Mechanic on the go

Project ID: Winter-2023-20

Session: BSCS Winter 2023

Project Advisor: Muhammad Abdullah

Submitted By

Asmar Aftab	70066819
Rana Ammar Talat	70070400
Sameed Akhtar Nawaz	70066860

Department of Computer Science & IT
The University of Lahore
Lahore, Pakistan

Declaration

We have read the project guidelines and we understand the meaning of academic dishonesty, in particular plagiarism and collusion. We hereby declare that the work we submitted for our final year project, entitled **Mechanic on the go** is original work and has not been printed, published or submitted before as final year project, research work, publication or any other documentation.

Group Member 1 Name: Asmar Aftab
SAP No: 70066819

Signature: **……**

Group Member 2 Name: Rana Ammar Talat
SAP No: 70070400

Signature: **……**

Group Member 3 Name: Sameed Akhtar Nawaz
SAP No: 70066860

Signature: **……**

Statement of Submission

This is to certify that **Asmar Aftab** Roll No. **70066819, Rana Ammar Talat** Roll No. **70070400** and **Sameed Akhtar Nawaz** Roll No. **70066860** have successfully submitted the final project named as: **Mechanic on the go**, at Computer Science & IT Department, The University of Lahore, Lahore Pakistan, to fulfill the partial requirement of the degree of **BS in Computer Science**.

Supervisor Name: Muhammad Abdullah

Signature: ……
Date: ……

Dedication

The members of our group ask our revered parents and teachers to assist us in every aspect of your life, whether it be moral or financially. Because of their belief on us we could be able to complete this document. Our group members dedicate this project to our teacher because of their effort we are capable to bring our idea in reality. For the benefit of our teachers and parents, we designed our Mechanic on the go project.

Acknowledgement

It is by the Beauty of Allah All-powerful, the Ruler and Creature of this Universe. whose power and glory enable all things to be accomplished, as well as his Prophet, PBUH, who will forever serve as a source of instruction and knowledge for the entire human race. We are grateful to Allah the All-Powerful for making it possible for us to complete this project and overcome all of its obstacles.

We would like to pay thanks to our project supervisor **"Mr Muhammad Abdullah"** Lecturer, The University of Lahore for his exceptional counsel and Support. We were able to deal with issues that arose throughout the project because of his expertise and knowledge. This project was made possible for us by his accommodating suggestions, constant encouragement, and generous supervision.

We would also like to thank all the teachers of Department of Computer Science, The University of Lahore, who helped us a lot during all the semesters. They always motivate us how to respond positively to criticism. We are professionals in the field of computer science because of their unwavering support.

Date .....

Abstract

Our project titles **"Mechanic on the go"** is a problem-solving application. A person who is traveling may face emergency scenarios like accident, tire burst, engine failure, low gas. In these circumstances people usually may get panic and mostly searching different platform may get confusing and time-consuming. Our app then will be best solution for users as it is the complete package of road emergency solution. This project is designed to accommodate both travelers and the mechanics. Anybody can use this app by just signing up or by calling/declaring emergency. Whenever the person is in problem regarding his/her vehicle he/she can use this app to locate the mechanics or wreckers around him. This app will be a very helpful execution in case of emergency This app will be providing opportunities to many mechanics as their ranking will be provided by users either they are on highway or in another city. As we are also targeting local mechanics and wreckers so it will be giving value to the people who are working day and night to get some good amount of their work.

Area of the Project

Area of project is Mobile Application Development.

Technologies used

- React Native
- React JS
- Node JS
- Express JS
- MongoDB

List of Figures

List of Tables

Table of Content

Chapter 1: Introduction to the Problem

1.1 Introduction

As Earth is becoming a global village everyone seeks for fast and reliable solutions for their problems. The travelers traveling on motorway got traveling emergency facilities easily and precisely but in local areas like subways, GT road, and other local connected roads lack of those on way facilities, our app is working to provide those services to those needy people. As our app has emergency services (motorway, highway police, fueling station) and can provide the mechanic from well-reputed companies or user can seek help from the closest private mechanics through this application.

You can get wrecker and mechanic by clicking our app and notifying that you simply need a mechanic or wrecker. Our app will then ask you for your current location and locate a mechanic or wrecker who is nearby.

You will be asked about your car problem if you know about your problem then you can fill the blank form otherwise you can say that you are not familiar with the problem. Then you will be given a data for the mechanics after that if you confirm that mechanic then his car/bike registration number will be provided to you to sure that you get the right guy. Once mechanic have received your location, information regarding your problem and your vehicle (like vehicle registration number, name, company name and model).

Our app will send you the notification that the mechanic is on his way to you and will show you his live location so you will be able to know when he will come to you. After getting services form the mechanic app will ask you to rate the experience. Same scenario works for wrecker you will locate the wrecker and the wrecker will come to your location and will take you to the closest mechanic.

We make sure that you deploy an excellent looking app that's simple to use, smooth operational and intuitive within the display. The key principles of an excellent interface design Simplicity, Familiarity, Clarity, Digestibility and we are following them.

1.2 Purpose

Our project is a problem-solving application. Let suppose a person is traveling and he faces any of these scenarios like accident, tire burst, engine failure, low gas. In these scenarios usually people may get panic and mostly searching different platform may get confusing and time-consuming. Our app then will be best solution for users as it is the complete package of road emergency solution.

This project is designed to accommodate both travelers and the mechanics. Anybody can use this app by just signing up or by calling/declaring emergency. Whenever the person is in problem regarding his/her vehicle he/she can use this app to locate the mechanics or wreckers around him. This app will be a very helpful execution in case of emergency This app will be providing opportunities to many mechanics as their ranking will be provided by users either they are on highway or in another city. As we are also targeting local mechanics and wreckers so it will be giving value to the people who are working day and night to get some good amount of their work.

1.3 Objective

Define your objectives what you want to achieve after the completion of the project This app will be helpful for all the people in this world after its implementation as it can work at any place working on the GPS of a device. As this app is performing as a helping tool for the people in need or emergency.

This app is easy to use and will have a friendly interface so that the user can get the help in case of emergency if they are traveling or having any issue with their car. Either they are in the street or on the GT road they can seek the help by just pressing a button.

Time and safety are the most important thing that this project will be giving value to. As if a person is traveling with some passengers or family and face some issues with the car than this app will be his priority as he will get the help of mechanic or wrecker to carry out his vehicle without leaving his vehicle alone with passengers. This will leap a step forward to close the technology gap.

1.4 Existing Solution

There are many applications that are working related to this application but the problem with them is they are working in a specific domain as some of related applications are below:

UBER: [1]

Uber is an application that connects passengers with drivers. To order a vehicle it is necessary to have a smartphone and to register within the mobile application by entering your name, email address, a cell phone number and MasterCard details if you want to be billed automatically at the end of the ride. Global positioning system within smartphone is used to determine the location of

driver and rider. The ride order appears on the nearest driver's smartphone application, and he/she can accept or reject the ride.

CAREEM: [2]

Careem is a company that provides an app-based car service that connects people to rides. It designs a web platform that gives a marketplace of community drivers where passengers can request a ride, see the value, and have it charged to their selected MasterCard or can give the cash to the driver. Careem app enables users to require a taxi, rent a car or a motorcycle, order food and delivery, shop online, and make digital transactions.

Star Of Service: [3]

Star of Service is a site that provides different professional service providers at one location. Like if a person needs an appointment with a mechanic or need to have appointment of plumber then this website is the best option for him to avail as it has a good reputation and have a lot of professional service providers that are expert in their specified field.

CURB: [4]

Curb is the number one taxi app in US that connects many drivers and riders with each other. They're providing US partners with opportunities to efficiently connect with audiences to satisfy their business goals also as provide riders and therefore the public with relevant local content and advertising to assist them stay connected while on the go

WRENCH: [5]

Wrench is an online auto repair shop without the shop. Their mobile mechanics service all kinds of cars and trucks, offering everything from oil changes and tune ups to brake jobs. Their ASE certified mechanics can perform most jobs right in your driveway or at your parking spot at work supplying you with the liberty to spend some time on more important things.

Your Mechanic: [6]

Your Mechanic is that the industry leader in mobile vehicle repair, maintenance, and inspection, offering 500+ services to consumers and fleets across the U.S. and Canada. They have built the mobility platform of the longer term, enabling us to service vehicles at the purpose of need, increase asset yield, and lower total cost. Their mobile model and cloud-based platform replace shops' physical infrastructure.

Technician.pk:

This company app is in development phase and its functioning are going to be to supply ease to seek out technician for the citizens in their neighborhood. The app will show the ratings and reviews regarding technician. So, a user can select the technician precisely for any fault associated with electronics. Basically Technician.pk is a web portal that intends to supply quality repair services for electrical appliances, computer networking, and solar energy to both residential and industrial areas.

Lookup.pk: [7]

Lookup.pk connects you to local businesses in your neighborhood and cities in Pakistan. They assist you to speak with companies to look, find and review their brands, products, and services. they supply their services everywhere in Pakistan. Automobile repairing and repair mechanical is one among their categories. You can get the appointment of the mechanic for the checkup of your car at your home.

1.5 Proposed Solution

We have high hope that our project has high success rate because this is the first time that this application is launch also it's a need. Mechanic seeking app is one of the apps that is working on SAAS (software as a service) model. Some time we are traveling, and our car got into some technical issue some time there is no mechanic around so in these types of scenarios this application has proper use. Also, it's very hectic to bring good mechanics so this application allow user to avail the best option at a one click. As, there is a rating and reviews which tells user about mechanic and how good is he. So based on these facts our application has high success rate.

As a client(driver) request for mechanic or wrecker and our app react on the request by providing/showing them the mechanics or wrecker that are near to him. So, he(client) can get the solution by telling the mechanic his problem and mechanic provides the solution to the client. And if he needs to carry his/her vehicle to mechanic place he can get the wrecker at his location. What does not exist in our project boundaries is that it doesn't deal any internal issues between mechanic and user.

Chapter 2: Software Requirement Specification

2.1 Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the complete Mechanic on the go software system by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs while defining high-level product features. The detailed requirements of the Mechanic on the go are provided in this document.

2.1.1 Purpose

This document provides requirement specification for the system Mechanic on the go. Only technical users and developers can use this document for further development in technology context.

2.1.2 Scope

This app is easy to use and will have a friendly interface so that the user can get the help in case of emergency if they are traveling or having any issue with their car. Either they are in the street or on the GT road they can seek the help by just pressing a button. Time and safety are the most important thing that this project will be giving value to. As if a person is traveling with some passengers or family and face some issues with the car than this app will be his first priority as he will get the help of mechanic or wrecker to carry out his vehicle without leaving his vehicle alone with passengers.

