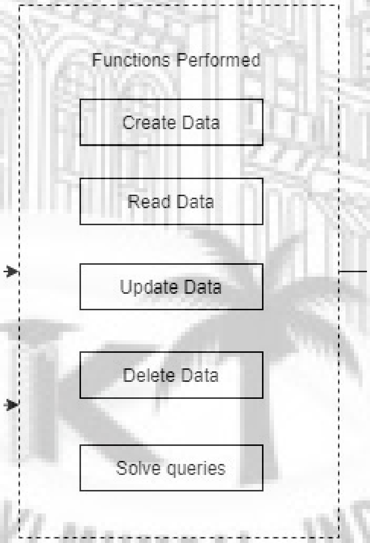


Figure 5.15: Admin Function Module

Admin can create data, read data, update data, delete data. And Admin also solve the queries of both the users. The output will be the Database Statistics which will show each and every details such as how many user has read the data, etc.

5.4 Systems Integration

System integration (SI) is an engineering process or phase concerned with joining different subsystems or components as one large system. It ensures that each integrated subsystem functions as required. Different Sub-Modules Integrated in one full System. SI is also used to add value to a system through new functionalities provided by connecting functions of different systems.

5.4.1 Class Diagram

A class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

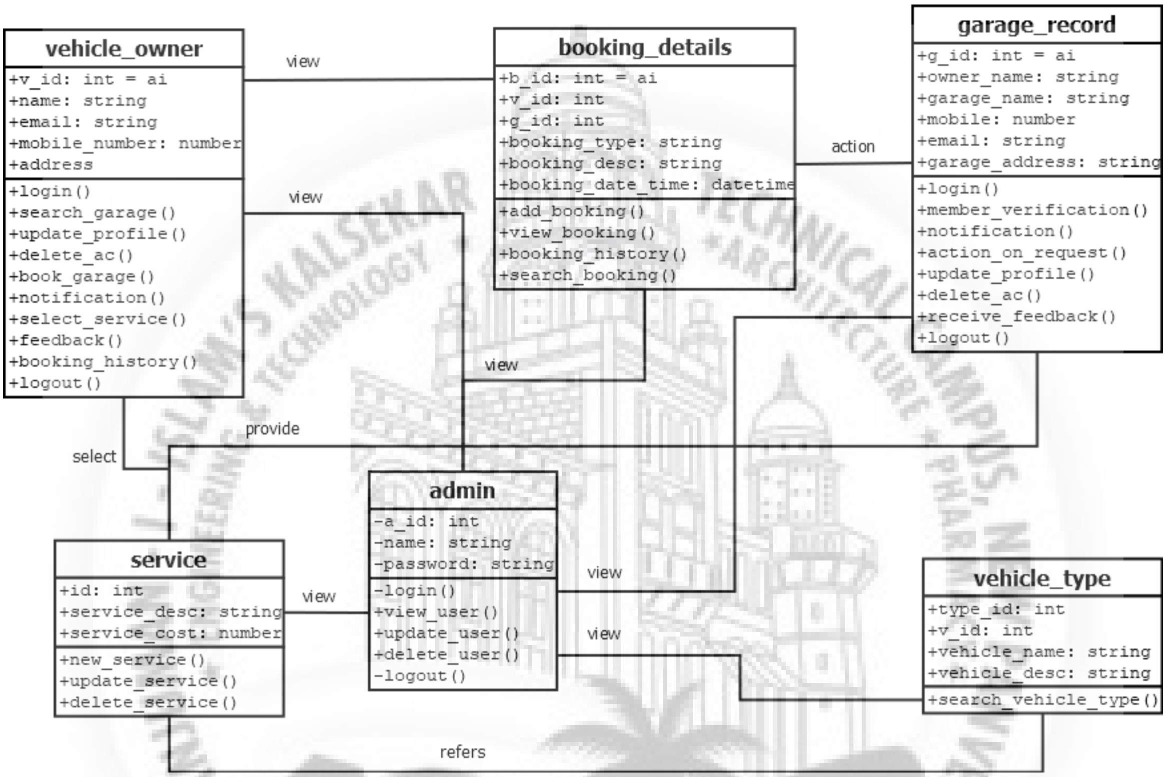


Figure 5.16: Class Diagram

5.4.2 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

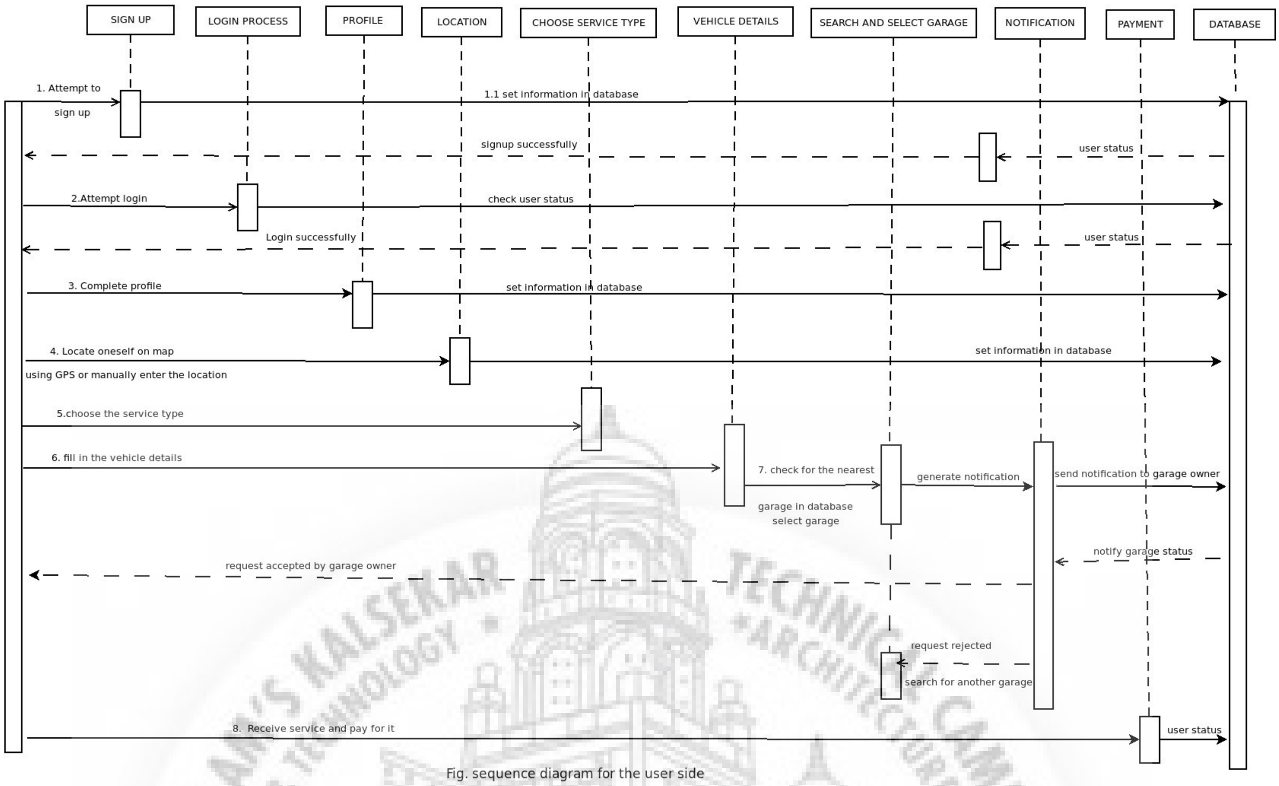

SIGN UP
LOGIN PROCESS
PROFILE
LOCATION
CHOOSE SERVICE TYPE
VEHICLE DETAILS
SEARCH AND SELECT GARAGE
NOTIFICATION
PAYMENT
DATABASE
1. Attempt to sign up
1.1 set information in database
signup successfully
user status
2.Attempt login
check user status
Login successfully
user status
3. Complete profile
set information in database
4. Locate oneself on map using GPS or manually enter the location
set information in database
5.choose the service type
6. fill in the vehicle details
7. check for the nearest garage in database select garage
generate notification
send notification to garage owner
notify garage status
request accepted by garage owner
request rejected
search for another garage
8. Receive service and pay for it
user status

Fig. sequence diagram for the user side

Figure 5.17: Sequence Diagram of User

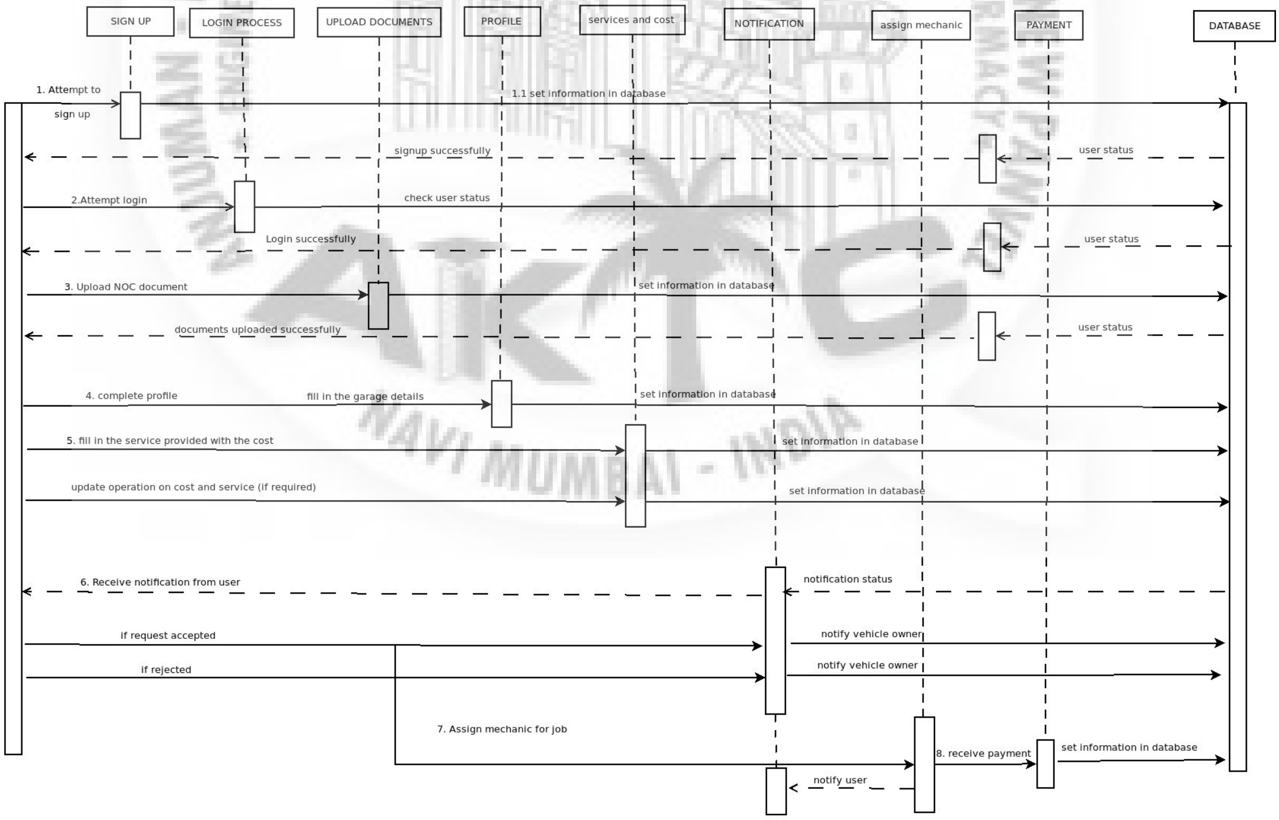


fig. sequence diagram for garage side

Figure 5.18: Sequence Diagram of Garage Side User

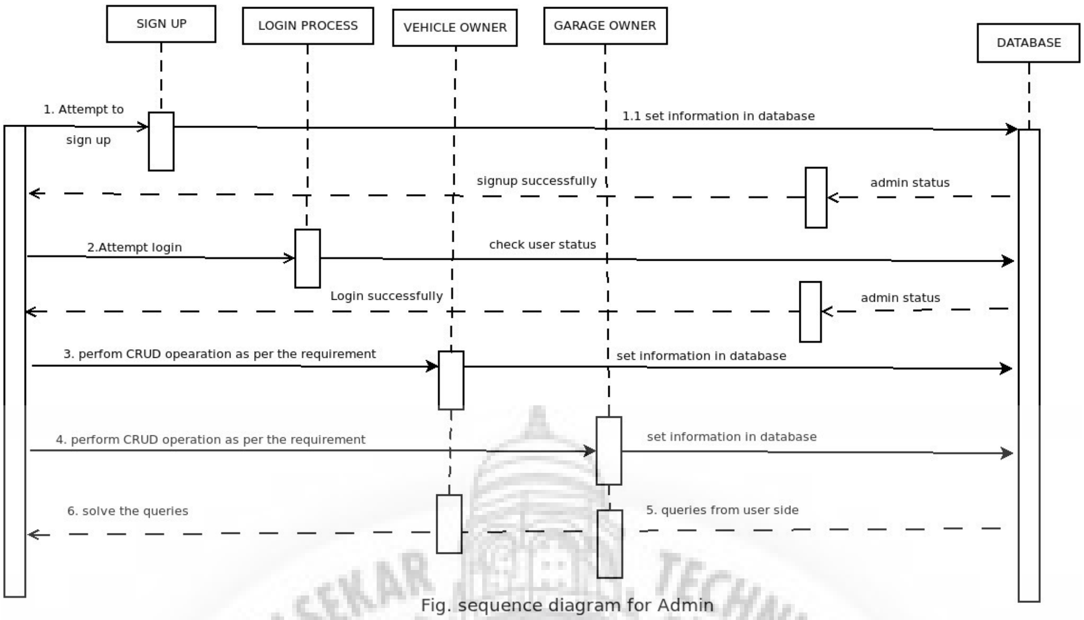

SIGN UP
LOGIN PROCESS
VEHICLE OWNER
GARAGE OWNER
DATABASE
1. Attempt to sign up
1.1 set information in database
signup successfully
admin status
2.Attempt login
check user status
Login successfully
admin status
3. perfom CRUD opearation as per the requirement
set information in database
4. perform CRUD operation as per the requirement
set information in database
6. solve the queries
5. queries from user side
Fig. sequence diagram for Admin

Figure 5.19: Sequence Diagram of Admin

5.4.3 Component Diagram

Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

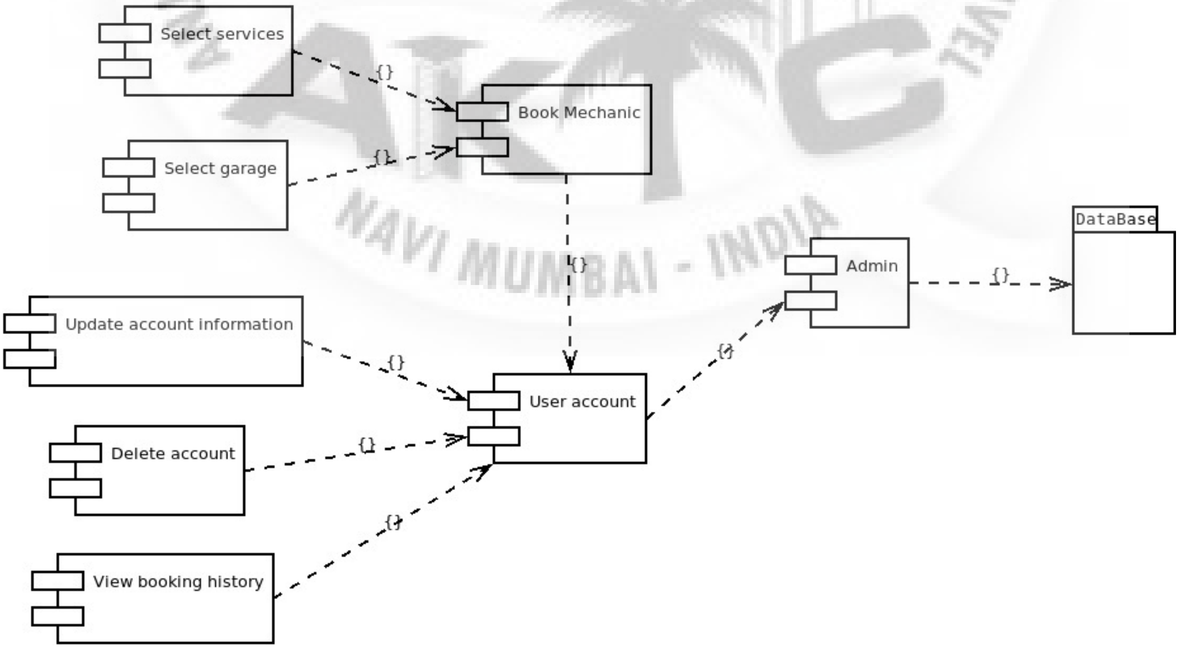


Figure 5.20: Component diagram of user side

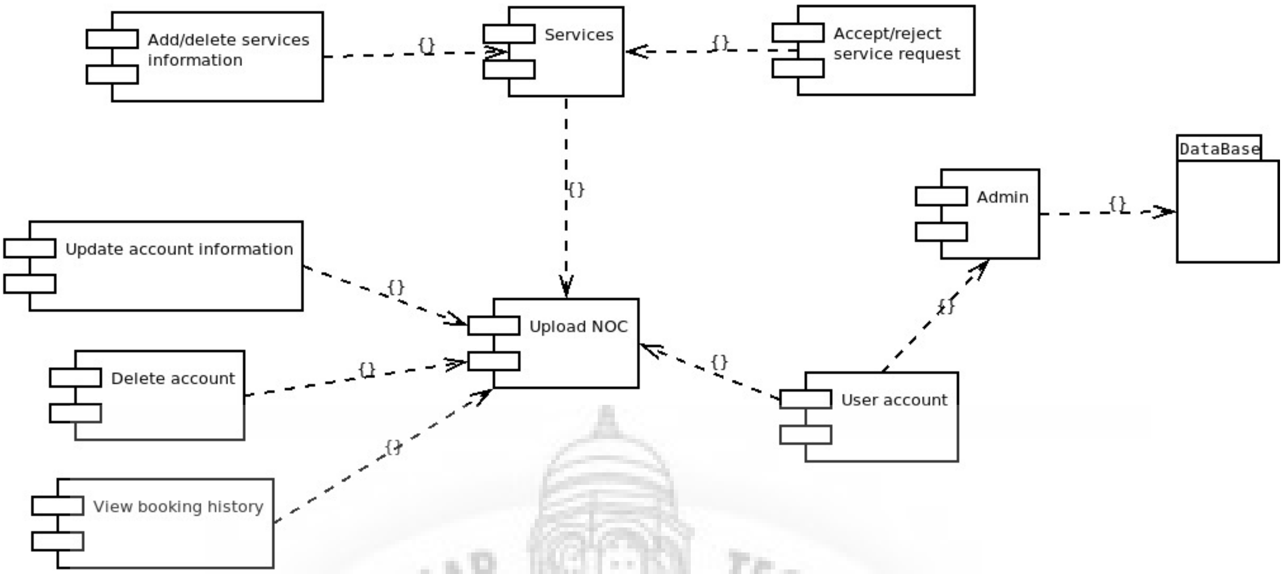


Figure 5.21: Component diagram of garage side side

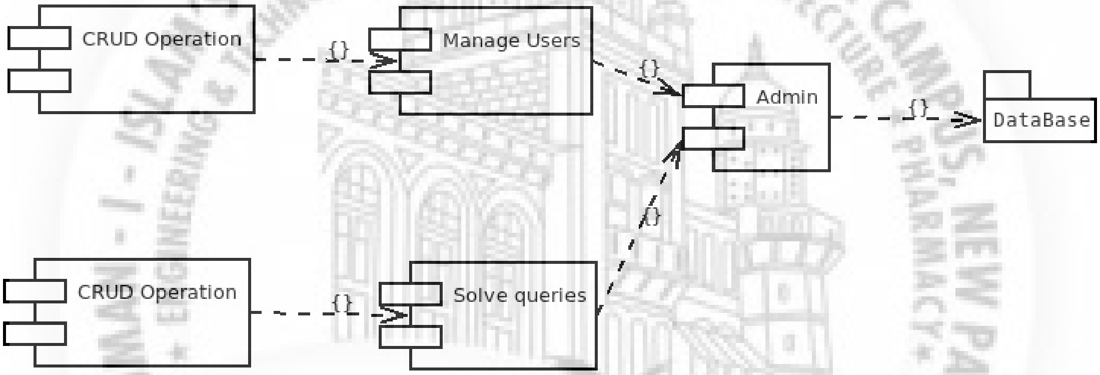


Figure 5.22: Component diagram of admin side

5.4.4 Deployment Diagram

A deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modeling the physical aspects of an object-oriented system. They are often be used to model the static deployment view of a system (topology of the hardware).

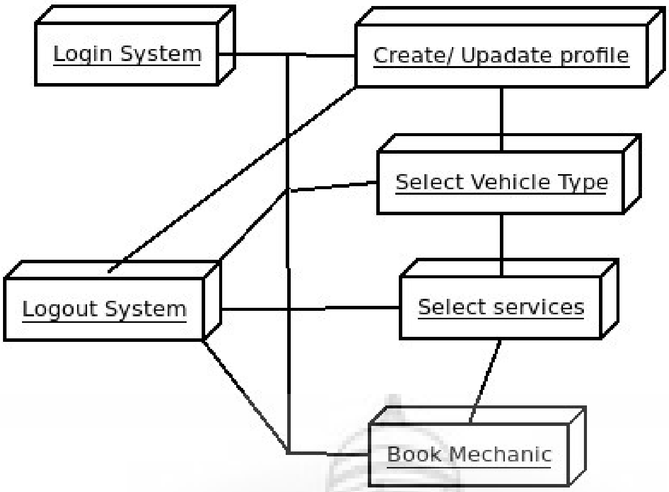


Figure 5.23: deployment diagram of user side

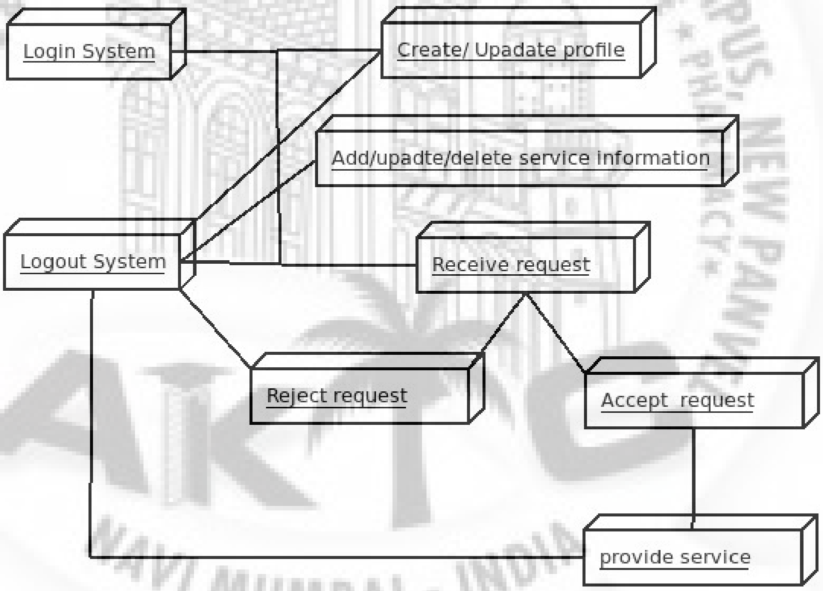


Figure 5.24: deployment diagram of garage side user

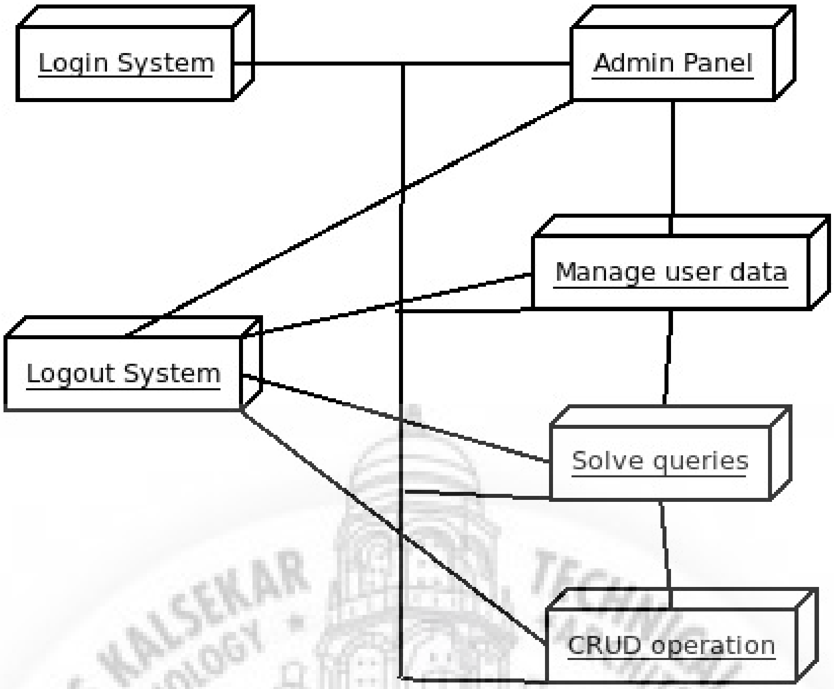


Figure 5.25: deployment diagram of admin side

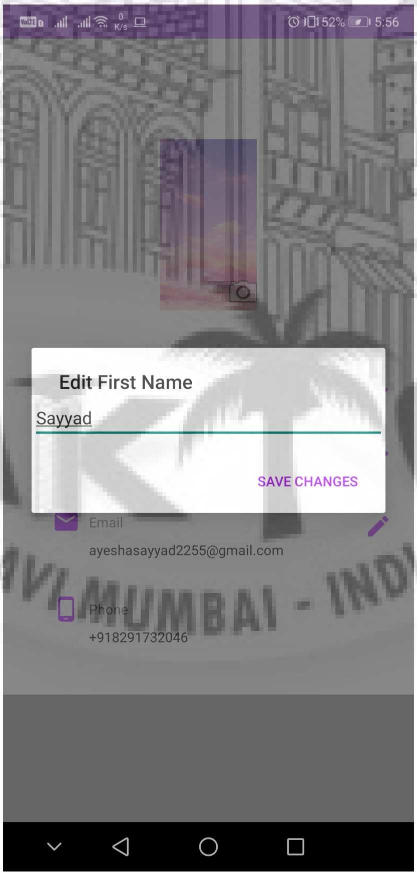
Chapter 6

Implementation

6.1 User side function module

6.2 Updating text

This module is build to update user data and save the updated data.



Edit First Name

Sayyad

ayeshasayyad2255@gmail.com

+918291732046

Figure 6.1: IMAGE CAPTION

```
public class EditDialog extends AppCompatActivity {

        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_editDialog);
            TextView text = findViewById(R.id.textView1);
            ImageView iText = findViewById(R.id.imageView1);
            AlertDialog dialog = new AlertDialog.Builder(this).create();
            EditText eText = new EditText(this);
            dialog.setTitle("Edit Text");
            dialog.setView(eText);

            dialog.setButton(DialogInterface.BUTTON_POSITIVE, "Save Changes",
            new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int
                                          which) {
                     text.setText(eText.getText());
                        update();
                     }
            });
                     iText.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                eText.setText(text.getText());
                dialog.show();
                }
         });
     }
}
```

For checking and unchecking the checkbox

This module is built for user to select multiple choices at the same time.

```
  @Override
public void onBindViewHolder(@NonNull final MyViewHolder holder,
                                         final int position) {
        holder.setItemClickListener(new ItemClickListener() {
        @Override
        public void onItemClick(View v, int pos) {
            CheckBox checkBox = (CheckBox) v;
            if (checkBox.isChecked()){
            checkedService.add(profiles.get(pos));
            } else if (!checkBox.isChecked()){
            checkedService.remove(profiles.get(pos));
            }
        }
    });
}
```

For dynamically creating a list

This module help us to create list which is dynamic in nature.

```
public class RecyclerActivity extends AppCompatActivity {
    RecyclerView recyclerView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_recycler);

        recyclerView = (RecyclerView)findViewById(R.id.myRecycle);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
        recyclerView.setItemAnimator(new DefaultItemAnimator());
        ArrayList<Profile> list = new ArrayList<Profile>();

        FirebaseDatabase.getInstance().getReference().child("Users").
        .addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            for (DataSnapshot dataSnapshot1 :                  dataSnapshot.
               getChildren()){
             Profile p = dataSnapshot1.getValue(Profile.class);
             list.add(p);
             }
            AdapterS sAdapter = new AdapterS(RecyclerActivity.this,list);
         recyclerView.setAdapter(sAdapter);
        }
        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {
         Toast.makeText(RecyclerActivity.this, "Oops Something went wrong",
         Toast.LENGTH_SHORT).show();
        }
     });
    }
}
```

For sending the message

This module enable user to send SMS to the other person.

```
public void notification(View view){
 String message = "Message";
 String number = phone;

 SmsManager mySmsManager = SmsManager.getDefault();
 mySmsManager.sendTextMessage(number,null,message,null, null);
}
```

For clicking the photo

In this section we have written code to click images and this code even allows user to crop image.

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
                                                        Intent data){
     super.onActivityResult(requestCode,resultCode,data);
```

```
            if(requestCode==CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE){
                CropImage.ActivityResult result = CropImage.getActivityResult(data)
                    ;
                if(resultCode == RESULT_OK){
                    mImageUri = result.getUri();
                    myImg.setImageURI(mImageUri);
                    final StorageReference imagename = Folder.child(”Image”+
                        mImageUri
                    .getLastPathSegment());
                    imagename.putFile(mImageUri).addOnSuccessListener
                    (new OnSuccessListener<UploadTask.TaskSnapshot>() {
                @Override
                public void onSuccess(UploadTask.TaskSnapshot
                                                taskSnapshot) {
                   Toast.makeText(ImageActivity.this, ”Uploaded”,
                    Toast.LENGTH_SHORT).show();
                    imagename.getDownloadUrl()
                    .addOnSuccessListener(new OnSuccessListener<Uri>() {
                    @Override
                    public void onSuccess(Uri uri) {
                    DatabaseReference ImageStore = FirebaseDatabase.getInstance().
                    getReference().child(”Users”).child(”Image”);
                    HashMap<String, String> hashMap = new HashMap<>();
                    hashMap.put(”image”, String.valueOf(uri));
                    ImageStore.setValue(hashMap);
                   }
               });
            }
        });
 }else if(resultCode == CropImage.CROP_IMAGE_ACTIVITY_RESULT_ERROR_CODE){
        Exception e = result.getError();
        Log.d(”error”,e.toString());
            }
        }
}
```

Live location tracking of user

Suppose a user wants to share its live location or wants other person to track its live location hence here is the module which track the live location of use.

```
package com.example.autorescuer_drivemecrazy;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.FragmentActivity;

import android.Manifest;
import android.app.Activity;
import android.app.ActivityManager;
import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.PowerManager;
import android.provider.Settings;
import android.telephony.SmsManager;
import android.util.Log;
import android.view.View;
import android.widget.SearchView;
import android.widget.Toast;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;

import java.io.IOException;
import java.util.List;
import java.util.Locale;
import android.content.Intent;
import android.os.PowerManager;
import android.net.Uri;
import android.os.Build;
import android.provider.Settings;

public class LiveTrackingF extends FragmentActivity implements
    OnMapReadyCallback{

    private GoogleMap mMap;
    LocationManager locationManager;
    private static final int REQUEST_LOCATION_PERMISSION = 1;
    Marker marker;
    String notfound, pin;
    LocationListener locationListener;
    public static final String  userId = FirebaseAuth.getInstance().
        getCurrentUser().getUid();
    public static final String TAG = ”TAG”;
    String UserName, phoneNo, phoneNo2, stop = null;
    final String p = FirebaseAuth.getInstance().getCurrentUser().getPhoneNumber
        ();
    private final int NOTIFICATION_ID = 001;
    NotificationManager manager;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_live_tracking_f);

        // Obtain the SupportMapFragment and get notified when the map is ready
            to be used.
        SupportMapFragment mapFragment = (SupportMapFragment)
            getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);


        notfound =  getString(R.string.notfound);
        stop = getIntent().getStringExtra(”stop”);

        FirebaseDatabase.getInstance().getReference().child(”Users”).child(”
            VehicleUsers”).child(p)
                .addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot snapshot) {
                        if (snapshot.hasChild(”Tracking Stopped”)){
                            FirebaseDatabase.getInstance().getReference(”Users”)
                                .child(”VehicleUsers”).child(p).child(”Tracking
                                Stopped”).removeValue();
                        }
                    }

                    @Override
                    public void onCancelled(@NonNull DatabaseError error) {

                    }
                });


        final DocumentReference docRef = FirebaseFirestore.getInstance().
            collection(”users”).document(userId);
        docRef.get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot
            >() {
             @Override
             public void onSuccess(DocumentSnapshot documentSnapshot) {
                 if (documentSnapshot.exists()) {
                     UserName = documentSnapshot.getString(”firstName”)+ ” ”+
                         documentSnapshot.getString(”lastName”);
                     FirebaseDatabase.getInstance().getReference(”Users”)
                             .child(”VehicleUsers”)
                             .child(p).child(”Vehicle Owner Name”).setValue(
                                 UserName);
                 } else {
                    Log.d(TAG, notfound);
                }
            }
        });
if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
            NotificationChannel channel =
                    new NotificationChannel("MyNotifications","MyNotifications",
                            NotificationManager.IMPORTANCE_DEFAULT);

            manager = getSystemService(NotificationManager.class);
            manager.createNotificationChannel(channel);
        }

        Intent intent = new Intent(this, LiveTrackingF.class);
        intent.putExtra("stop","true");
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 1, intent,
             PendingIntent.FLAG_UPDATE_CURRENT);


        NotificationCompat.Builder builder = new NotificationCompat.Builder(
            LiveTrackingF.this,
                "MyNotifications");
        builder.setSmallIcon(R.drawable.ic_edit_black_24dp);
        builder.setContentTitle("Your Live Tracking has been started\uD83D\uDCCD
            ");
        builder.setContentText("Tap to stop");
        builder.setPriority(NotificationCompat.PRIORITY_DEFAULT);
        builder.setContentIntent(pendingIntent);
        builder.setOngoing(true);
        builder.setAutoCancel(true);
        NotificationManagerCompat notificationManagerCompat =
                NotificationManagerCompat.from(LiveTrackingF.this);
        notificationManagerCompat.notify(NOTIFICATION_ID, builder.build());




        if (stop != null && stop.equals("true")){

            new AlertDialog.Builder(this)
                    .setTitle("End live location?")
                    .setMessage("Are you sure you want to end it?")
                    .setNegativeButton(android.R.string.no, new DialogInterface.
                        OnClickListener() {
                        public void onClick(DialogInterface arg0, int arg1) {
                            marker.remove();
                            startActivity(new Intent(LiveTrackingF.this,
                                LiveTrackingF.class));
                            finish();
                        }
                    })
                    .setPositiveButton(android.R.string.yes, new DialogInterface
                        .OnClickListener() {

                        public void onClick(DialogInterface arg0, int arg1) {
                            FirebaseDatabase.getInstance().getReference("Users")
                                .child("VehicleUsers").child(p).child("Tracking
                                Stopped").setValue("1");
                            manager.cancelAll();
                            locationManager.removeUpdates(locationListener);
                            startActivity(new Intent(LiveTrackingF.this,
                                DashboardActivity.class));
                            finish();
                        }
                    }).create().show();

        }
        FirebaseDatabase.getInstance().getReference().child("Users").child("
           VehicleUsers").child(p)
                .addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot snapshot) {
                        phoneNo = snapshot.child("LiveTracking Phone1").child("1
                            ").getValue(String.class);
                        pin = snapshot.child("Pin").getValue(String.class);

                        if (snapshot.hasChild("LiveTracking Phone2")){
                            phoneNo2 = snapshot.child("LiveTracking Phone2").
                                child("2").getValue(String.class);
                        }
                    }

                    @Override
                    public void onCancelled(@NonNull DatabaseError error) {

                    }
                });


        mapFragment.getMapAsync(this);
        locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
        if (ActivityCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_FINE_LOCATION)
                != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new String[]
                            {Manifest.permission.ACCESS_FINE_LOCATION},
                    REQUEST_LOCATION_PERMISSION);
        }
        else{

            locationListener = new LocationListener() {
                @Override
                public void onLocationChanged(Location location) {
                    double latitude = location.getLatitude();
                    double longitude = location.getLongitude();