2.1.3 Definitions, acronyms, and abbreviations

- UOL (University of Lahore)
- SDLC: Software Development Life Cycle.
- DFD: Data Flow Diagram.
- CRUD : Create Read Update Delete
- API : Application Program Interface

2.2 Overall description

This document contains the problem statement that the current system is facing which is hampering the growth opportunities of the company. It further contains a list of the stakeholders and users of the proposed solution. It also illustrates the needs and wants of the stakeholders that were identified in the brainstorming exercise as part of the requirements workshop. It further lists and briefly describes the major features and a brief description of each of the proposed system. The following SRS contains the detail product perspective from different stakeholders. It provides the detail

product functions of Mechanic on the go with user characteristics permitted constraints, assumptions and dependencies and requirements subsets.

2.2.1 Product perspective

The mobile application for on-demand vehicle repair and assistance can be considered as an independent and self-contained product, providing a comprehensive solution within its own ecosystem. It enables users to request and receive vehicle repair services directly through the app, without the need for external interfaces or dependencies.

- **System interfaces** The software will interact with the user's mobile device, utilizing its functionalities such as GPS for location tracking, camera for capturing vehicle-related images, and push notifications for communication purposes. The app will also integrate with external services, such as mapping APIs for accurate navigation and payment gateways for secure transactions.
- **User interfaces** The user interface of the mobile include screens for service requests, mechanic selection, service history, cost estimation, payment processing, and communication channels. The layout and design of these interfaces will be intuitive, visually appealing, and easy to navigate.
- **Hardware interfaces** The mobile application itself does not have specific hardware dependencies. However, it requires the user's mobile device to have basic hardware capabilities such as GPS, camera, and internet connectivity for optimal functionality.
- **Software interfaces** The mobile application may utilize software interfaces, such as APIs for mapping services, payment gateways, and communication channels. These interfaces enable seamless integration with external systems and services, enhancing the overall functionality of the application.
- **Communications interfaces** The app will require network connectivity to communicate with external services, such as real-time location tracking and communication between users and mechanics. It will utilize standard network protocols to establish connections and transmit data securely.
- **Memory** As system has to perform operation mobile device should be efficient in utilizing memory while using app or network connectivity. So that system can store all the information.
- **Operations** The normal operations of the application involve users submitting service requests, mechanics accepting and fulfilling those requests, and users making payments for the provided services.
- **Site adaptation requirements.** Before installing this app, Customers and admins need to check the availability of internet in device. Also, mobile phone can be Android or IOS based.

2.2.2 Product functions

ID:	FR_01			
Name:	Create Account			
Description	Input	Output	Requirements	Basic Work Flow
Enter details to create account	First Name, LastName, Email, Password, Phone Number	Account created	Internet Connectivity required	Enter correct information and click submit button System save the record in database

Table 1.0 Functional Requirement Create Account

ID:	FR_02			
Name:	Login			
Description	Input	Output	Requirements	Basic Work Flow
System user will Login to the system.	Enter Email and your password For login	QR code is generated.	Successful integration of all system components	Enter the correct credentials and click LOGIN button.

Table 1.1 Functional Requirement Login

ID:	FR_03			
Name:	Reset Password			
Description	Input	Output	Requirements	Basic Work Flow
Click on Reset Password	New Password	The system will send OTP to user Email for verification	Internet Connectivity required	The OTP send to Email for verificationafter that you will reset your password

Table 1.2 Functional Requirement Reset password

ID:	FR_04			
Name:	Logout			
Description	Input	Output	Requirements	Basic Work Flow
The system will allow the user to logout from their account.	Click on logout button	The system will lead the user to login screen when logout successfully.	Internet Connectivity required	Click on Logout button will lead you to login page with logout details

Table 1.3 Functional Requirement Logout

ID:	FR_05			
Name:	User Profile			
Description	Input	Output	Requirements	Basic Work Flow
You can edit and add more information after login	First Name, Last Name, Email, Password,Phone Number	The system will add user account personal information	Internet Connectivity required	The system will edit add more information of the user.

Table 1.4 Functional User Profile

ID:	FR_06			
Name:	Locating / Get Mechanics			
Description	Input	Output	Requirements	Basic Work Flow
You can locate mechanics	Current location	The system will show the list available mechanics	Internet Connectivity required	Click on Mechanic's image or name will show his profile

Table 1.5 Functional Locating / Get Mechanics

ID:	FR_07			
Name:	View Mechanic's Profiles			
Description	Input	Output	Requirements	Basic Work Flow
When user wants to see profile of **mechanics**	Click on Mechanic's image or name	Mechanic Profile will be displayed like location, rating and top comments etc.	Internet Connectivity required	Click on Mechanic's image or name will show his profile

Table 1.6 Functional View Mechanic's Profile

ID:	FR_08			
Name:	Request Mechanic			
Description	Input	Output	Requirements	Basic Work Flow
User can request for mechanic	click on request button	Mechanic will be assigned to that user and user can see his location	Internet Connectivity required	Clicking on request button to submit his requirement and request for mechanic

Table 1.7 Functional Request Mechanic

ID:	FR_09			
Name:	Request for Wrecker			
Description	Input	Output	Requirements	Basic Work Flow
User can request for wrecker	click on request button	Wrecker will be assigned to that user and user can see his location	Internet Connectivity required	Clicking on request button to submit his requirement and request for wrecker

Table 1.8 Functional Request for Wrecker

ID:	FR_10			
Name:	Request for Petrol			
Description	Input	Output	Requirements	Basic Work Flow
User can request for petrol	Clicking on request petrol button	Nearby filling stations will be displayed	Internet Connectivity required	Clicking on request button will show you a list of nearby filling stations

Table 1.9 Functional Request for Petrol

ID:	FR_11			
Name:	Chatbot System			
Description	Input	Output	Requirements	Basic Work Flow
User can ask question using AI integrated chatbot system	Clicking on chat with us	A new chat screen will open	Internet Connectivity required	Clicking on chat with us button will open a new chat screen in the app

Table 1.10 Functional Chatbot System

ID:	FR_12			
Name:	Feedback mechanism			
Description	Input	Output	Requirements	Basic Work Flow
User will give feedback regarding the service he used	Clicking on give feedback	Feedback section will appear	Internet Connectivity required, should have used a service	Clicking on feedback will open a feedback section where user can give feedback

Table 1.11 Feedback Mechanism

2.2.3 User characteristics

There are mainly three kinds of users of Mechanic on the go.

Customers / Vehicle owner : These users are customer who will access the system for getting services regarding their vehicle repairs. Users have minimal technical knowledge and hence need intuitive navigation aids and simple page layouts.

Mechanics/Wreckers : These users are Repair Experts who provide services to customers.

Admin: Administrator is responsible for maintaining the Mechanic on the go system and will be involved in software fixes, deployment and regular maintenance.

2.2.4 Constraints

Following are the main constraints

- Our Android/IOS application will run on Android version Android 6.0.1 and iOS 7.0 or higher.
- Our application can connect user and nearby mechanics only in a specific radius.
- This application will not deal any internal issues between mechanic and user.
- It requires an active internet connection to function.
- Unregistered vehicle cannot be entertained.

2.2.5 Assumptions and dependencies

The project is supposed to be a user-friendly web and mobile application, which would be able to provide the recommendation of mechanics in your surroundings on the basis of the feedback provided by the other users. The services users will get are:

- The user must have an Android smartphone that can run the app.
- The user's smartphone must have a functioning GPS (Global Positioning System).
- The user must have an internet connection to use our system (application or website).
- Mechanic should meet our standardized protocols.

2.2.6 Apportioning of requirements

This subsection of the SRS should identify requirements that may be delayed until future versions of the system.

2.3 Specific requirements

This section will describe the functional and non-functional requirements of System at a sufficient level of detail for the designers to design a system satisfying the User requirements and testes to verify that the system satisfies the requirements.

2.3.1 Functional Requirement

FR – 01 User Sign Up:

FR 01 – 01	User enters his/her Full name
FR 01 – 02	User enters his/her username
FR 01 – 03	User enters password (must contain alpha-numeric and special characters and should be at least six characters long)
FR 01 – 04	User confirms his/her password
FR 01 – 05	User accepts Terms and conditions checkbox
FR 01 – 06	User clicks on the signup button
FR 01 – 07	Error is displayed in-case of any failed step
FR 01 – 08	User can also signup with their Google account

FR – 02 User Log in:

FR 02 – 01	User enters his/her Full name
FR 02 – 02	User confirms his/her password
FR 02 – 03	User accepts Terms and conditions checkbox
FR 02 – 04	User clicks on the log in button
FR 02 – 05	Error is displayed in-case of any failed step
FR 02 – 06	User can also sign in with their Google account

FR – 03 Forget Password:

FR 03 – 01	User enters his/her Email
FR 03 – 02	User gives the verification code he'd received through email
FR 03 – 03	User enters new password
FR 03 – 04	User confirms the new password
FR 03 – 05	User clicks the reset password button
FR 03 – 06	Error is displayed in-case of any failed step

FR – 04 User Log Out:

FR 04 – 01	User clicks on sign out button
FR 04 – 02	He/ She is redirected to login page

FR – 05 User Profile:

FR 05 – 01	User goes to profile page
FR 05 – 02	He/ She can update his personal information

FR – 06 Locating / Get Mechanics:

FR 06 – 01	User gives the requirements.
FR 06 – 02	User applies the filter
FR 06 – 03	User clicks the Locate button

FR – 07 View Mechanics Profiles:

FR 07 – 01	User can click if he/she wants to see profile of **mechanics**
FR 07 – 02	Mechanic Profile will be displayed like location, rating and top comments etc.
FR 07 – 03	User can give his feedback by clicking on the give feedback button
FR 07 – 04	User will give its rating, comment and the required information
FR 07 – 05	User feedback will be added.
FR 07 – 06	Error is displayed in-case of any failed step

FR – 08 Request for Mechanic:

FR 08 – 01	User can click request button to submit his requirement and request for mechanic
FR 08 – 02	**List of recommended mechanics will be displayed**
FR 08 – 03	User can select a mechanic between there filtered mechanics
FR 08 – 04	User can see their location
FR 08 – 05	Error is displayed in-case of any failed step

FR – 09 Request for Wrecker:

FR 09 – 01	User can click request button to submit his request for Wrecker
FR 09 – 02	**List of nearby wreckers will be displayed**
FR 09 – 03	User can select a mechanic between there filtered wrecker
FR 09 – 04	User can see their location
FR 09 – 05	Error is displayed in-case of any failed step

Request for Petrol:

FR 10 – 01	User can click request button to submit his request for petrol filling
FR 10 – 02	**Nearby filling stations will be displayed**
FR 10 – 03	User can ask different fueling stations contact details
FR 10 – 04	User can see their location
FR 10 – 05	Error is displayed in-case of any failed step

Chatbot System:

FR 10 – 01	User can ask question using our AI integrated chatbot system
FR 10 – 02	**AI chatbot give customer support**
FR 10 – 03	User can ask their queries about our app
FR 10 – 04	User can seek guidance
FR 10 – 05	Our AI bot can display all the information related to user's query

Feedback mechanism*:*

FR 10 – 01	We collect user's feedback
FR 10 – 02	**User can give rating and feedback**
FR 10 – 03	User can see all the feedback also rating of our application

2.3.2 Non-functional Requirements

- **Usability**
  System must present a formal documentation tutorial in the form of pop-ups for newbies. The system should only focus on what is primary / important for the user and don't focus on what would put the user in any type of mess or confusion. Web Application should be information orientated while native app should be task orientated.