                    //get the location name from latitude and longitude
                    LocationHelper helper = new LocationHelper(
                            location.getLongitude(),
                            location.getLatitude(),
                            getAddress(location.getLatitude(),location.
                                getLongitude()).toString()
                    );
                    String p = FirebaseAuth.getInstance().getCurrentUser().
                        getPhoneNumber();
                    FirebaseDatabase.getInstance().getReference("Users")
                            .child("VehicleUsers")
                            .child(p).child("Tracking Location").setValue(helper
                                );

                    Geocoder geocoder = new Geocoder(getApplicationContext());
                    LatLng latLng = new LatLng(latitude, longitude);
                    if (marker != null){
                        marker.remove();
                        marker = mMap.addMarker(new MarkerOptions().position(latLng).icon(BitmapDescriptorFactory.fromResource(R.drawable.artgallery)).title(UserName));
                        mMap.setMaxZoomPreference(20);
                        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng, 17));
                    }
                    else{
                        marker = mMap.addMarker(new MarkerOptions().position(latLng).icon(BitmapDescriptorFactory.fromResource(R.drawable.artgallery)).title(UserName));
                        mMap.setMaxZoomPreference(20);
                        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng, 17));
                    }
                }

                @Override
                public void onStatusChanged(String provider, int status, Bundle extras) {

                }

                @Override
                public void onProviderEnabled(String provider) {

                }

                @Override
                public void onProviderDisabled(String provider) {

                }
            };
            locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, locationListener);
            locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, locationListener);
        }
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
    }


    private String getAddress(double Latitude, double Longtitude){
        String address="";
        Geocoder geocoder = new Geocoder(LiveTrackingF.this , Locale.getDefault());
        try {
            List<Address> addresses = geocoder.getFromLocation(Latitude, Longtitude, 1);
            if ((address!=null)){
                Address returnAddress = addresses.get(0);
                StringBuilder stringBuilderReturnAddress = new StringBuilder("");

                for (int i=0;i<=returnAddress.getMaxAddressLineIndex();i++){
                    stringBuilderReturnAddress.append(returnAddress.getAddressLine(i)).append("\n");
                }
                address = stringBuilderReturnAddress.toString();
            }else{
                Toast.makeText(LiveTrackingF.this, "Address not found", Toast.LENGTH_SHORT).show();
            }
        } catch (Exception e) {
            Toast.makeText(LiveTrackingF.this, e.getMessage().toString(), Toast.LENGTH_SHORT).show();
        }
        return address;
    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();
        Intent i = new Intent(LiveTrackingF.this, DashboardActivity.class);
        i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(i);
        finish();
    }

}
```

6.3 Location module

In this module we have build Map.The map includes a marker, also called a pin, to indicate a specific location.Location here is vehicle users current location and garage nearby location within range 10km.

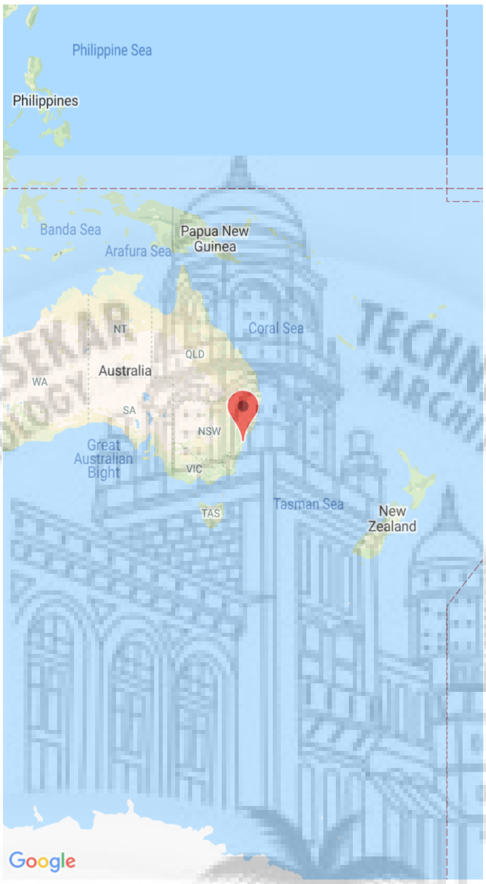


Figure 6.2: Map with marker

```
public class MapsActivity extends FragmentActivity implements
    OnMapReadyCallback {
        private GoogleMap mMap;
        double latitude,longitude;

        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_maps);
             LatLng latLng = new LatLng(latitude,longitude);
    mMap.addMarker(new MarkerOptions().position(latLng).title("Marker")
 .icon(bitmapDescriptorFromVector(getApplicationContext(),
R.drawable.artgallery)));

mMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng,10));
 }
}
```

6.4 Database module

In this module of project all the data related to garage, garage user and vehicle user is managed by Firebase. In this module we are trying to store and manage updated data of user.The reason to choose this database is that:

1. Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud. It keeps your data in sync across client apps through realtime listeners.
2. The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client, with cross-platform support for iOS, Android, Web and more.

Updating text module

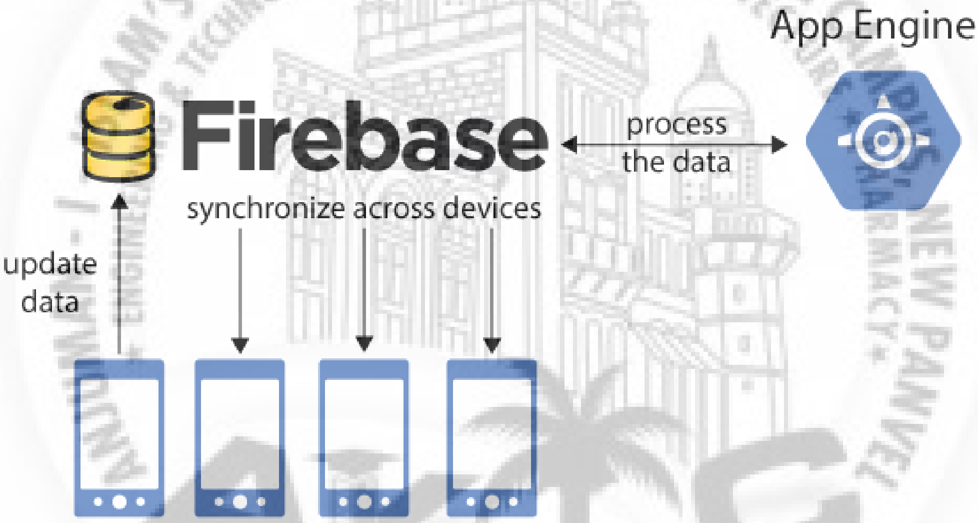


Figure 6.3: updating data module

```
public static final String userId = FirebaseAuth.getInstance().
                        getCurrentUser().getUid();
 private void update()
 {
        String text = text.getText().toString().trim();
        FirebaseFirestore db = FirebaseFirestore.getInstance();
        DocumentReference ref = db.collection("users").document(userId);
        ref.update("text",text)
        .addOnCompleteListener(new OnCompleteListener<Void>()
        {
            @Override
             public void onComplete(@NonNull Task<Void> task)
             {
                if (task.isSuccessful())
                {
                    Toast.makeText(UpdateActivity.this, "Profile Updated",
                    Toast.LENGTH_SHORT).show();
            }
            else
            {
                    Toast.makeText(UpdateActivity.this, ”Profile Not Updated”,
                     Toast.LENGTH_SHORT).show();
                    }
            }
        });
}
```

6.5 Admin side function module

Here we have build a website using html,css,javascript files.

index page:index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html" charset="utf-8">
  <title>Login</title>
  <link rel="stylesheet" href="css/lg.css" type="text/css">
  <link rel="stylesheet" href="https://font.googleapis.com/css?family=Play">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="css/style.css">
</head>
    <body>
<div class="card">
    <nav class="navbar navbar-inverse navbar-fixed-top">
      <ul class="nav navbar-nav navbar-right">
        <li><a href="index.html"><span class="glyphicon glyphicon-log-in"></span> Login</a></li>
      </ul>
    </nav>

  <div class="signin">
    <form method="POST" action="#">
        <h2 style="color: rgb(102, 102,255)">Log In</h2>
      <input type="text" id="email" name="email" placeholder="Email" required>
      <input type="password" id="password" name="password" placeholder="Password" required><br><br>
      <input onclick="userLogin()" type="button" id="login" name="submit" value="Log In">
            <h5>OR</h5>
            <input onclick="gmaillogin()" type="button" id="glogin" style="background: rgb(219,4,4); " name="submit" value="Login with Google">
           <br><br>
          Don't Have an Account?<a href="signup.html">&nbsp;Sign Up</a>
          <br><div id="status">

          </div>
    </form>
    </div>

    <script src="https://www.gstatic.com/firebasejs/7.10.0/firebase-app.js"></script>
    <script src="https://www.gstatic.com/firebasejs/7.10.0/firebase-auth.js"></script>
    <script src="https://www.gstatic.com/firebasejs/7.10.0/firebase-database.js"></script>
    <script src="https://www.gstatic.com/firebasejs/7.10.0/firebase-storage.js"></script>
  <script src="https://www.gstatic.com/firebasejs/7.10.0/firebase-analytics.js"></script>
    <script src="js/index.js"></script>
    <script type='text/javascript' src="js/jquery.js"></script>
    <script src="js/login.js" type="text/javascript"></script>
    <script src="js/gmailLogin.js" type="text/javascript"></script>


    </body>
</html>
```

signup page: signup.html

```
<!DOCTYPE html>
<html lang=”en”>
<head>
        <meta http-equiv=”Content-Type” content=”text/html” charset=”utf-8”>
        <title>Sign Up</title>
        <meta name=”viewport” content=”width=device-width, initial-scale=1”>
        <link rel=”stylesheet” href=”css/signup.css” type=”text/css”>
        </head>
    </head>
<body>
  <div class=”container”>
           <form method=”post” action=”” enctype=”multipart/form-data”>
               <h2 style=”color: rgb(102, 102,255)”>Sign Up</h2>
               <input type=”email” id=”email” name=”email” placeholder=”Email
                   Address” required><br><br>
               <input type=”password” class=”form-control” name=”password” id=”
                   password” placeholder=”Password” required><br><br>
               <input type=”password” class=”form-control” name=”cpassword” id=
                   ”cpassword” placeholder=”Confirm Password” required><br><br>
               <input type=”button” onclick=”createUser();” name=”btn_register”
                    class=”btn btn-primary” id=”submit” value=”Submit”><br>
               Already Have an Account?<a href=”index.html” style=”text-
                   decoration: none; font-family: 'Play', sans-serif; color:
                   rgb(255, 102, 102);”>&nbsp;Log In</a>
               </form>
           </div>

    <!-- The core Firebase JS SDK is always required and must be listed first
       -->
    <script src=”https://www.gstatic.com/firebasejs/7.10.0/firebase-app.js”></
        script>
    <script src=”https://www.gstatic.com/firebasejs/7.10.0/firebase-auth.js”></
        script>
    <script src=”https://www.gstatic.com/firebasejs/7.10.0/firebase-database.js”
       ></script>
    <script src=”https://www.gstatic.com/firebasejs/7.10.0/firebase-storage.js”
       ></script>

    <!-- TODO: Add SDKs for Firebase products that you want to use
        https://firebase.google.com/docs/web/setup#available-libraries -->
    <script src=”https://www.gstatic.com/firebasejs/7.10.0/firebase-analytics.js
       ”></script>

    <script src=”js/index.js”></script>
    <script src=”js/jquery.js”></script>
    <script src=”js/signup.js”></script>
</body>
</html>
```

javascript file:gmailLogin.js

```
function gmaillogin(){
    var provider = new firebase.auth.GoogleAuthProvider();
    firebase.auth().signInWithPopup(provider).then(function(){
        window.location=”https://console.firebase.google.com/project/autorescuer
            -drive-me-crazy/authentication/users”;

    }).catch(function(error){
        var errorMessage = error.message;
        alert(errorMessage);
    });
}
```

javascript file:login.js

```
function userLogin(){
    var email = document.getElementById(”email”).value;
    var password = document.getElementById(”password”).value;

    firebase.auth().signInWithEmailAndPassword(email,password).then(function(){
        window.location=”https://console.firebase.google.com/project/autorescuer
            -drive-me-crazy/authentication/users”;

    }).catch(function(error){
        var errorMessage = error.message;
        alert(errorMessage);
    });

}
```

Chapter 7

System Testing

System testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

7.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Login into system	Android mobile kitkat or higher version	After entering Mobile number and OTP User get registerd	User account should be created
T02	Mapping current location	Enable location	after enabling location current location will be fetched	Located current position shown
T03	Enter vehicle details	select details from drop-down menu	data will be stored	nearby garage will be shown on map
T04	Garage selection	within range 10km	garage services will be shown, user need to select services.	Service request will be send to selected garage
T05	service request status	If accepted/rejected by garage owner	message will be sent to user	SMS will be sent to user
T06	Live Tracking	Enable Location and shared pin with person	If pin matched correctly live tracking will be enabled	live location will be send
T07	Spare parts	book spare parts	nearby garages with spare parts will be shown	purchase spare parts

T08	Services and spare part	add or edit services, spare parts and cost	data added or updated accordingly	data updated sucessfully
T09	Upload picture	Picture with good quality	Camera or gallery will be open to upload picture	picture uploaded successfully
T010	Booking History	previous booking details	booking history will be fetched	booking history shown successfully
T011	Contact Us	call or message on whatsApp or gmail	Feedback given successfully	Successful feedback
T012	Login page for admin	Username and Password required	Login into firebase console	Successful login

7.2 Sample of a Test Case

Title: Login into system – Authenticate successfully using OTP sent on registered mobile
Description: A registered user should be able to successfully login at system.

Precondition: the user must already be registered with an email address and password.

Assumption: a supported browser is being used.

Test Steps:

1. Download APK or Application
2. In the 'mobile' field, enter the mobile of the registered user.
3. Click the 'Next' button.
4. Enter the OTP
5. Click on 'Submit'

Expected Result: User account should be created
Actual Result: Here user account is created after login successfully upload the image of result

Title: Mapping Current Location – Enable Location to locate on Map
Description: A registered user can locate themselves on Map after enabling the location.

Precondition: Location already enabled.

Assumption: good internet connection to fetch location .

Test Steps:

1. Click on Map icon
2. Fetch the location
3. Click the 'Back' button.

Expected Result: correct location should be shown on map
Actual Result: Location fetched and shown successfully
upload the image of result

Title: Enter vehicle details
Description: Here user should be able to choose the brand and model of their vehicle.

*Precondition:*Application should have all possible vehicle details information.

Assumption: Garage provide service to the user's vehicle.

Test Steps:

1. Click on 'Repair services'
2. Select the vehicle details from menu provided.

Expected Result: Vehicle details should be stored in database and nearby garages will be shown on map
Actual Result: garages within range of 10 km has shown successfully.

Title: Garage selection
Description: User should be able to choose garage of their own choice.

Precondition: Garage should have account on Application.

Assumption: registered garage is available within the range 10km.

Test Steps:

1. As soon as user click on icon of garage shown on map garage services page will pop-up
2. Click on services you want.
3. Click the ‘Continue’ button.
4. Check the service cost.

Expected Result:services with cost should be selected and bill should generate correctly
Actual Result: service name and bill generated successfully.

Title: service request status
Description: Selected garage should be willing to do the job.
Precondition: Mechanic should be able to do the requested service
Assumption: Garage have mechanic available to do the job

Test Steps:

1. Click on ’Send Notification’
2. In case wants to cancel request then, click on ’Cancel’
3. wait for request approval from selected garage
4. Service request status will be sent to both the users.

Expected Result: Selected garage should get the message related to user’s request.
Actual Result: Notification sent successfully.

Title: Live tracking
Description: Live tracking is successfully working with correct pin shared with person.
Precondition: Enable Location and share pin

Assumption: location is enabled and shared pin is correct.

Test Steps:

1. Enable location
2. Enable live tracking option
3. Enable live location sharing option
4. Enter the mobile number whom you wish to share live location
5. Create a 5 digit pin
6. Click on 'Next'
7. Pin is shared with mobile number entered
8. the person who received the 5 digit pin navigate to Home page
9. Click on 'For tracking purpose'
10. Enter Mobile number to be tracked
11. Click on 'Continue'
12. Enter 5 digit pin
13. Click on 'Continue'
14. Automatically navigate to Map showing live location.

Expected Result: Live tracking should enable
Actual Result: Live tracking enabled successfully.

Title: spare parts
Description: Spare parts of vehicle bought successfully from nearby garage.
Precondition: Garage should have requested spare part available
Assumption: Spare parts are available in garages.

Test Steps:

1. Navigate to Main page and click on ’Spare parts’
2. Map with garage will be shown
3. Select the garage.
4. Select the product.
5. Click on ’Continue’.
6. View product details.
7. Click on ’Add to cart’ after selecting the product with quantity.
8. Take look at bill generated.
9. Click on ’Sent order request’.

Expected Result: order request should be sent to the selected garage
Actual Result: Order request sent successfully.

Title: services and spare parts
Description: Garage owner can edit add or delete the service and spare parts.

Precondition: Garage should provide the service and can edit the information related to the services and spare parts.

Assumption: Garage side user has updated the information according to need.

Test Steps:

1. Open Application
2. Click on ’Garage services’ from navigation drawer.
3. Add service with cost.
4. Click on ’Save details’.
5. Click on Edit symbol to edit data.
6. Click on ’Update service name’ or ’Update service cost’ .

7. Enter the data.
8. Click on 'Save changes'.

Expected Result: Data should be updated.
Actual Result: Service cost and name added and updated successfully.

Title: Upload image
Description: Profile picture for both the users should be uploaded and saved in database. NOC picture of garage should be uploaded in order to register the garage.

*Precondition:*NOC should be valid. and for profile picture should be clear.

Assumption: NOC is valid. And pictures are clear

Test Steps:

1. Navigate to navigation drawer to upload picture.
2. Click on 'Profile'.
3. Click on image symbol.
4. select the image or click the image to upload.
5. To upload NOC Navigate to Home page.
6. Click on 'I am a garage owner'.
7. Click on 'Upload NOC'
8. select the image or click the image to upload.

Expected Result: Picture should be saved in database.
Actual Result: Picture uploaded successfully.

Title: booking history
Description: Previous booking related to services of garage has provided to the users with full information is stored in database and it is fetched successfully.

Precondition: user has atleast one service booking.
Assumption: User has history to be shown.

Test Steps:

1. Navigate to navigation drawer.
2. Click on ’Booking history’.

Expected Result: All previous services or services request data should be fetched.
Actual Result: Previous booking data shown successfully.

Title: contact us
Description: A user must be able to send feedback or response to developers. *Precondition:* A registered email or Valid phone number or WhatsApp account needed. *Assumption:* User has valid email and mobile number .

Test Steps:

1. Navigate to navigation drawer.
2. Click on ’Contact us’
3. Select the mode of communication.

Expected Result: All the communication method should work fine and user message sent successfully.
Actual Result:user message sent successfully

Title: Admin Login Page
Description: Authentication successfully using valid email address and credential.
Precondition: Valid email and password is needed.
Assumption: supported browser is being used with good internet connection

Test Steps:

1. Open Browser
2. Navigate to the login page using link .
3. Click the ‘Login’ button after entering valid email and password.

Expected Result: User should be able to login.
Actual Result: Authentication successful

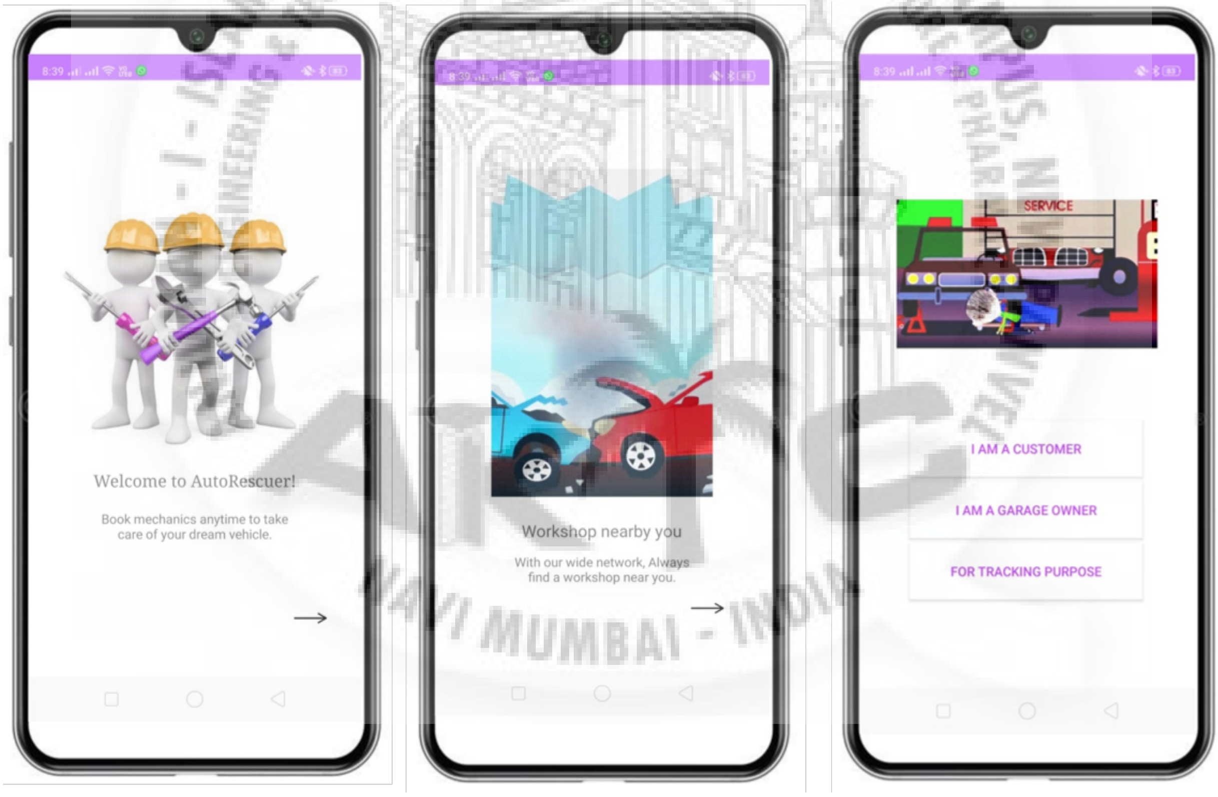
7.2.1 Software Quality Attributes

Availability-1 : The system shall be available to users all the time.
Availability-2 : The system shall always have something to function and always pop up error messages in case of component failure.
Efficiency-1 : The system shall provide the right tools to support all its features.
Flexibility-1 : The system should be flexible enough to fetch user’s correct location and nearby location of garages(range 10km)

Chapter 8

Screenshots of Project

8.1 Vehicle owner side:



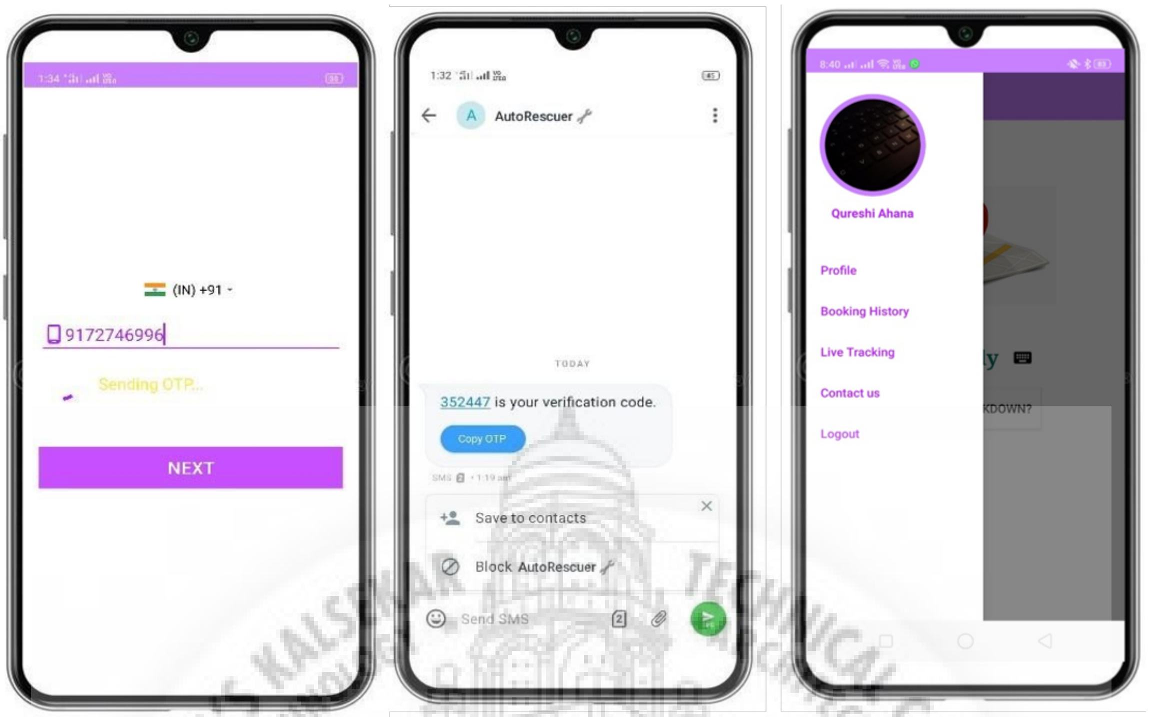
Welcome to AutoRescuer!

Book mechanics anytime to take care of your dream vehicle.

Workshop nearby you

With our wide network, Always find a workshop near you.

Figure 8.1: Application information and user login

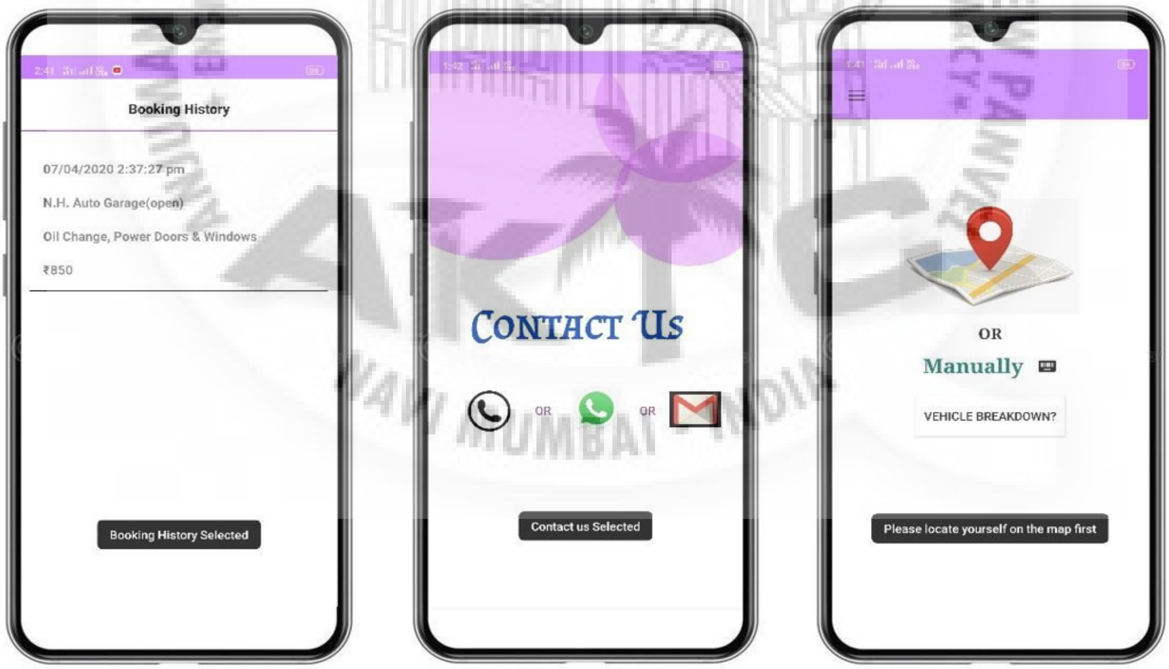


9172746996

Sending OTP...

352447 is your verification code.

Figure 8.2: registration of a user



07/04/2020 2:37:27 pm

N.H. Auto Garage(open)

Oil Change, Power Doors & Windows

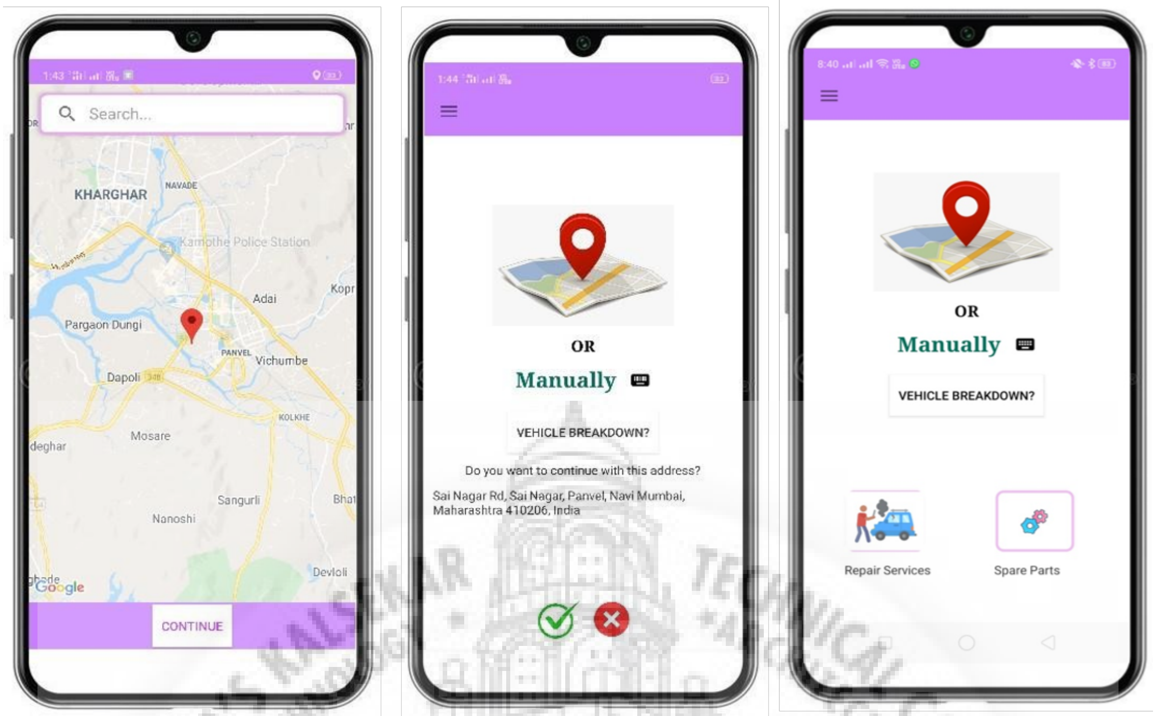
₹850

Booking History Selected

Contact us Selected

Please locate yourself on the map first

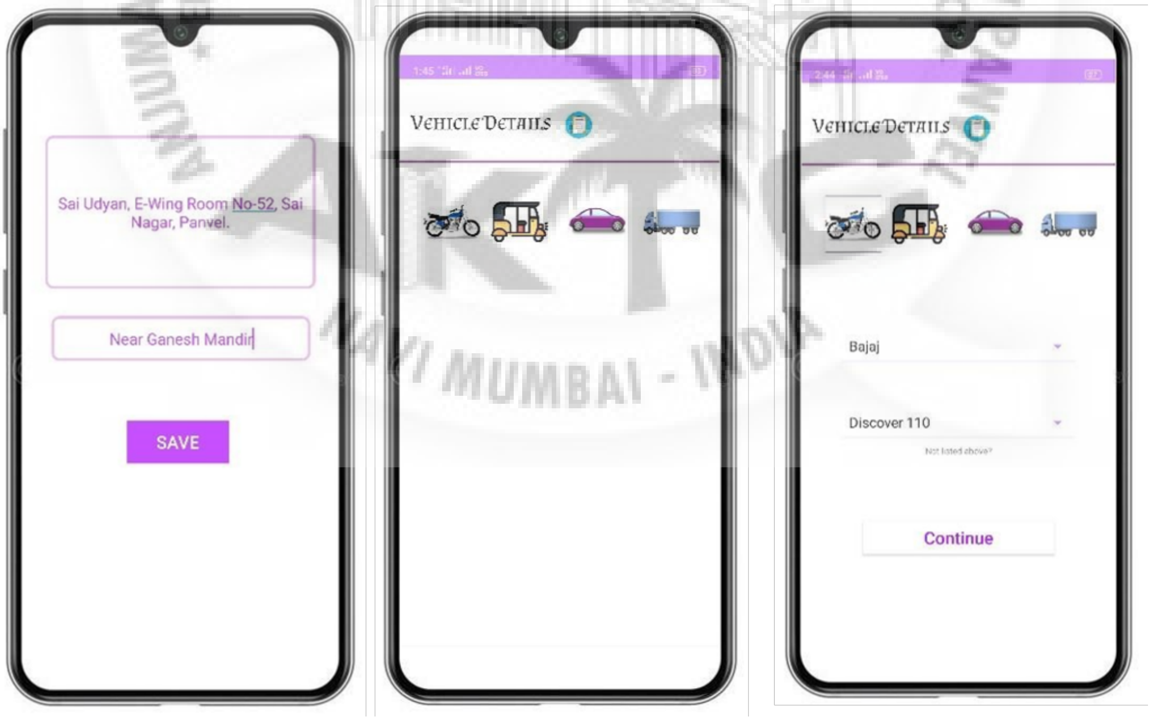
Figure 8.3: Vehicle user functions



Do you want to continue with this address?

Sai Nagar Rd, Sai Nagar, Panvel, Navi Mumbai, Maharashtra 410206, India

Figure 8.4: Locating on map



Sai Udyan, E-Wing Room No-52, Sai Nagar, Panvel.