- **Reliability**
  System should only respond to correct and valid feedbacks and ignore the fake ones. System can also block the fake user permanently

- **Performance**
  Performance of the system must be good. System's quick response will outline its performance irresponsive of the mass interacting with him.

- **Portability**
  Application should be designed and developed to be easily portable across multiple mobile platforms (iOS, Android) and devices. It should be compatible with different operating system versions and screen sizes, allowing users to access the app seamlessly regardless of their device. Additionally, data portability features should be provided to enable users to transfer their data when switching devices or platforms.

- **Maintainability**
  The system should be maintained properly. Whenever there is a problem, it must be fixed as soon as possible. Moreover, the system must be timely updated according to the needs of the user

- **License Agreement**
  System is not an open-source system, and all rights are reserved with the developer and in order to deploy it at any organization, seeking permission from developers

Chapter 3: Use Case Analysis

3.0.1 User Sign up Use Case Diagram:

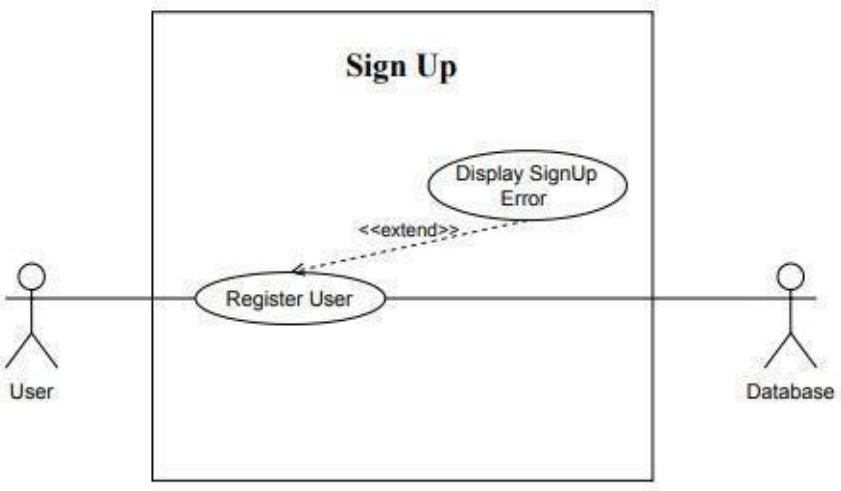


Figure 1 Usecase Diagram User Signu

Usecase diagram detail

Use Case ID	**UC_01**	
Use Case Name	**Sign Up**	
Description	Sign up allows users to register into Mechanic on the go.	
Primary Actor	Customer/ Mechanic/ Wrecker	
Secondary Actor	Database	
Pre-Condition	Customer / Mechanic / Wrecker must open the Mechanic on the go app to register his/herprofile.	
Post-Condition	Customer / Mechanic / Wrecker Information has been saved into the System	
Basic Flow	Actor Action	System Action
	1. This use case starts whenuser try to register their information. 2. User must enter information which includes First Name, Last Name , Phone Number, Email Address,Home Address, City, Password to register for an account. 3. System responds by notifying the user that their registration requesthas been sent for approval. This use case ends here	What would system do according to the information
Alternate Flow	User already has an account.	

Table 2 Usecase Sign Up

3.0.2 Login Use Case Diagram:

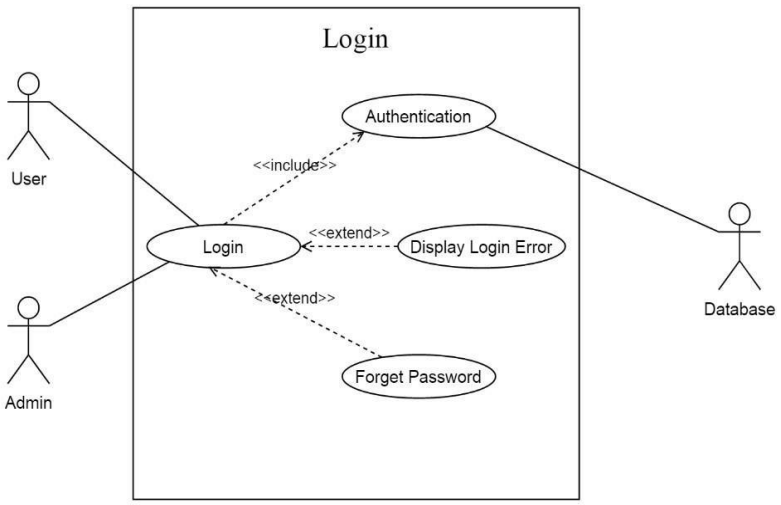


Figure 2 Usecase Diagram Login

Use Case ID	**UC_02**	
Use Case Name	**Login**	
Description	Login allows users to login to Mechanic on the go.	
Primary Actor	Customer / Mechanic / Wrecker	
Secondary Actor	Database	
Pre-Condition	Admin/User should've their account registered beforehand.	
Post-Condition	Admin/User has been Login into the application, successfully.	
Basic Flow	Actor Action	System Action
	User will open the login screen and provide the credentials and log into the system.	System will retrieve database to authenticate the credentials of the system and let user to login if credentials are right.
Alternate Flow	None	

Table 3Usecase Login

3.0.3 Logout Use Case Diagram:

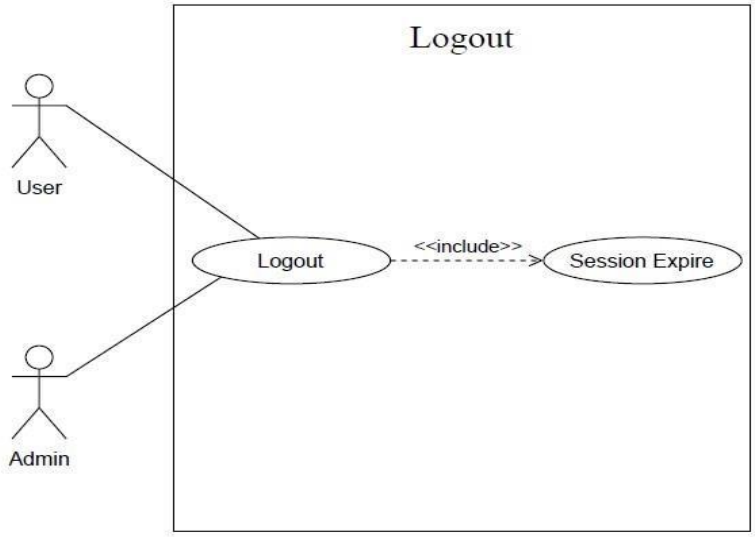


Figure 3Usecase Diagram Logout

Use Case ID	**UC_03**	
Use Case Name	**Logout**	
Description	The users can logout from the application.	
Primary Actor	Customer / Mechanic / Wrecker	
Secondary Actor	Database	
Pre-Condition	User should be signed in his/her account.	
Post-Condition	User has been successfully logged out from the application.	
Basic Flow	Actor Action	System Action
	User will press logout button	System will logout the user from itand user will no longer be able to make changes to system database.
Alternate Flow	The user has already logged out.	

Table 4 Usecase Login

3.0.4 Locate Mechanic Use Case Diagram:

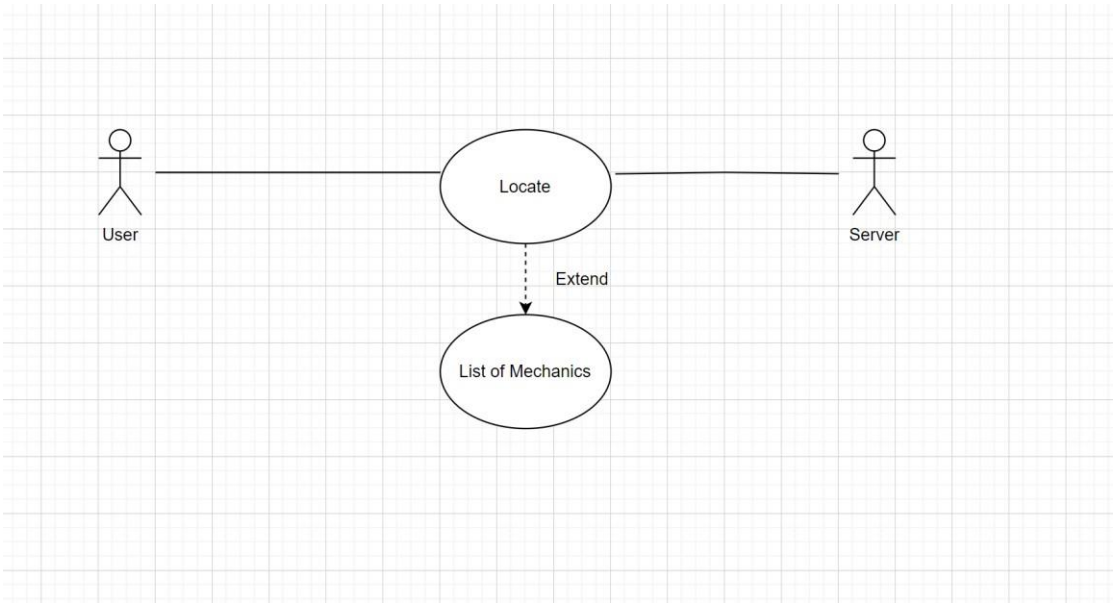


Figure 4 Usecase Diagram Locate Mechanic

Use Case ID	**UC_04**	
Use Case Name	**Locate Mechanic**	
Description	User can locate nearby mechanic using his/her location	
Primary Actor	Customer	
Secondary Actor	Database, Google Map, API	
Pre-Condition	User has been logged in.	
Post-Condition	User has carried out its desire goal	
Basic Flow	Actor Action	System Action
	1. User opens the app. 2. User logs in 3. User chooses either of the task he wants to perform 4. User performs the task by following the valid set of operations	
Alternate Flow	In case of forget password, user will be provided with the reset link by the app.	

Table 5 Usecase Locate Mechanuc

3.0.5 Request for Mechanic Use Case Diagram:

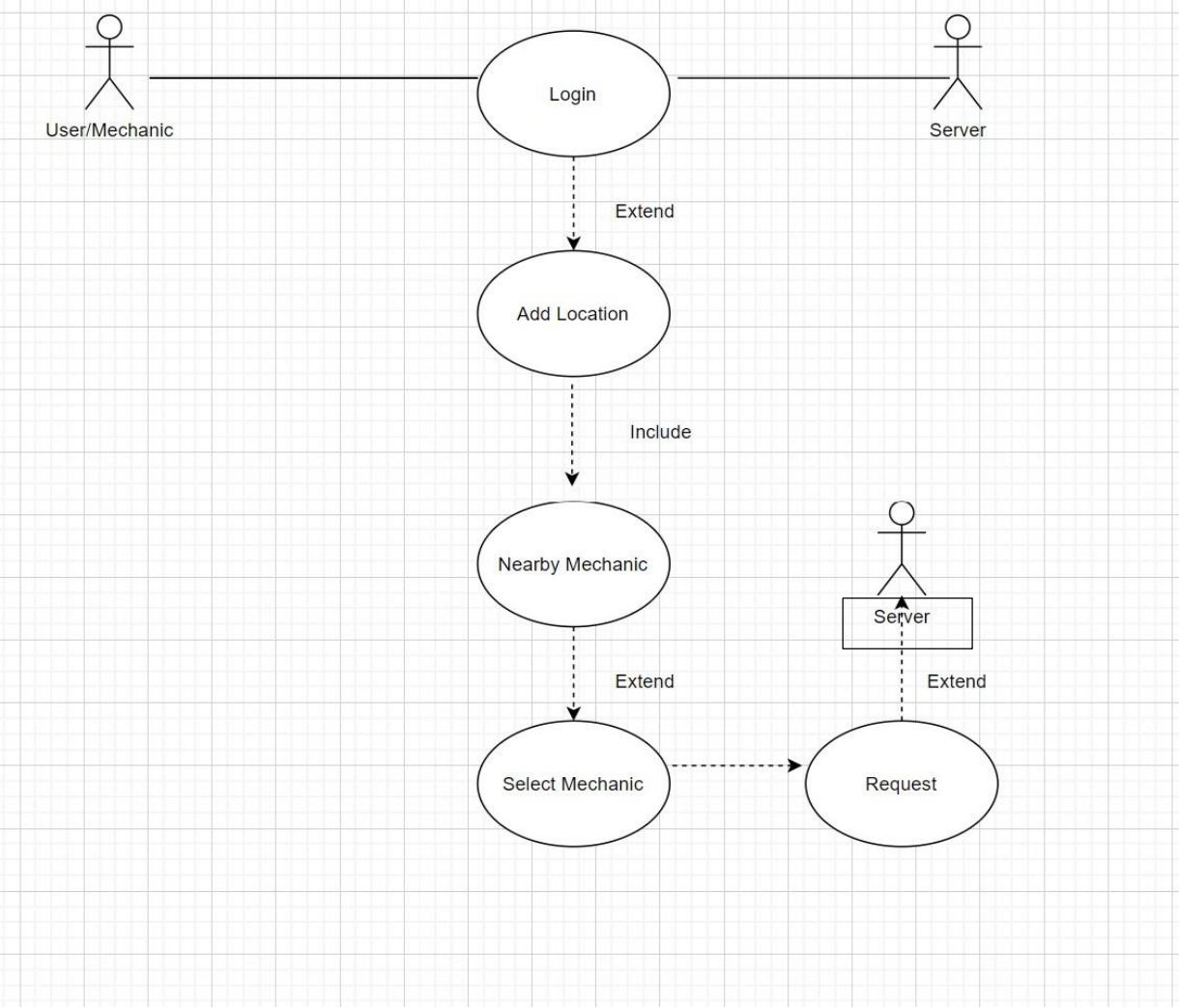


Figure 5 Usecase Diagram Request for Mechanic

Use Case ID	**UC_05**
Use Case Name	**Request for Mechanic**
Description	After entering user's location, a user can now request for a mechanic.
Primary Actor	Customer
Secondary Actor	Database, Mechanic
Pre-Condition	User has logged in.
Post-Condition	User has carried out its desire goal
Basic Flow	Actor Action
	1. User opens the app. 2. User logs in 3. User chooses either of the task he wants to perform i.e. Request for Mechanic. 4. User performs the task by following the valid set of operations
Alternate Flow	None

Table 6 Usecase Request for Mechanuc

3.0.6 Request for Wrecker

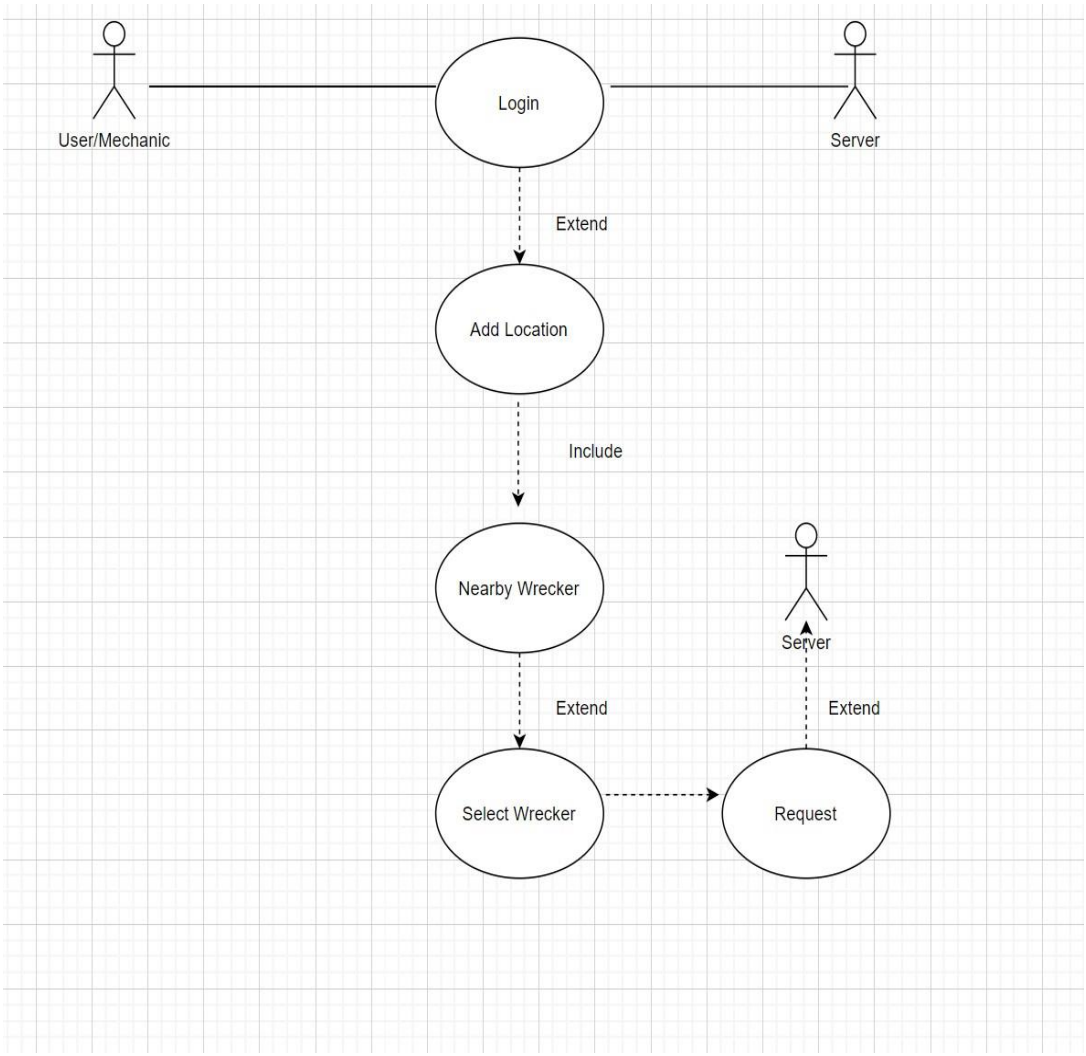


Figure 6 Usecase Diagram Request for Wrecker

Use Case ID	**UC_06**
Use Case Name	**Request for Wrecker**
Description	After entering user's location, a user can now request for a Wrecker.
Primary Actor	Customer
Secondary Actor	Database, Mechanic
Pre-Condition	Location must be valid.
Post-Condition	Nearest Wrecker location have been shown to user.
Basic Flow	Actor Action
	1. User opens the app. 2. User logs in 3. User chooses either of the task he wants to perform i.e. Request for Wrecker 4. User performs the task
Alternate Flow	None

Table 7 Usecase Request for Wrecker

3.0.7 Request for Petrol

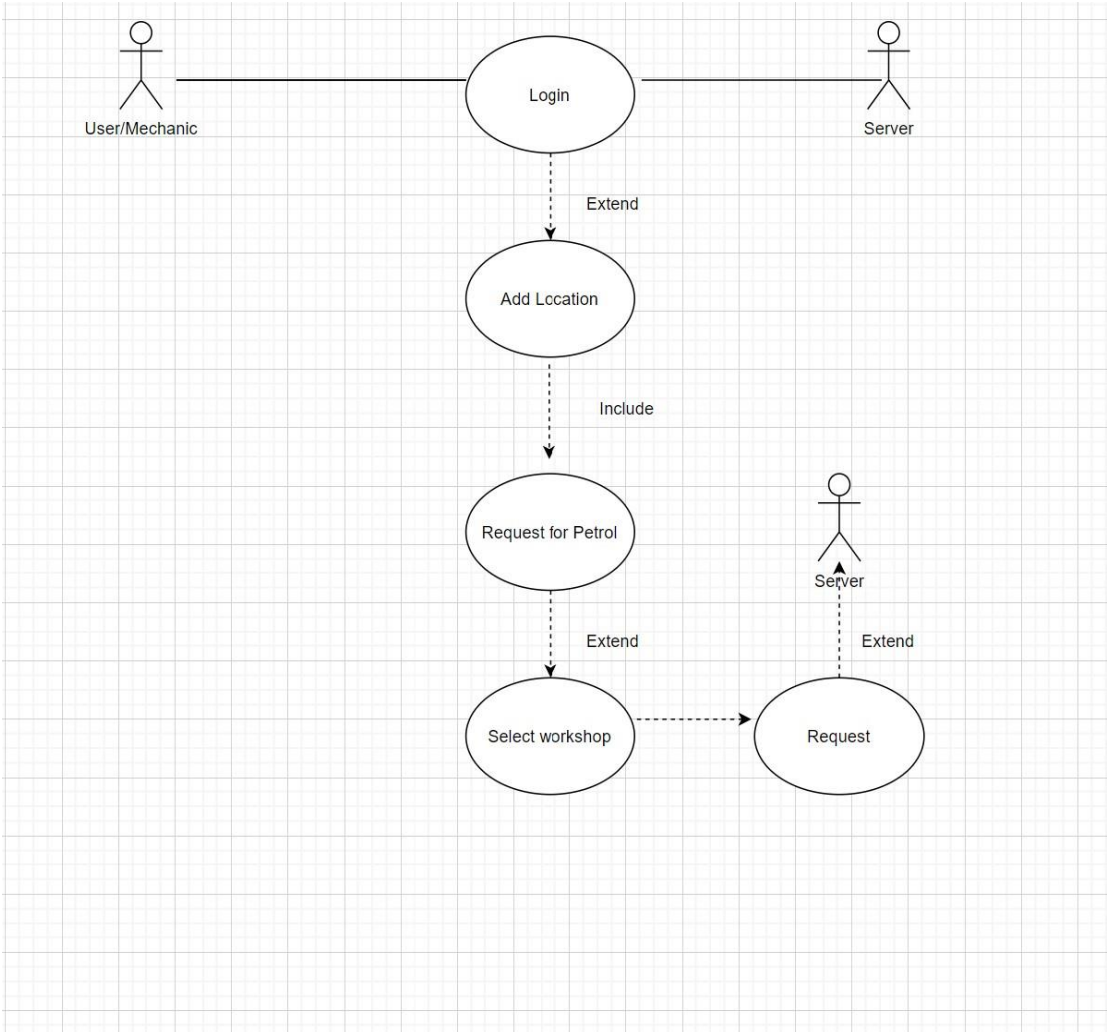


Figure 6 Usecase Diagram Request for Wrecker

Use Case ID	**UC_07**
Use Case Name	**Request for Petrol**
Description	After entering user's location, a user can now request for a Petrol.
Primary Actor	Customer
Secondary Actor	Database, Mechanic
Pre-Condition	Location must be valid.
Post-Condition	Nearest filling stations and workshops with availability of petrol shown.
Basic Flow	Actor Action
	1. User opens the app. 2. User logs in 3. User chooses either of the task he wants to perform i.e. Request for Petrol 4. User performs the task
Alternate Flow	None