Near Ganesh Mandir

Bajaj

Discover 110

Figure 8.5: Vehicle details selection

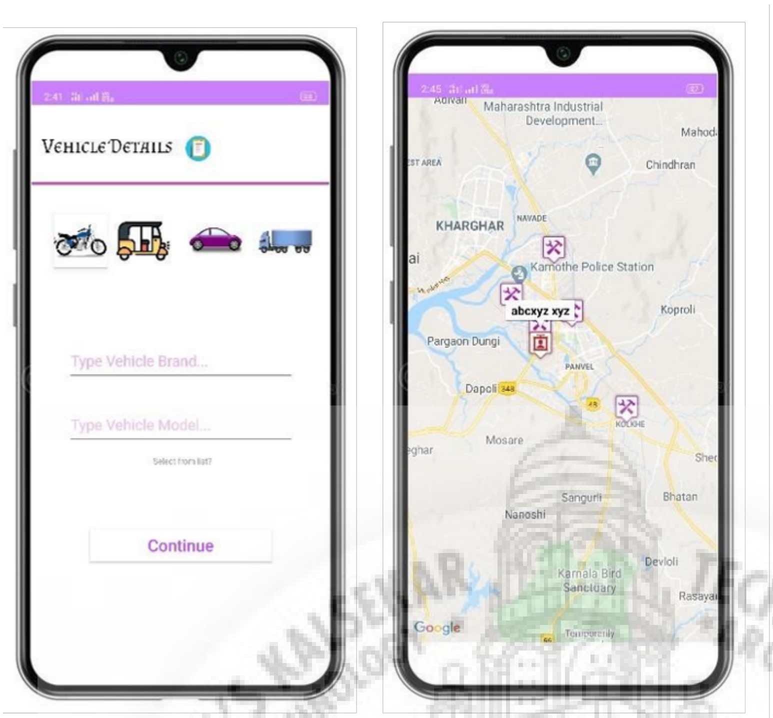
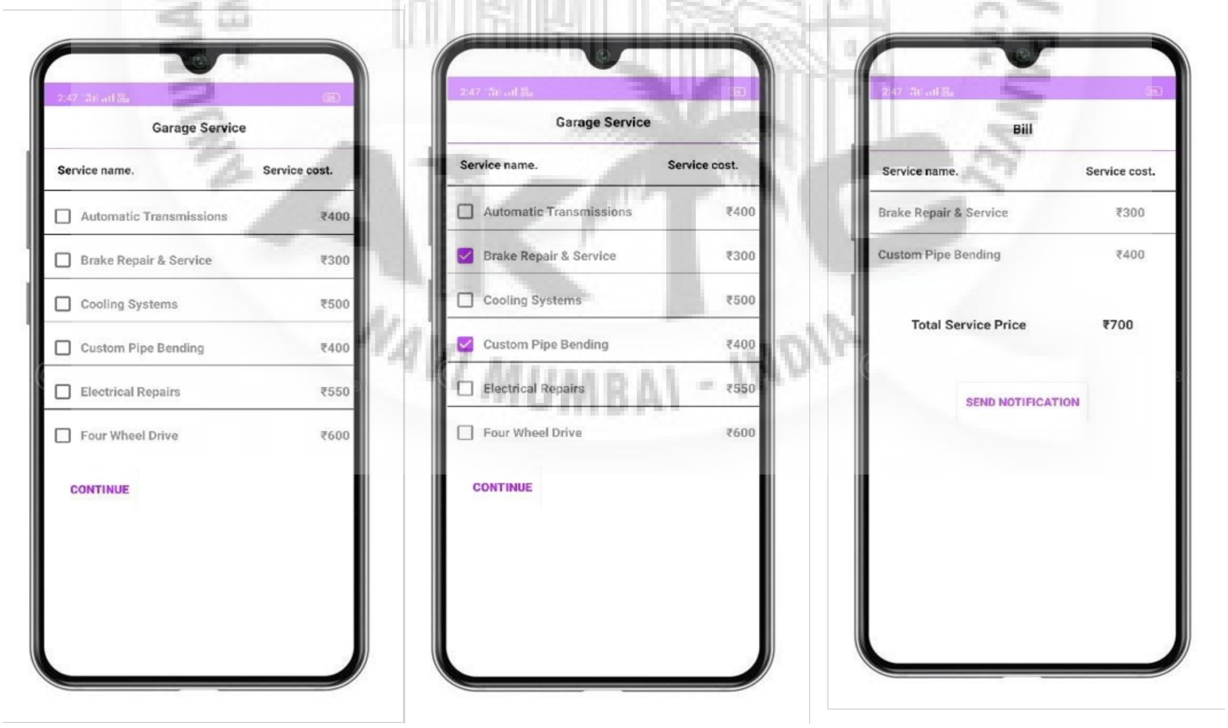


Figure 8.6: Garage selection


Garage Service

Service name.	Service cost.
□ Automatic Transmissions	₹400
□ Brake Repair & Service	₹300
□ Cooling Systems	₹500
□ Custom Pipe Bending	₹400
□ Electrical Repairs	₹550
□ Four Wheel Drive	₹600

CONTINUE

Garage Service

Service name.	Service cost.
□ Automatic Transmissions	₹400
☑ Brake Repair & Service	₹300
□ Cooling Systems	₹500
☑ Custom Pipe Bending	₹400
□ Electrical Repairs	₹550
□ Four Wheel Drive	₹600

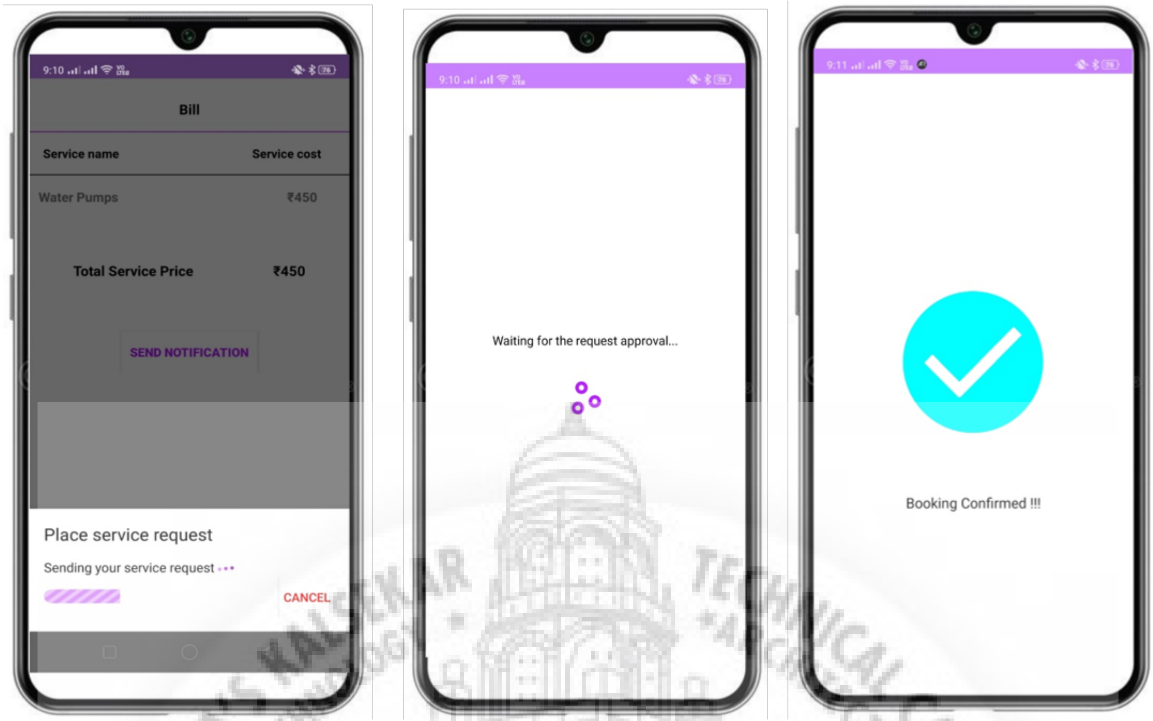
CONTINUE

Bill

Service name.	Service cost.
Brake Repair & Service	₹300
Custom Pipe Bending	₹400
Total Service Price	₹700

SEND NOTIFICATION

Figure 8.7: Service selection



9:10

Bill

Service name	Service cost
Water Pumps	₹450
Total Service Price	₹450

SEND NOTIFICATION

Place service request

Sending your service request •••

CANCEL

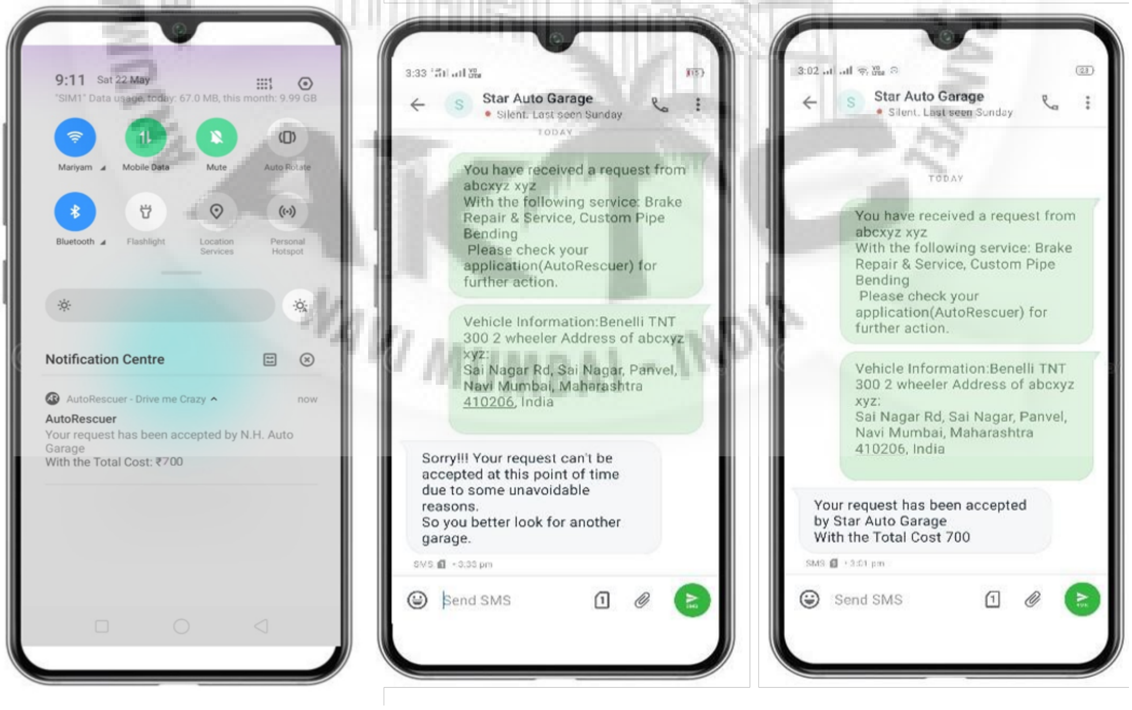
9:10

Waiting for the request approval...

9:11

Booking Confirmed !!!

Figure 8.8: Service request status



9:11 Sat 22 May

"SIM1" Data usage today: 67.0 MB, this month: 9.99 GB

Notification Centre

AutoRescuer - Drive me Crazy now

AutoRescuer
Your request has been accepted by N.H. Auto Garage
With the Total Cost: ₹700

3:33

Star Auto Garage
Silent. Last seen Sunday

TODAY

You have received a request from abcxyz xyz
With the following service: Brake Repair & Service, Custom Pipe Bending
Please check your application(AutoRescuer) for further action.

Vehicle Information:Benelli TNT 300 2 wheeler Address of abcxyz xyz:
Sai Nagar Rd, Sai Nagar, Panvel, Navi Mumbai, Maharashtra 410206, India

Sorry!!! Your request can't be accepted at this point of time due to some unavoidable reasons.
So you better look for another garage.

SMS • 3:33 pm

Send SMS

3:02

Star Auto Garage
Silent. Last seen Sunday

TODAY

You have received a request from abcxyz xyz
With the following service: Brake Repair & Service, Custom Pipe Bending
Please check your application(AutoRescuer) for further action.

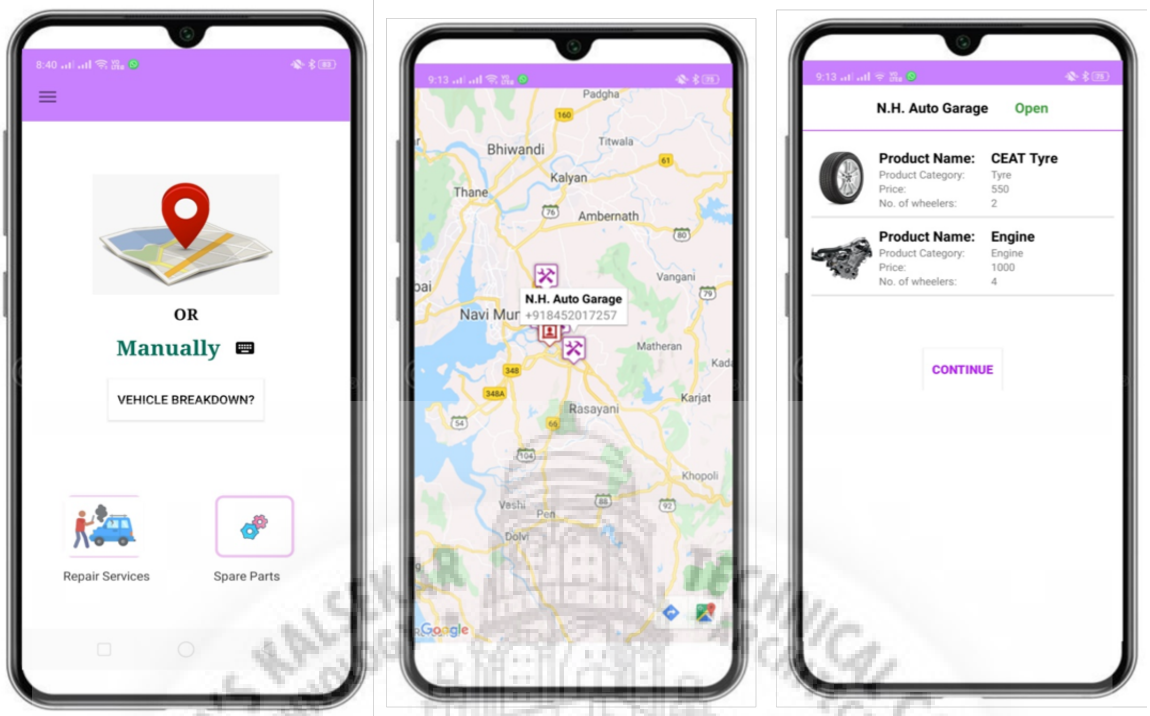
Vehicle Information:Benelli TNT 300 2 wheeler Address of abcxyz xyz:
Sai Nagar Rd, Sai Nagar, Panvel, Navi Mumbai, Maharashtra 410206, India

Your request has been accepted by Star Auto Garage
With the Total Cost 700

SMS • 3:01 pm

Send SMS

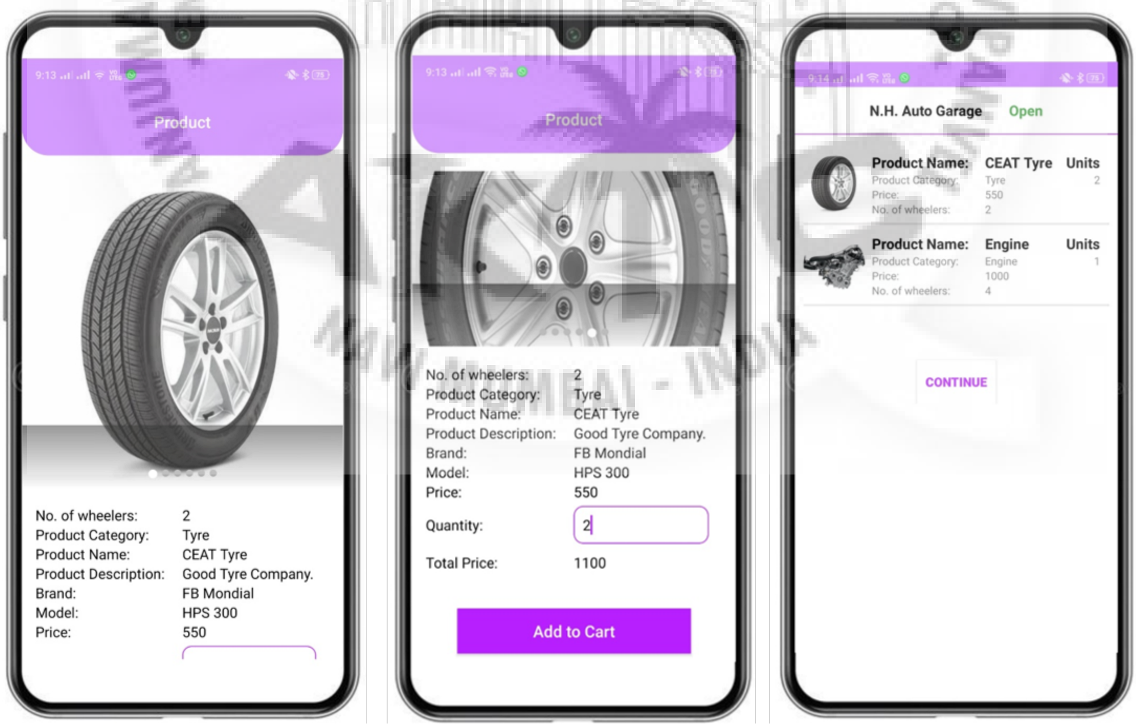
Figure 8.9: Notification status



N.H. Auto Garage Open

Product Name:	CEAT Tyre
Product Category:	Tyre
Price:	550
No. of wheelers:	2
Product Name:	Engine
Product Category:	Engine
Price:	1000
No. of wheelers:	4

Figure 8.10: Garage selection for spare parts



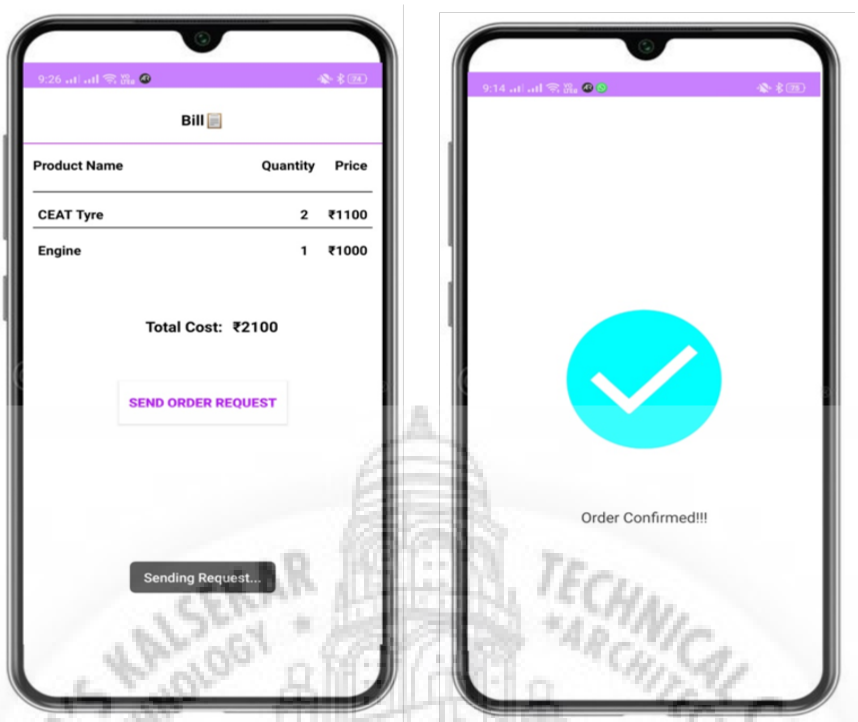
No. of wheelers:	2
Product Category:	Tyre
Product Name:	CEAT Tyre
Product Description:	Good Tyre Company.
Brand:	FB Mondial
Model:	HPS 300
Price:	550

No. of wheelers:	2
Product Category:	Tyre
Product Name:	CEAT Tyre
Product Description:	Good Tyre Company.
Brand:	FB Mondial
Model:	HPS 300
Price:	550
Quantity:	2
Total Price:	1100

N.H. Auto Garage Open

Product Name:	CEAT Tyre	Units	
Product Category:	Tyre		2
Price:	550		
No. of wheelers:	2		
Product Name:	Engine	Units	
Product Category:	Engine		1
Price:	1000		
No. of wheelers:	4		