Table 8 Usecase Request for Petrol

3.0.8 Chatbot system Use Case Diagram:

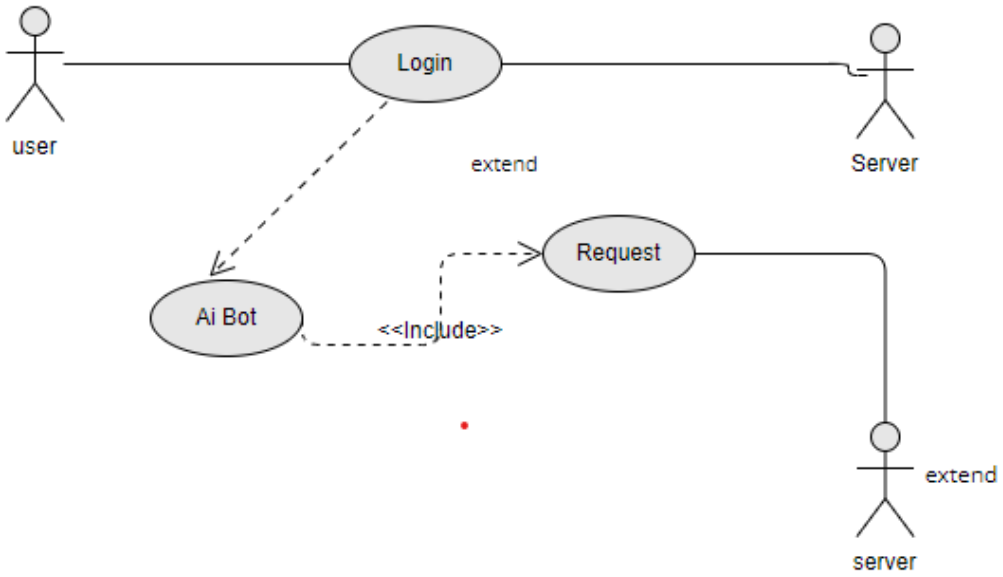


Figure 7 Usecase Diagram Chatbot system

Use Case ID	**UC_08**
Use Case Name	**Chatbot system**
Description	Ask Question related to any problem that you are facing with our application
Primary Actor	Customer, Mechanic
Secondary Actor	None
Pre-Condition	Question must be related to our application.
Post-Condition	All the information related to our application is displayed
Basic Flow	Actor Action
	1. User opens the app. 2. User logs in 3. User chooses either of the task he wants to perform i.e. Ask Question 4. User performs the task
Alternate Flow	None

Table 9 Usecase Chatbot System

3.0.9 Feedback Mechanism Use Case Diagram:

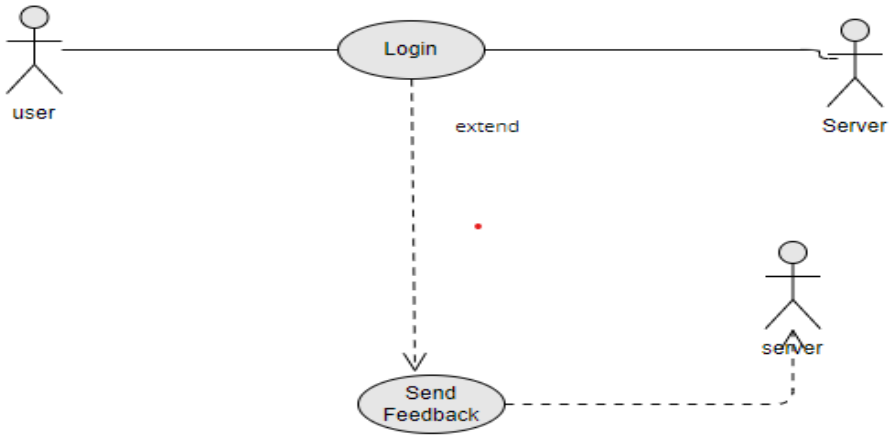


Figure 8 Usecase Diagram Feedback Mechanism

Use Case ID	**UC_09**
Use Case Name	**Feedback Mechanism**
Description	User can give feedback and rating to Mechanic, Wreckers
Primary Actor	Customer
Secondary Actor	None
Pre-Condition	User must be logged in
Post-Condition	Feedback given successfully.
Basic Flow	Actor Action
	1. User opens the app. 2. User logs in 3. User chooses either of the task he wants to perform i.e. Give Feedback 4. User performs the task
Alternate Flow	None

Table 10 Usecase Feedback Mechanism

Chapter 4: Design

In this section, we provide the design analysis of our modules including the following designs

1. Architecture Diagram
2. ERD with data dictionary
3. Data Flow diagram
4. Class Diagram
5. Activity Diagram
6. Sequence Diagram
7. Collaboration Diagram
8. State Transition Diagram
9. Component Diagram
10. Deployment Diagram

4.1 Architecture Diagram

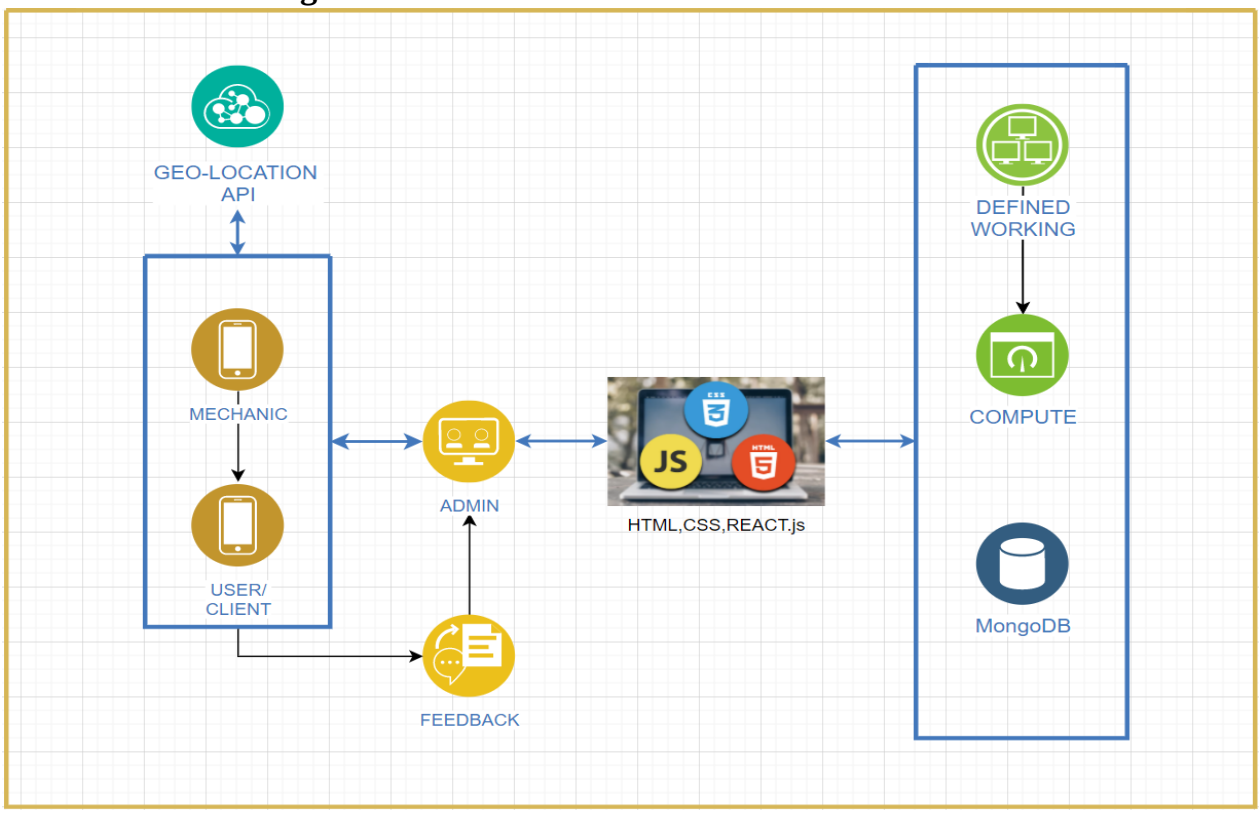


Figure 2 Architecture Diageam

4.2 ERD with data dictionary

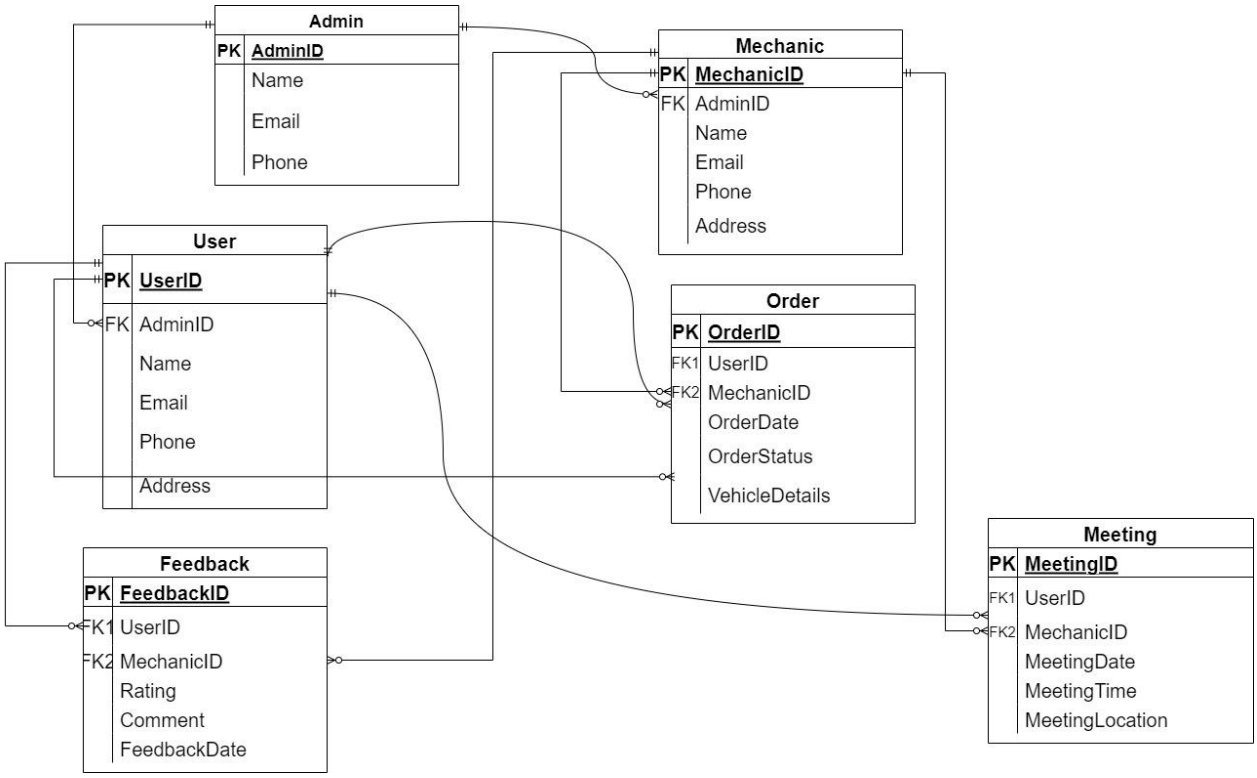


Figure 3 ERD

4.3 Data Flow diagram

Data flow diagram includes two levels

4.3.1 The level 0

The flow of information inside the system is defined in this level

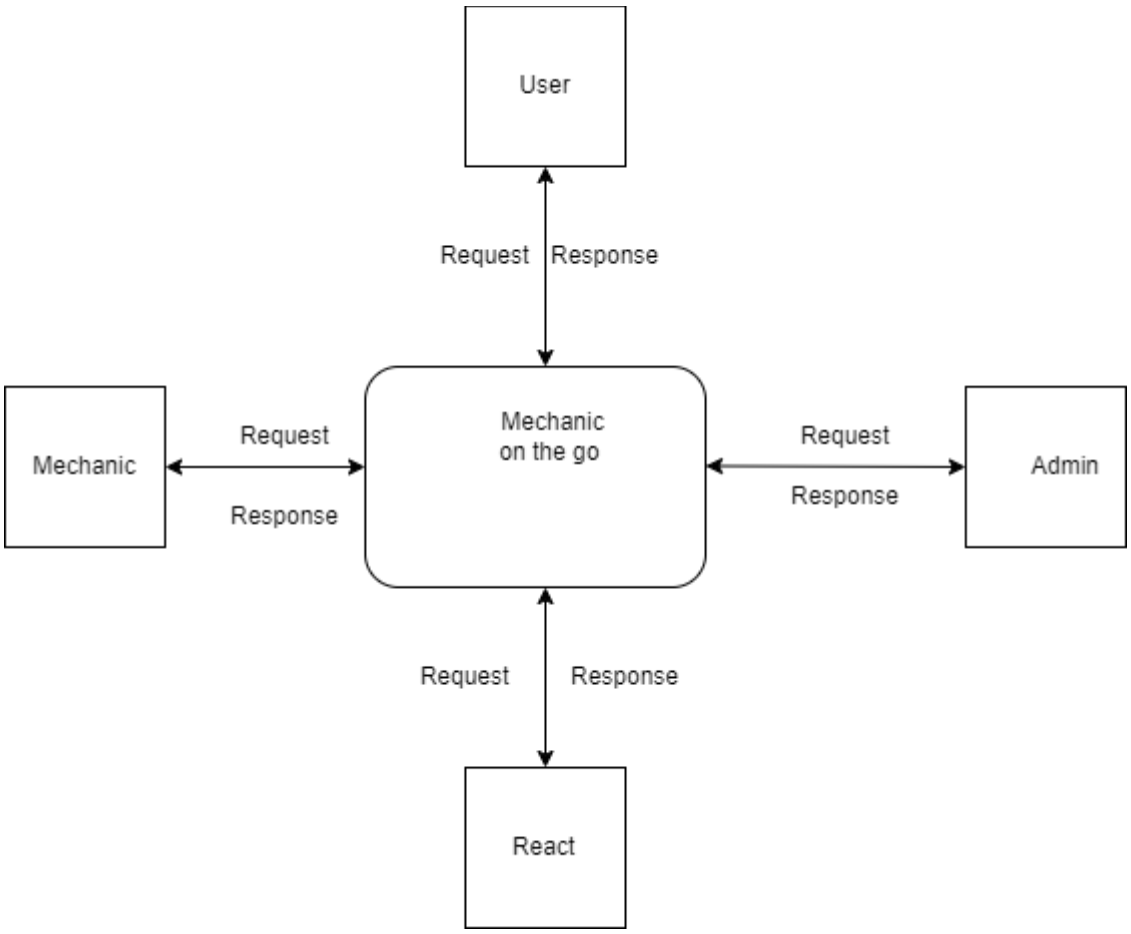


Figure 4 Level 0 DFD

4.3.2 The level 1

The flow of information outside the system is defined in this level

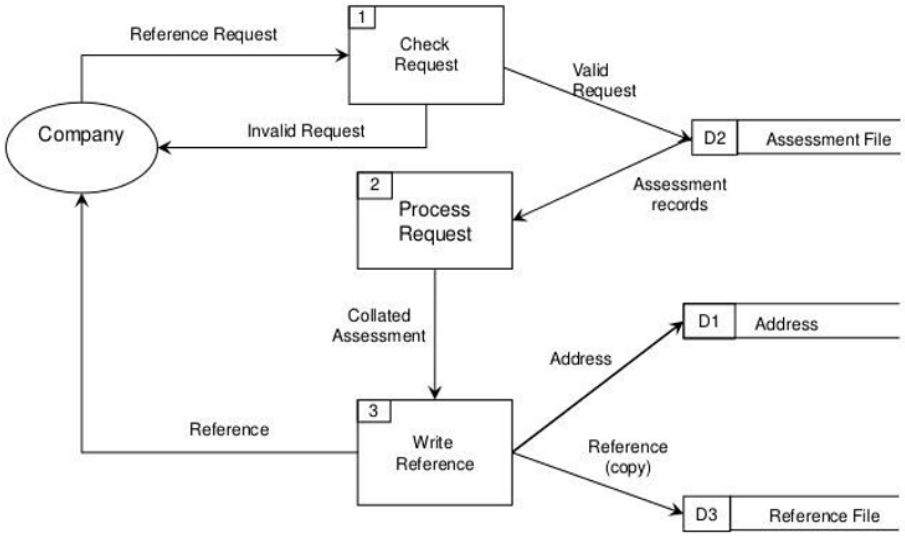


Figure 5 Level 1 DFD

4.4 Class Diagram

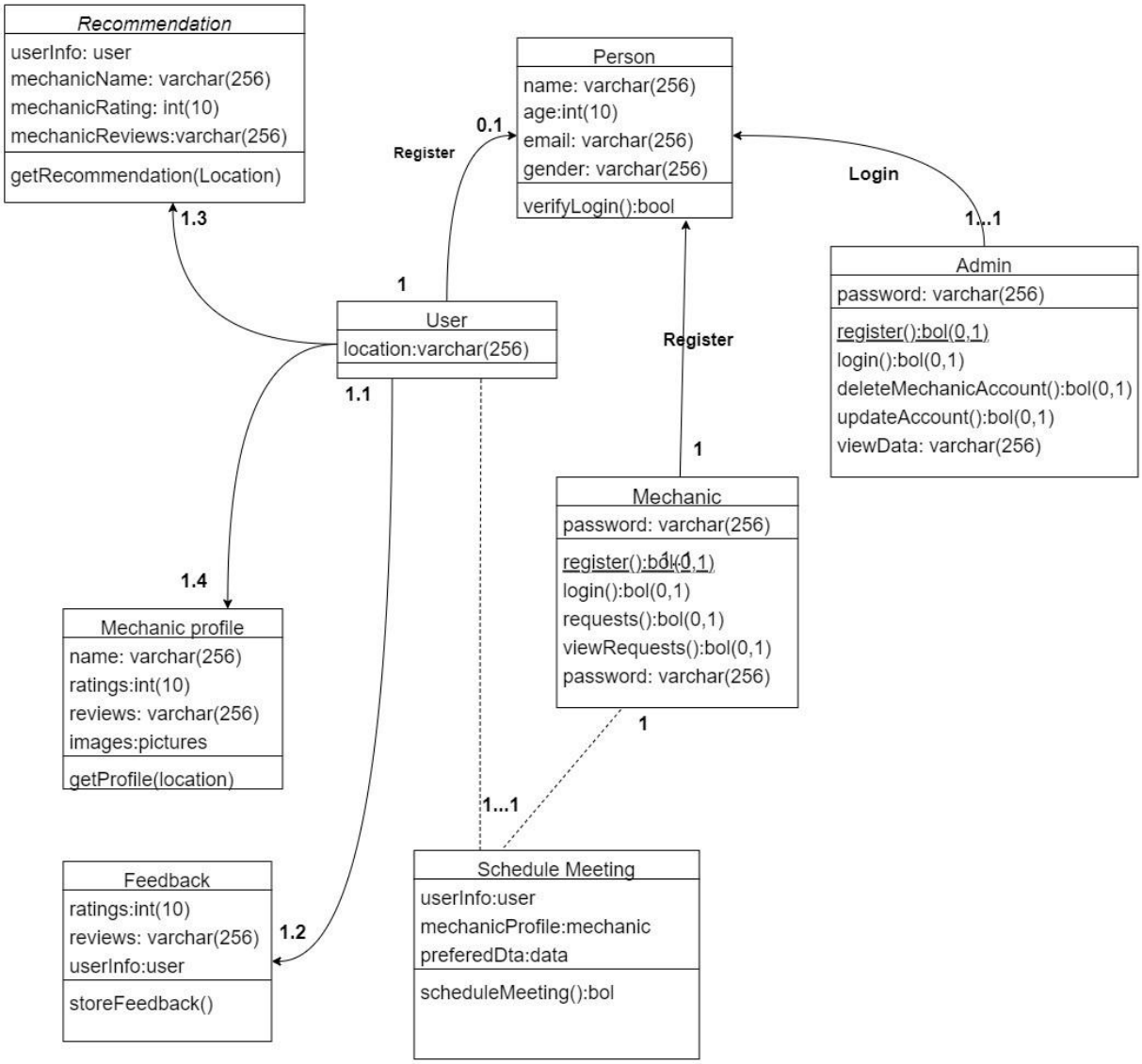


Figure 6 Class Diagram

4.5 Activity Diagram

Login

The flow of activities of login shows in Figure. The users and admin will enter their credentials, the System will match them with the database and the appropriate function will run.