Figure 8.11: spare parts information



Bill

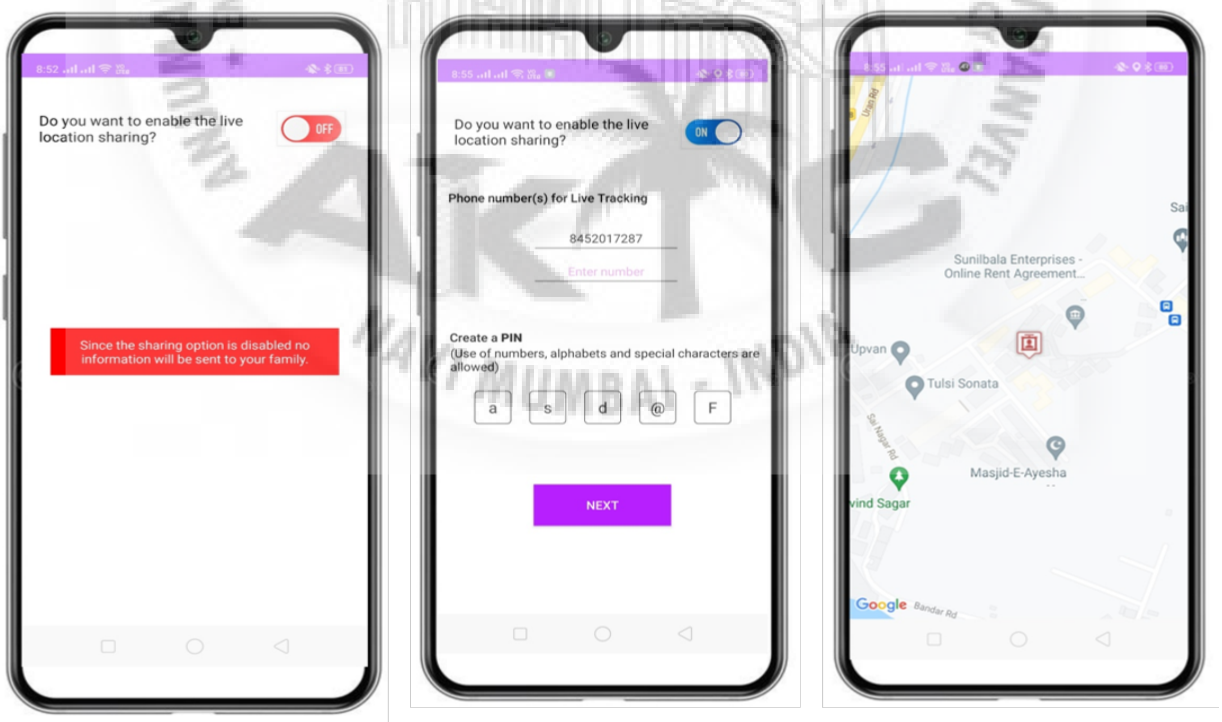
Product Name	Quantity	Price
CEAT Tyre	2	₹1100
Engine	1	₹1000

Total Cost: ₹2100

Sending Request...

Order Confirmed!!!

Figure 8.12: order details and order status



Do you want to enable the live location sharing?

Since the sharing option is disabled no information will be sent to your family.

Do you want to enable the live location sharing?

Phone number(s) for Live Tracking

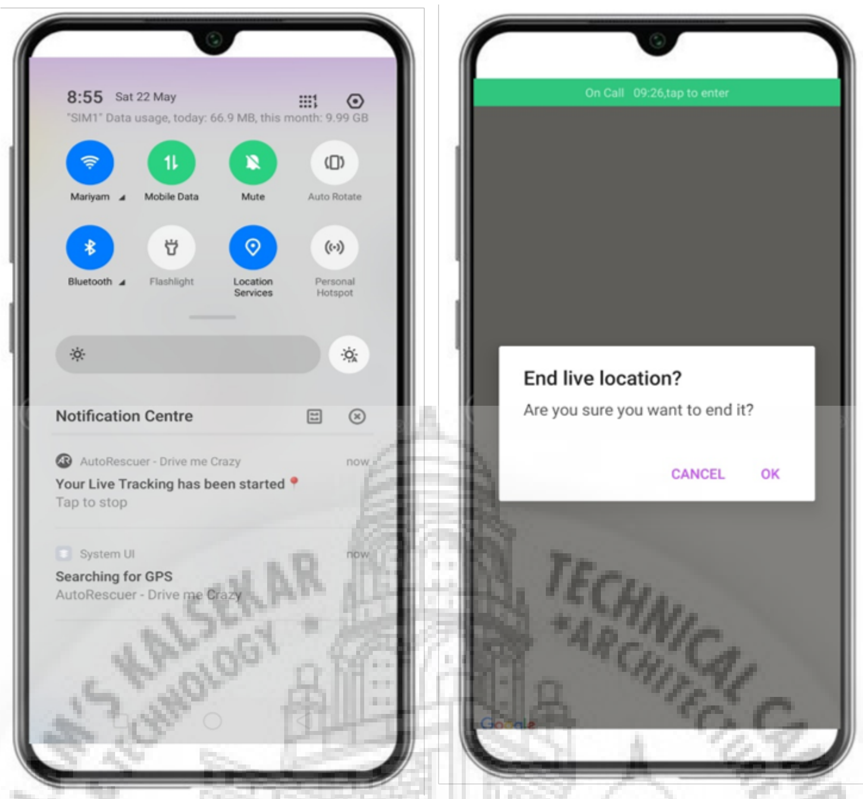
8452017287

Enter number

Create a PIN
(Use of numbers, alphabets and special characters are allowed)

a s d @ F

Figure 8.13: Enable live tracking



8:55 Sat 22 May

"SIM1" Data usage, today: 66.9 MB, this month: 9.99 GB

Notification Centre

AutoRescuer - Drive me Crazy now

Your Live Tracking has been started

Tap to stop

System UI now

Searching for GPS

AutoRescuer - Drive me Crazy

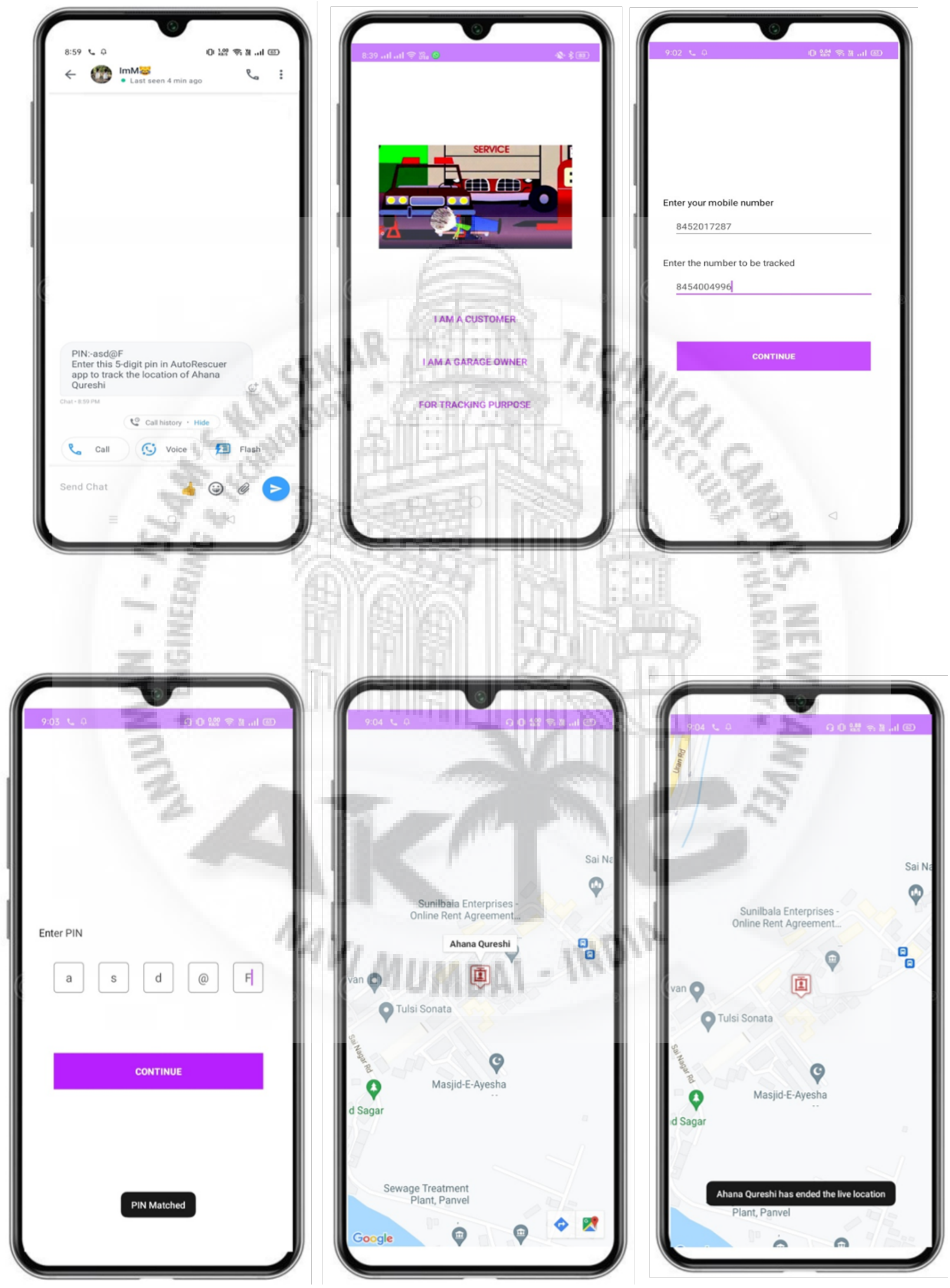
On Call 09:26,tap to enter

End live location?

Are you sure you want to end it?

Figure 8.14: Live tracking status

8.2 Live tracking:



ImM😸
Last seen 4 min ago

PIN:-asd@F
Enter this 5-digit pin in AutoRescuer app to track the location of Ahana Qureshi

Chat • 8:59 PM

Enter your mobile number: 8452017287

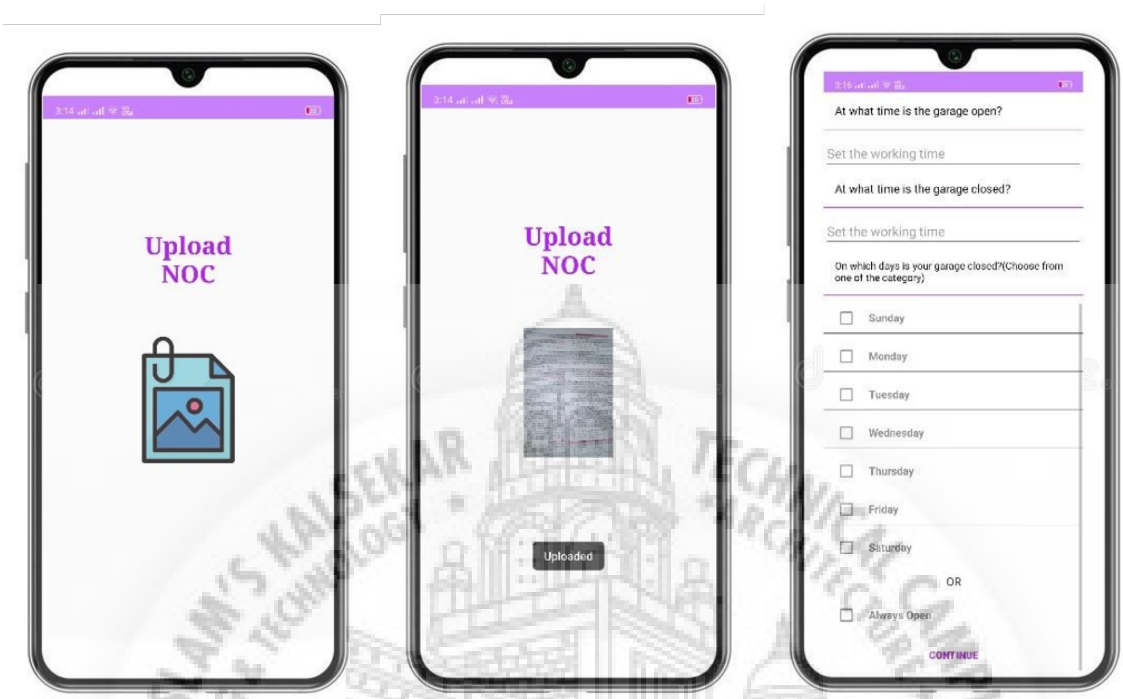
Enter the number to be tracked: 8454004996

Enter PIN: a s d @ F

Ahana Qureshi has ended the live location

Figure 8.15: Live tracking module

8.3 Garage owner side:



Upload NOC

Upload NOC

Uploaded

At what time is the garage open?

Set the working time

At what time is the garage closed?

Set the working time

On which days is your garage closed?(Choose from one of the category)

□ Sunday

□ Monday

□ Tuesday

□ Wednesday

□ Thursday

□ Friday

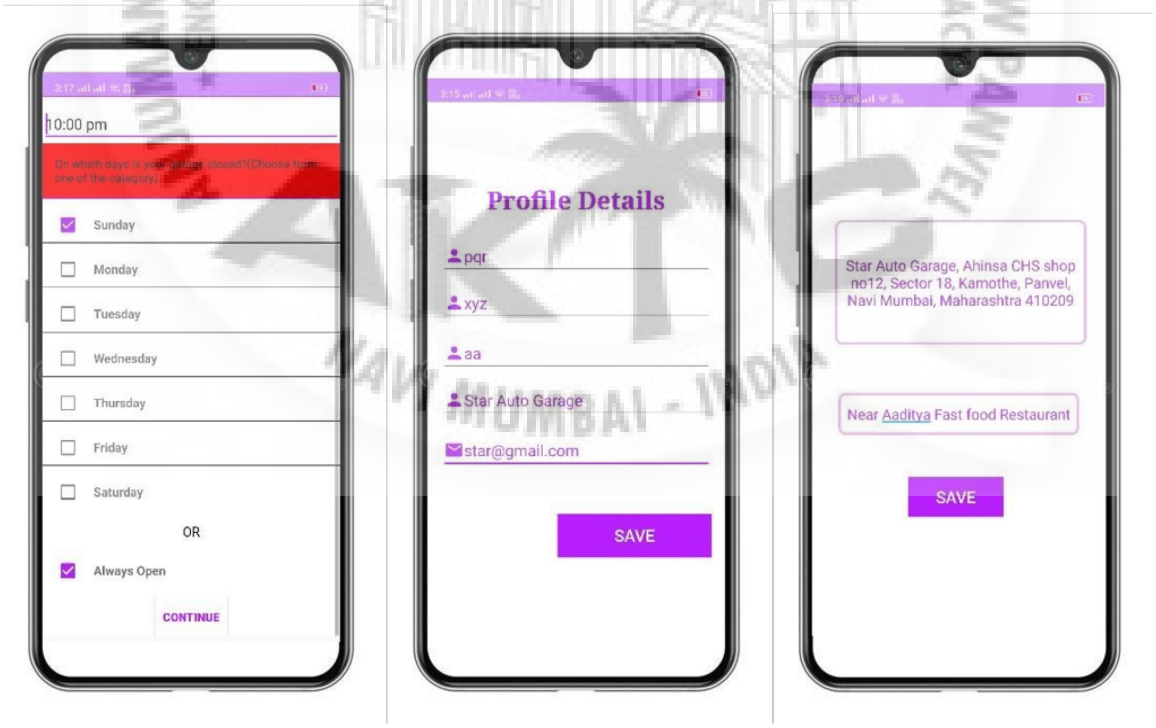
□ Saturday

OR

□ Always Open

CONTINUE

Figure 8.16: Garage registration process



10:00 pm

On which days is your garage closed?(Choose from one of the category)

☑ Sunday

□ Monday

□ Tuesday

□ Wednesday

□ Thursday

□ Friday

□ Saturday

OR

☑ Always Open

CONTINUE

Profile Details

pqr

xyz

aa

Star Auto Garage

star@gmail.com

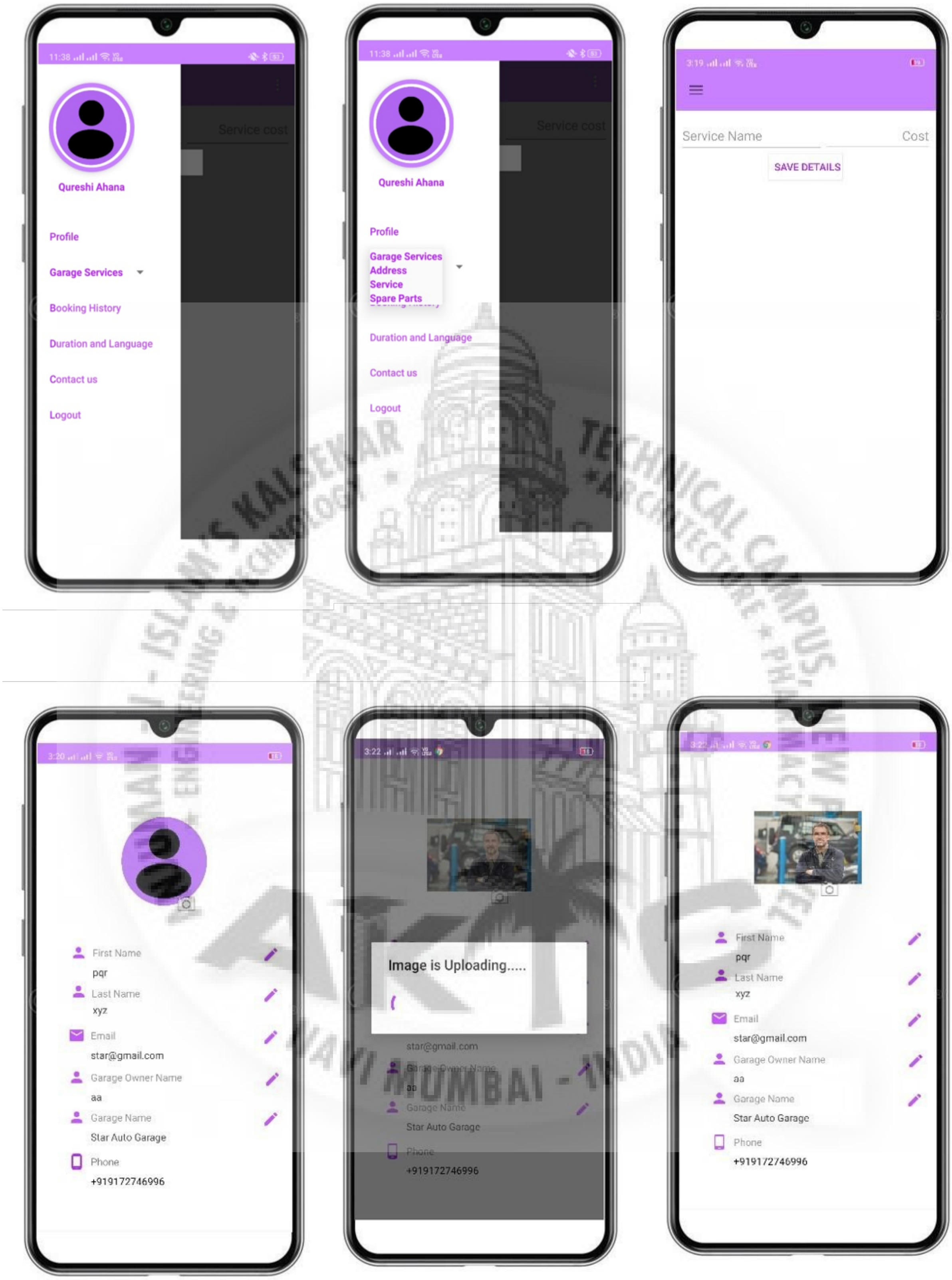
SAVE

Star Auto Garage, Ahinsa CHS shop no12, Sector 18, Kamothe, Panvel, Navi Mumbai, Maharashtra 410209

Near Aaditya Fast food Restaurant

SAVE

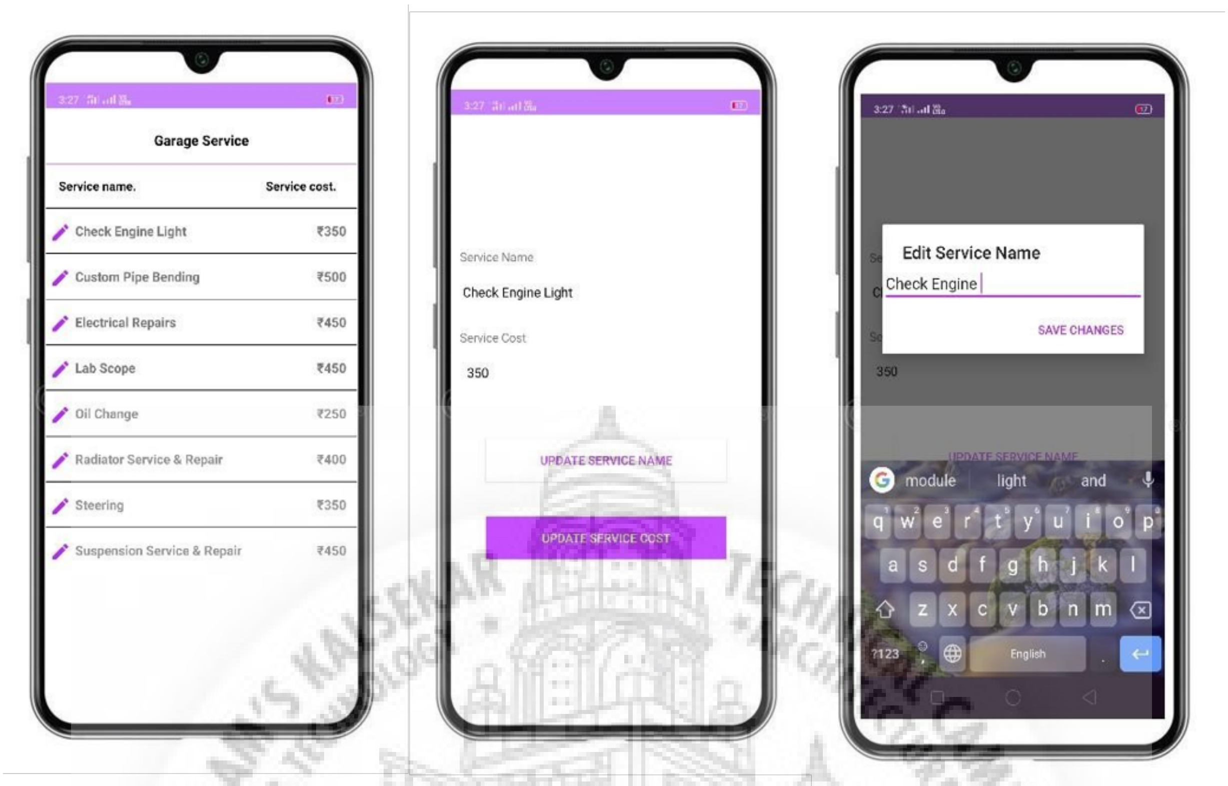
Figure 8.17: Garage information



First Name
pqr
Last Name
xyz
Email
star@gmail.com
Garage Owner Name
aa
Garage Name
Star Auto Garage
Phone
+919172746996

First Name
pqr
Last Name
xyz
Email
star@gmail.com
Garage Owner Name
aa
Garage Name
Star Auto Garage
Phone
+919172746996

Figure 8.18: Garage function module



Garage Service

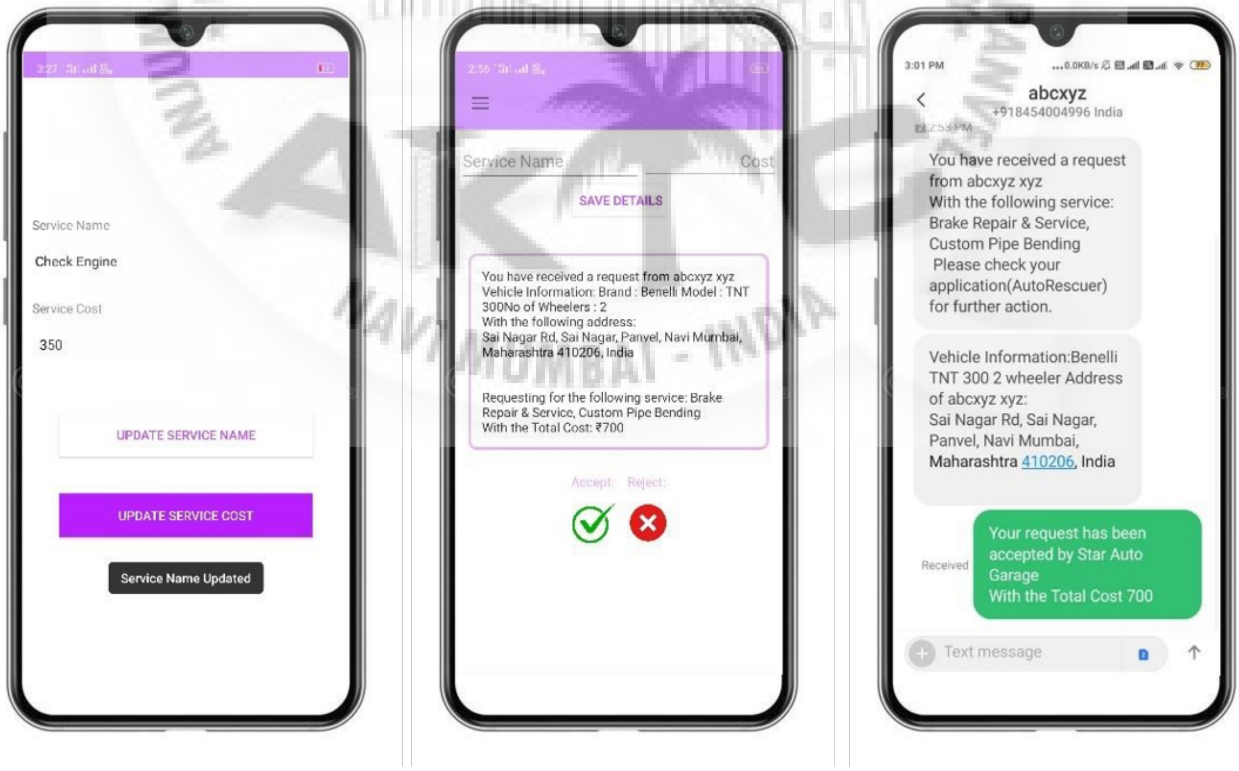
Service name.	Service cost.
Check Engine Light	₹350
Custom Pipe Bending	₹500
Electrical Repairs	₹450
Lab Scope	₹450
Oil Change	₹250
Radiator Service & Repair	₹400
Steering	₹350
Suspension Service & Repair	₹450

Service Name: Check Engine Light

Service Cost: 350

Edit Service Name: Check Engine

Figure 8.19: Garage services with cost



Service Name: Check Engine

Service Cost: 350

Service Name Updated

You have received a request from abcxyz xyz
Vehicle Information: Brand : Benelli Model : TNT 300No of Wheelers : 2
With the following address:
Sai Nagar Rd, Sai Nagar, Panvel, Navi Mumbai, Maharashtra 410206, India

Requesting for the following service: Brake Repair & Service, Custom Pipe Bending
With the Total Cost: ₹700

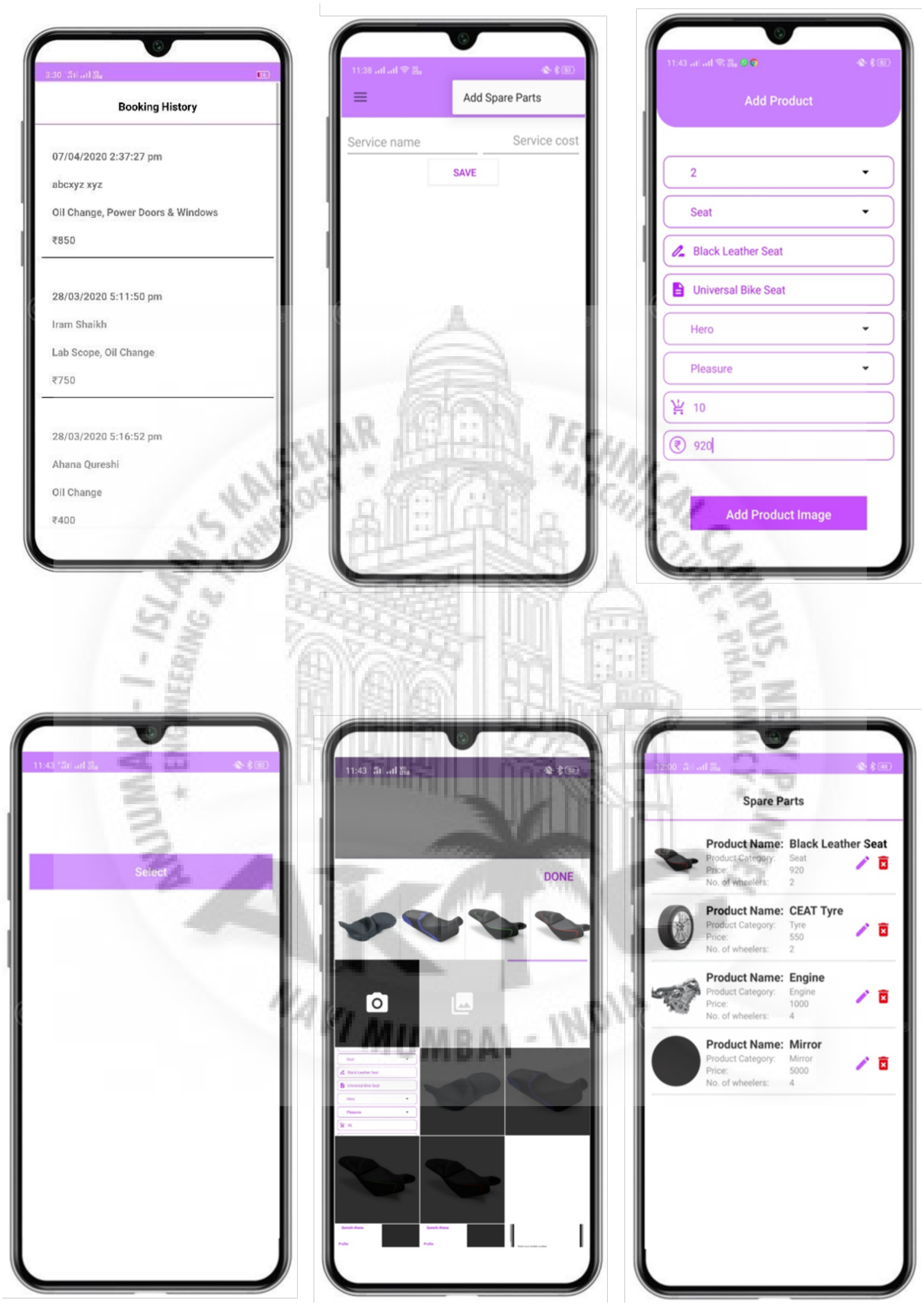
abcxyz
+918454004996 India

You have received a request from abcxyz xyz
With the following service:
Brake Repair & Service,
Custom Pipe Bending
Please check your application(AutoRescuer) for further action.

Vehicle Information:Benelli TNT 300 2 wheeler Address of abcxyz xyz:
Sai Nagar Rd, Sai Nagar, Panvel, Navi Mumbai, Maharashtra 410206, India

Your request has been accepted by Star Auto Garage
With the Total Cost 700

Figure 8.20: User's service request status



Booking History

07/04/2020 2:37:27 pm

abcxyz xyz

Oil Change, Power Doors & Windows

₹850

28/03/2020 5:11:50 pm

Iram Shaikh

Lab Scope, Oil Change

₹750

28/03/2020 5:16:52 pm

Ahana Qureshi

Oil Change

₹400

Add Spare Parts

Service name | Service cost

SAVE

Add Product

2

Seat

Black Leather Seat

Universal Bike Seat

Hero

Pleasure

10

920

Add Product Image

Select

DONE

Spare Parts

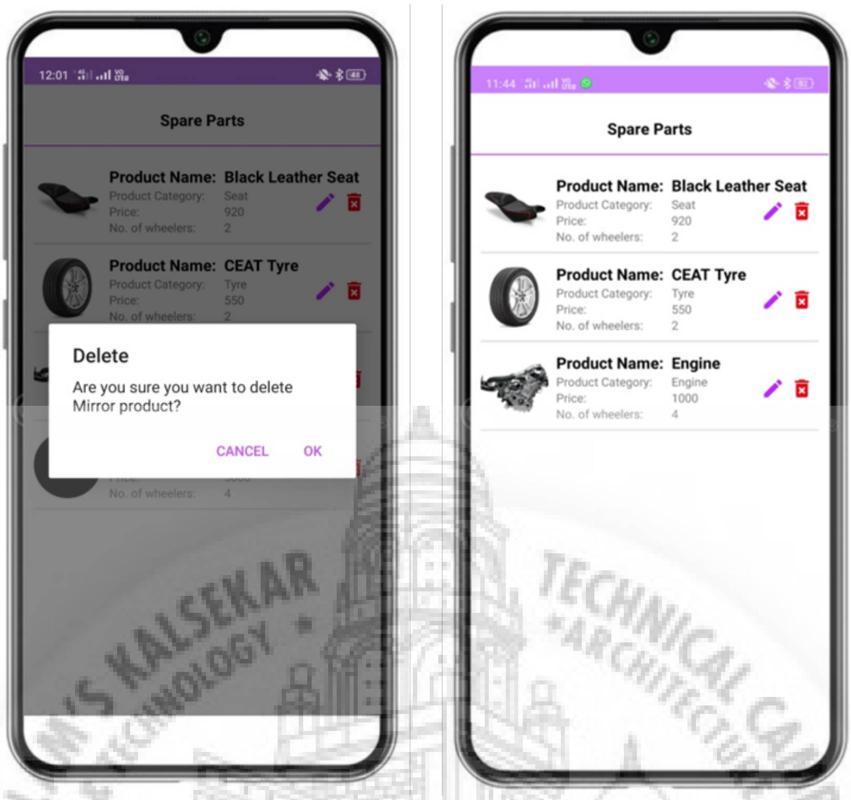
Product Name: Black Leather Seat
Product Category: Seat
Price: 920
No. of wheelers: 2

Product Name: CEAT Tyre
Product Category: Tyre
Price: 550
No. of wheelers: 2

Product Name: Engine
Product Category: Engine
Price: 1000
No. of wheelers: 4

Product Name: Mirror
Product Category: Mirror
Price: 5000
No. of wheelers: 4

Figure 8.21: Booking history and spare parts with cost



12:01

Spare Parts

Product Name: Black Leather Seat
Product Category: Seat
Price: 920
No. of wheelers: 2

Product Name: CEAT Tyre
Product Category: Tyre
Price: 550
No. of wheelers: 2

Delete

Are you sure you want to delete Mirror product?