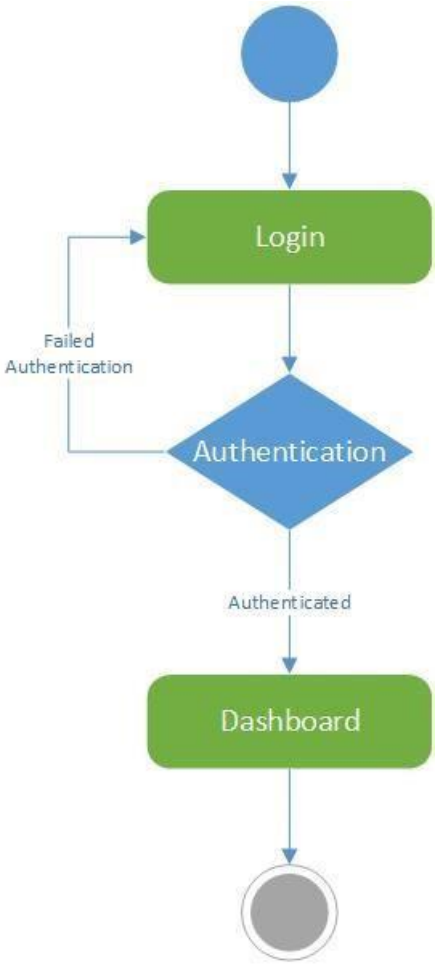


Figure 7 Activity Login

Register User

The flow of activities of Register user show in Figure.