CANCEL OK

No. of wheelers: 4

11:44

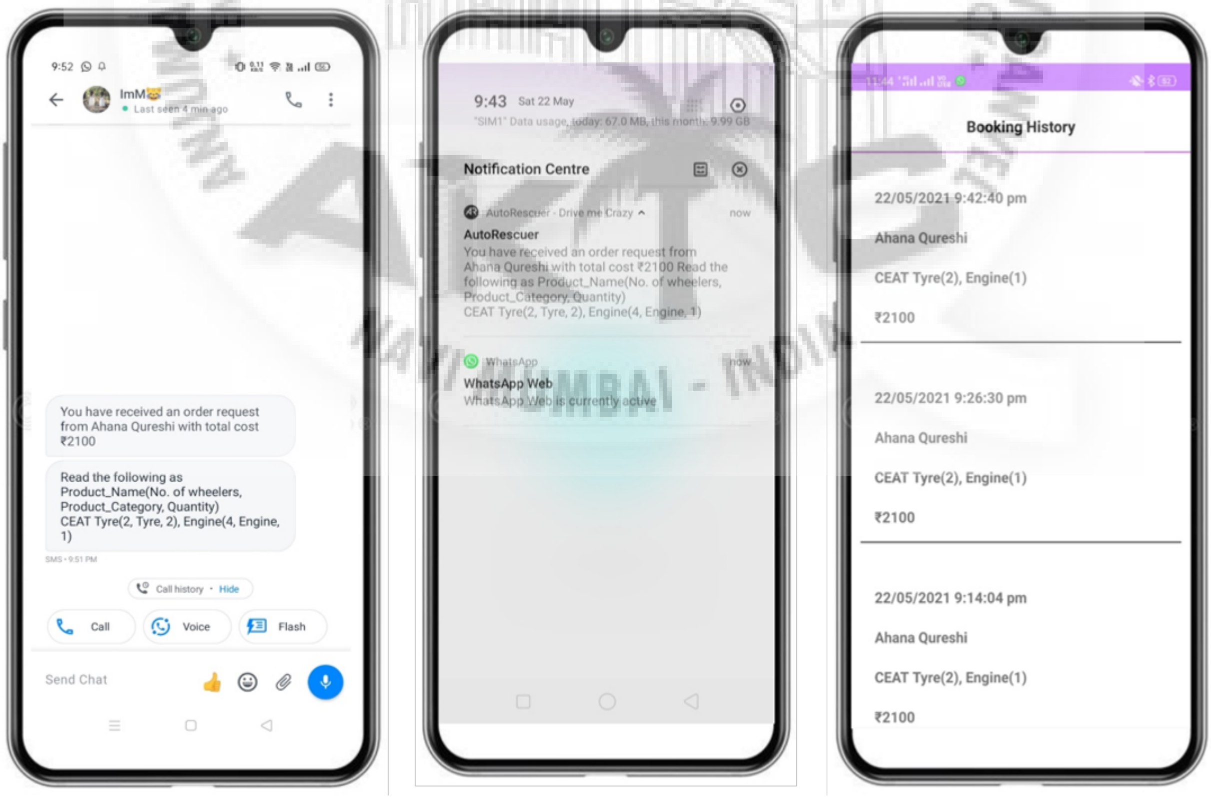
Spare Parts

Product Name: Black Leather Seat
Product Category: Seat
Price: 920
No. of wheelers: 2

Product Name: CEAT Tyre
Product Category: Tyre
Price: 550
No. of wheelers: 2

Product Name: Engine
Product Category: Engine
Price: 1000
No. of wheelers: 4

Figure 8.22: spare parts with cost



9:52

ImM

Last seen 4 min ago

You have received an order request from Ahana Qureshi with total cost ₹2100

Read the following as Product_Name(No. of wheelers, Product_Category, Quantity)
CEAT Tyre(2, Tyre, 2), Engine(4, Engine, 1)

SMS • 9:51 PM

Call history • Hide

Call Voice Flash

Send Chat

9:43 Sat 22 May

"SIM1" Data usage, today: 67.0 MB, this month: 9.99 GB

Notification Centre

AutoRescuer - Drive me Crazy now

AutoRescuer

You have received an order request from Ahana Qureshi with total cost ₹2100 Read the following as Product_Name(No. of wheelers, Product_Category, Quantity)
CEAT Tyre(2, Tyre, 2), Engine(4, Engine, 1)

WhatsApp now

WhatsApp Web

WhatsApp Web is currently active

11:44

Booking History

22/05/2021 9:42:40 pm

Ahana Qureshi

CEAT Tyre(2), Engine(1)

₹2100

22/05/2021 9:26:30 pm

Ahana Qureshi

CEAT Tyre(2), Engine(1)

₹2100

22/05/2021 9:14:04 pm

Ahana Qureshi

CEAT Tyre(2), Engine(1)

₹2100

Figure 8.23: Booking history and spare parts

8.4 Admin Side:

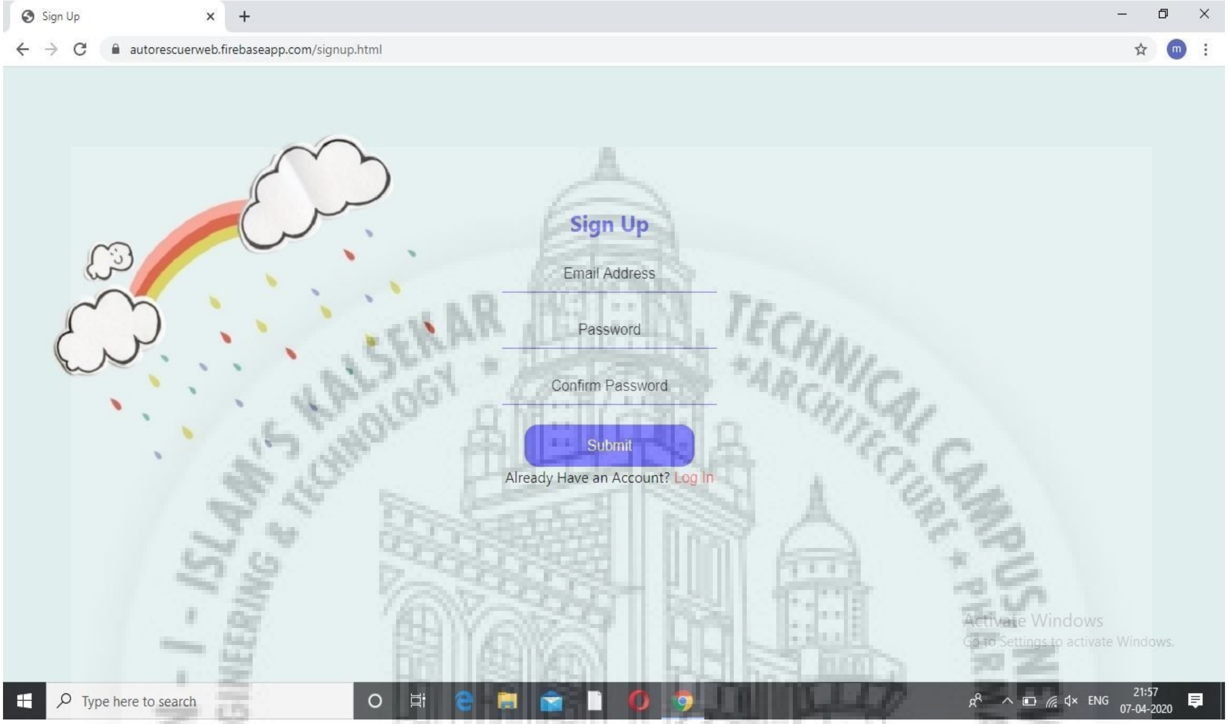


Figure 8.24: Sign-up page for Admin

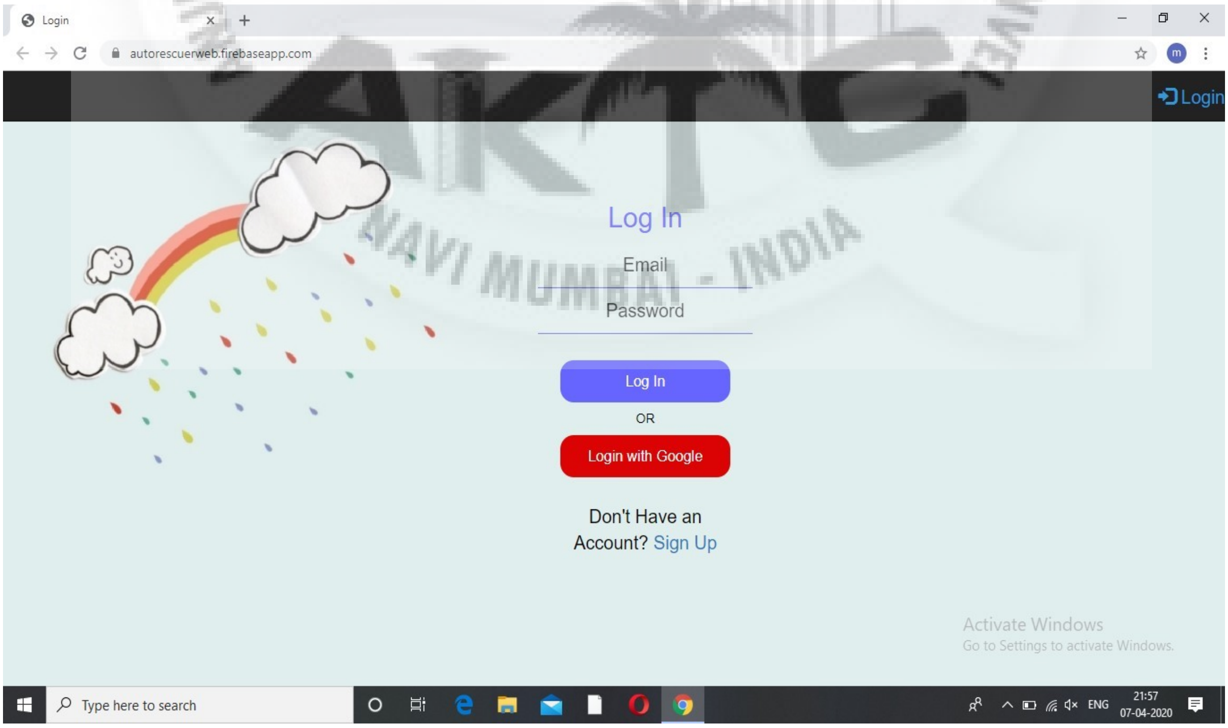
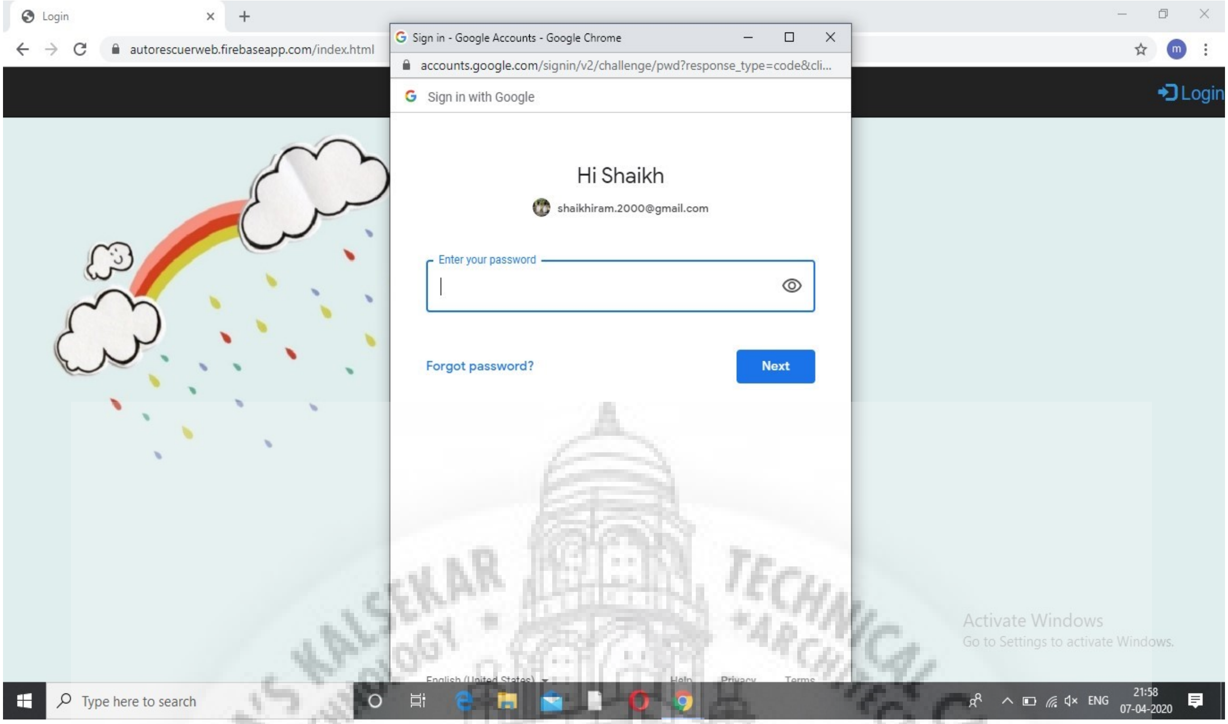


Figure 8.25: Login page for Admin

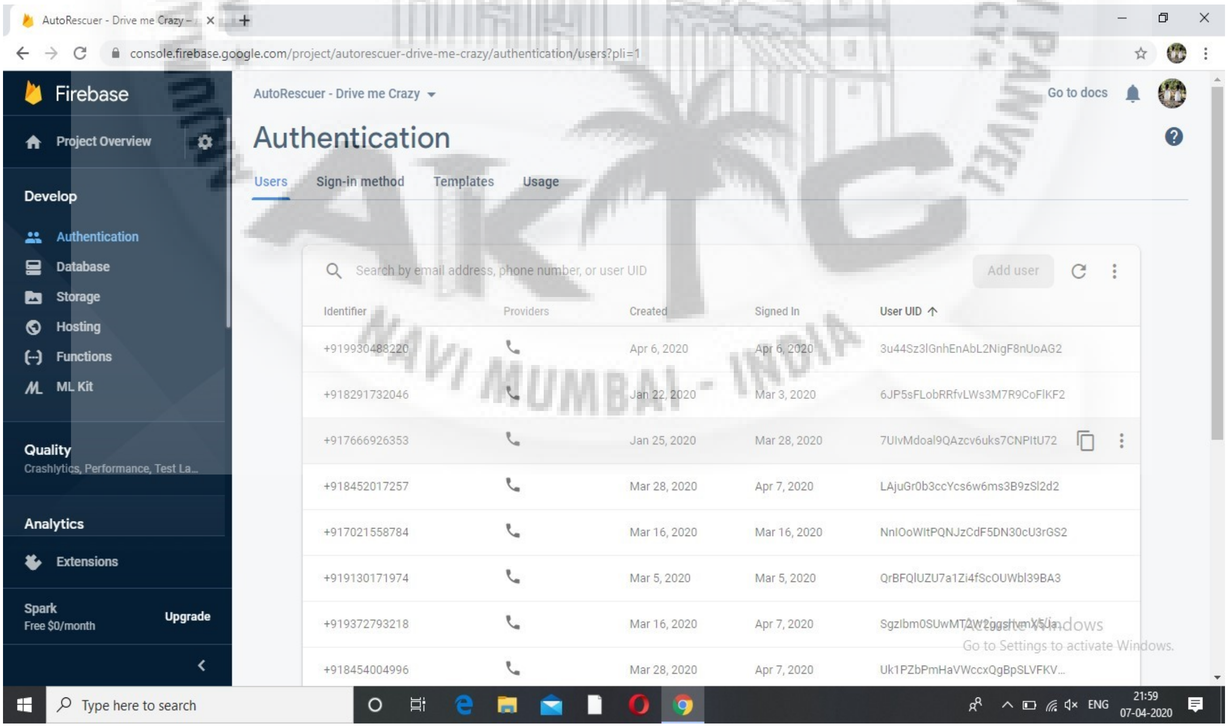

Login
autorescuerweb.firebaseapp.com/index.html
Sign in - Google Accounts - Google Chrome
accounts.google.com/signin/v2/challenge/pwd?response_type=code&cli...
Sign in with Google
Login
Forgot password?
Next
Type here to search
ENG
21:58
07-04-2020

Hi Shaikh

shaikhiram.2000@gmail.com

Enter your password

Figure 8.26: Login via Gmail


AutoRescuer - Drive me Crazy
console.firebase.google.com/project/autorescuer-drive-me-crazy/authentication/users?pli=1
Firebase
Project Overview
Develop
Authentication
Database
Storage
Hosting
Functions
ML Kit
Quality
Crashlytics, Performance, Test La...
Analytics
Extensions
Spark
Free $0/month
Upgrade
AutoRescuer - Drive me Crazy
Go to docs
Users
Sign-in method
Templates
Usage
Search by email address, phone number, or user UID
Add user
Type here to search
ENG
21:59
07-04-2020

Authentication

Identifier	Providers	Created	Signed In	User UID ↑
+919930488220	☎	Apr 6, 2020	Apr 6, 2020	3u44Sz3lGnhEnAbL2NigF8nUoAG2
+918291732046	☎	Jan 22, 2020	Mar 3, 2020	6JP5sFLobRRfvLWs3M7R9CoFlKF2
+917666926353	☎	Jan 25, 2020	Mar 28, 2020	7UIvMdoal9QAzcv6uks7CNPItU72
+918452017257	☎	Mar 28, 2020	Apr 7, 2020	LAjuGr0b3ccYcs6w6ms3B9zSl2d2
+917021558784	☎	Mar 16, 2020	Mar 16, 2020	NnIOoWItPQNJzCdF5DN30cU3rGS2
+919130171974	☎	Mar 5, 2020	Mar 5, 2020	QrBFQlUZU7a1Zi4fScOUWbl39BA3
+919372793218	☎	Mar 16, 2020	Apr 7, 2020	SgzIbm0SUwMT2W2ggsHvmX5Ua
+918454004996	☎	Mar 28, 2020	Apr 7, 2020	Uk1PZbPmHaVWccxQgBpSLVFKV...

Figure 8.27: Firebase database

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

The proposed system has been designed and developed to work efficiently. It is very useful, the user can easily use this application. Any user can track the mechanic shop using their mobile from any location any time using this application. The application is free of cost and does not require any additional device. The application will store users GPS data and can proceed with suitable recommendations regarding service centers.

This application is reliable to use. We can implement it on a large scale if it works really well on a small scale. We can make it way better by implementing various new technologies and programming language which are getting introduced daily in this vast world. We can work on different aspects of this system to make it look way better than it is today, and it is our duty that we should try at least to take it further.
Every system has some limitations and yes our system also has some limitations on the database. It will have some time constraints too. Limitations also occurs on availability on mechanic and services that if a person wants something and it is not available then the person needs to compromise with his/her need.

9.2 Future Scope

- PUC and other vehicle related reminder.
- Emergency contact.
- Videos related to vehicle services.
- Making a complete deployable application.
- Feedback and rating.

References

[1] Tracking of Automobile Service Centers Using Android Application(visit mechanic) https://ieeexplore.ieee.org/abstract/document/9121172 Published in: 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)

[2] Mobile Mechanic – An innovative step towards Digital Automobile Service https://ieeexplore.ieee.org/document/8537243 Published in:2018 International Conference on Smart City and Emerging Technology (ICSCET)

[3] Reduction of servicing and maintenance time of car – a future need- https://ieeexplore.ieee.org/document/7449161 Published in:2016 International Conference on Advances in Human Machine Interaction (HMI)

[4] www.mytvsaccessories.com

[5] https://iwheels.co/

[6] https://tracxn.com/d/trending-themes/Startups-in-Automobile-Repair