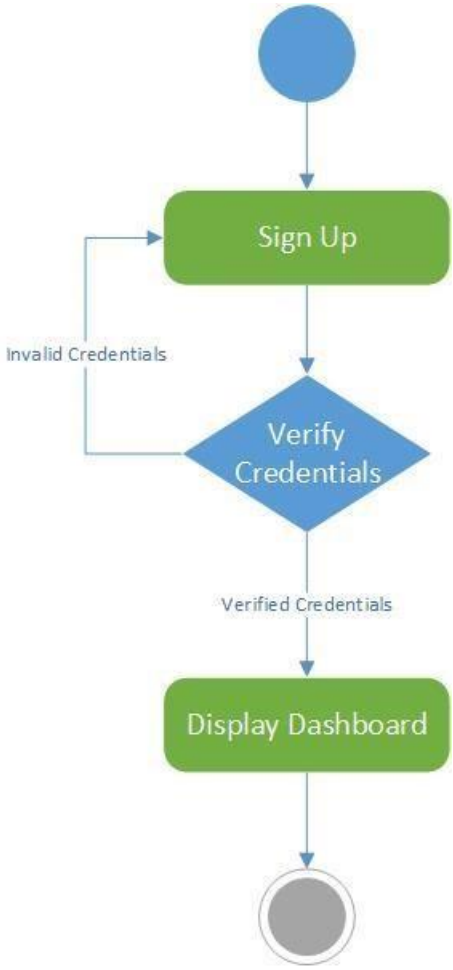


Figure 8 Activity Register User

System Recommendations

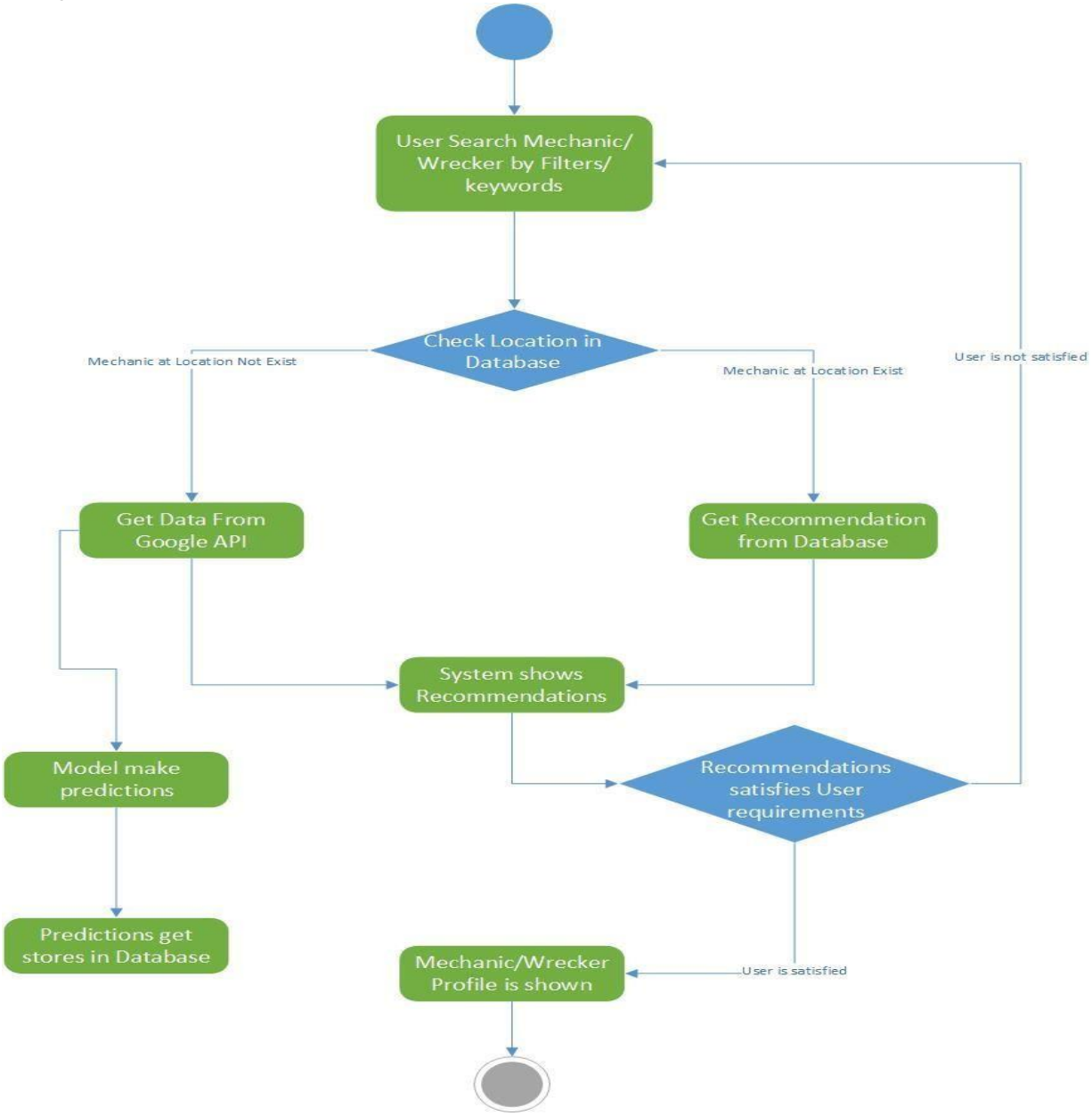


Figure 9 Activity System Recommendations

Forgot Password

Activity Diagram for Forgot Password

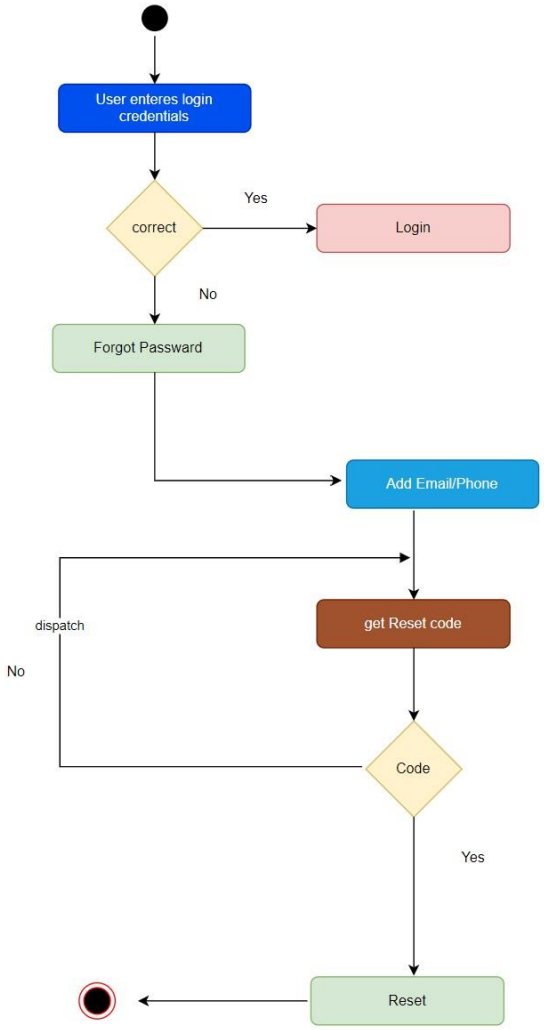


Figure 10: Forgot password

Locate Mechanic

Activity Diagram for Locate Mechanic

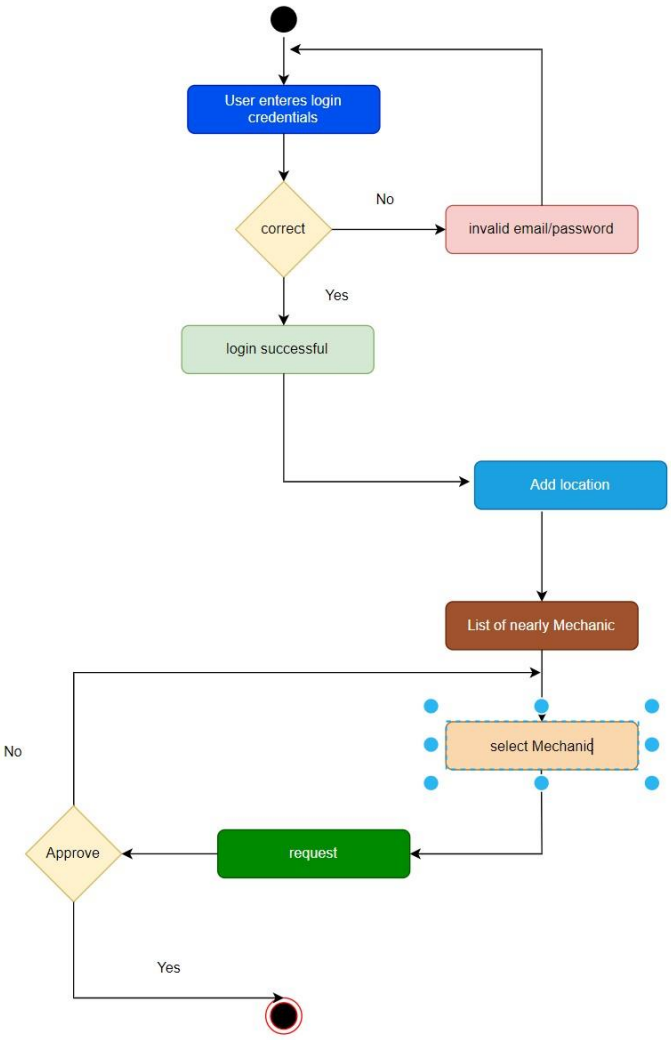


Figure 11 Locate Mechanic

Request Mechanic

Activity Diagram for Request Mechanic

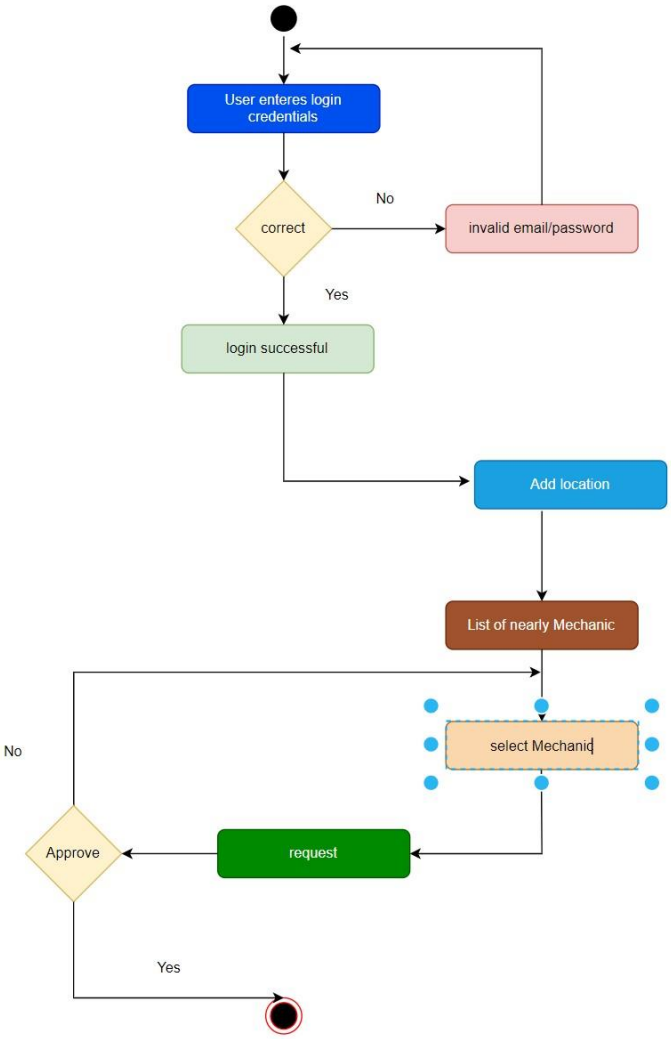


Figure 12 Request Mechanic

Request Wrecker

Activity Diagram for Request wrecker

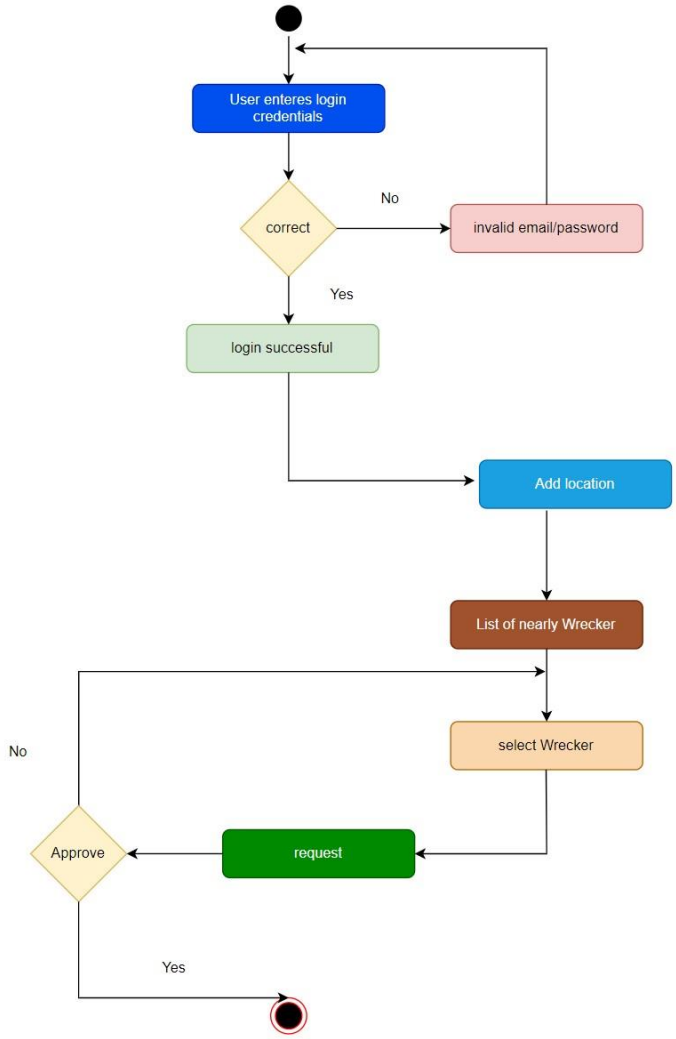


Figure 13 Request Wrecker

Request petrol

Activity Diagram for Request Petrol

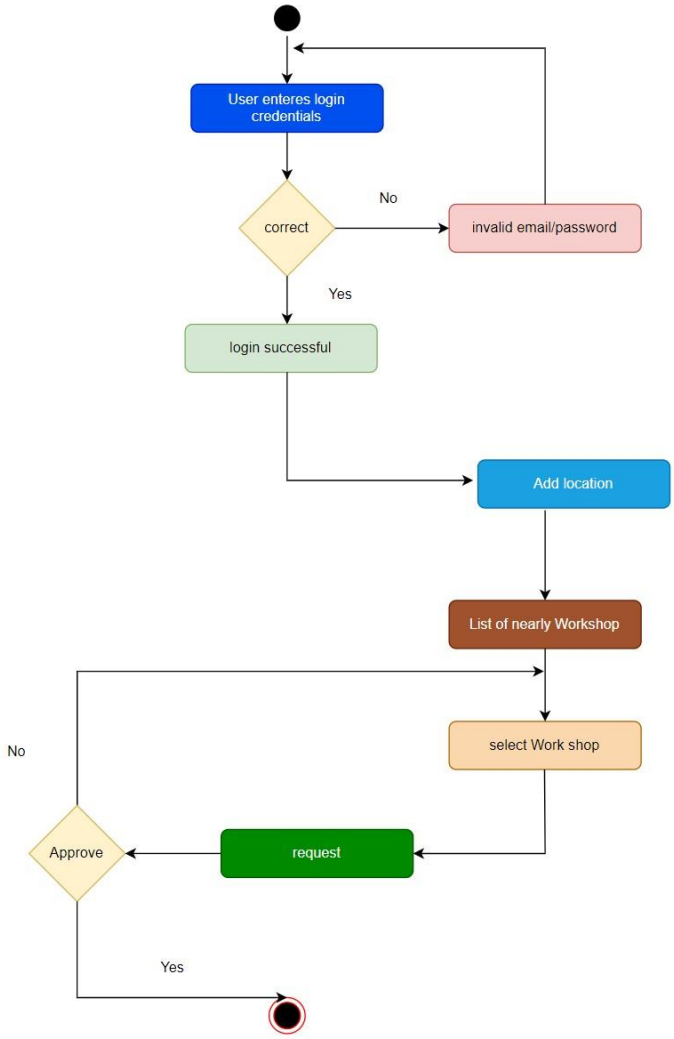


Figure 14 Request Petrol

User's and Mechanic's Profile

Activity Diagram for User's and Mechanic's Profile

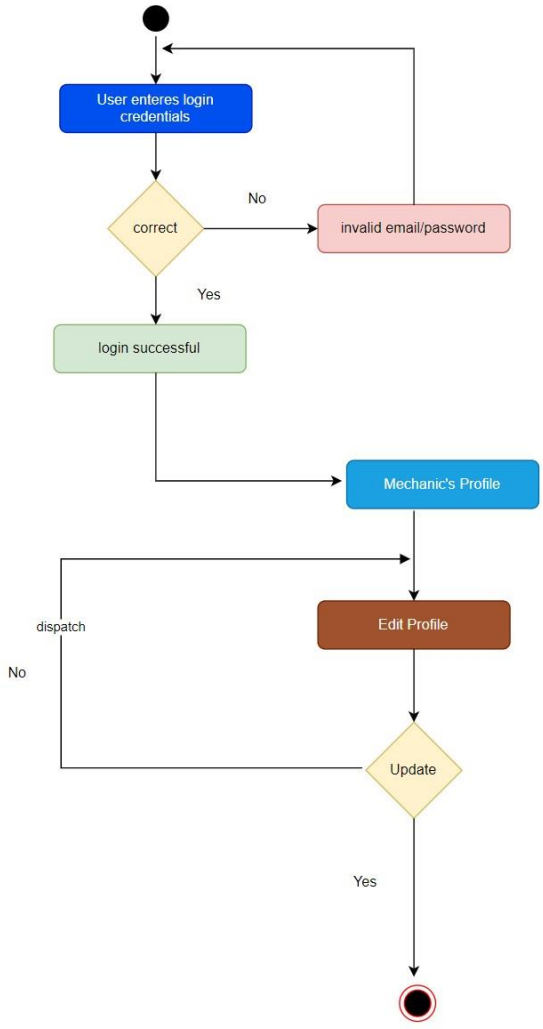


Figure 15 User's Profile

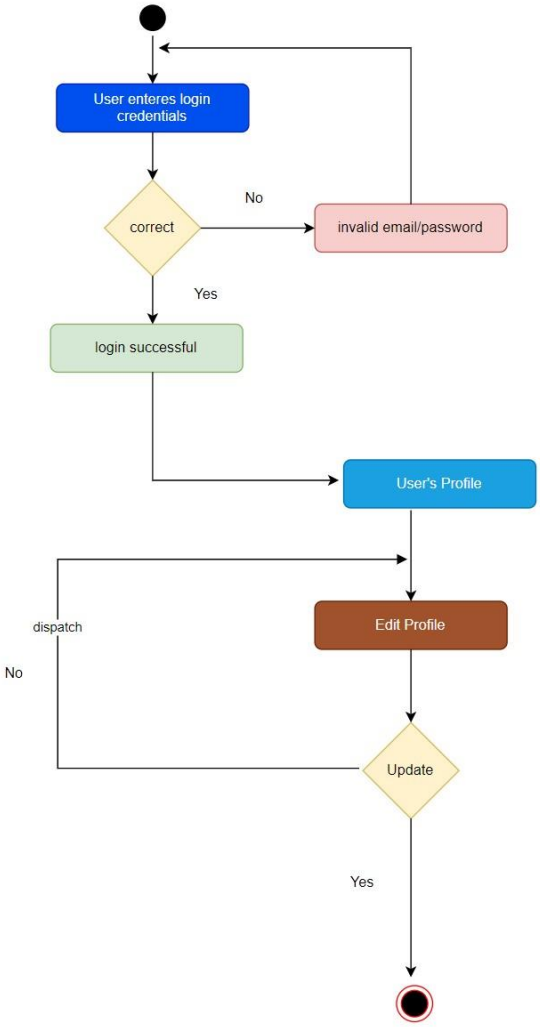


Figure 16 Mechanic's Profile

ChatBot

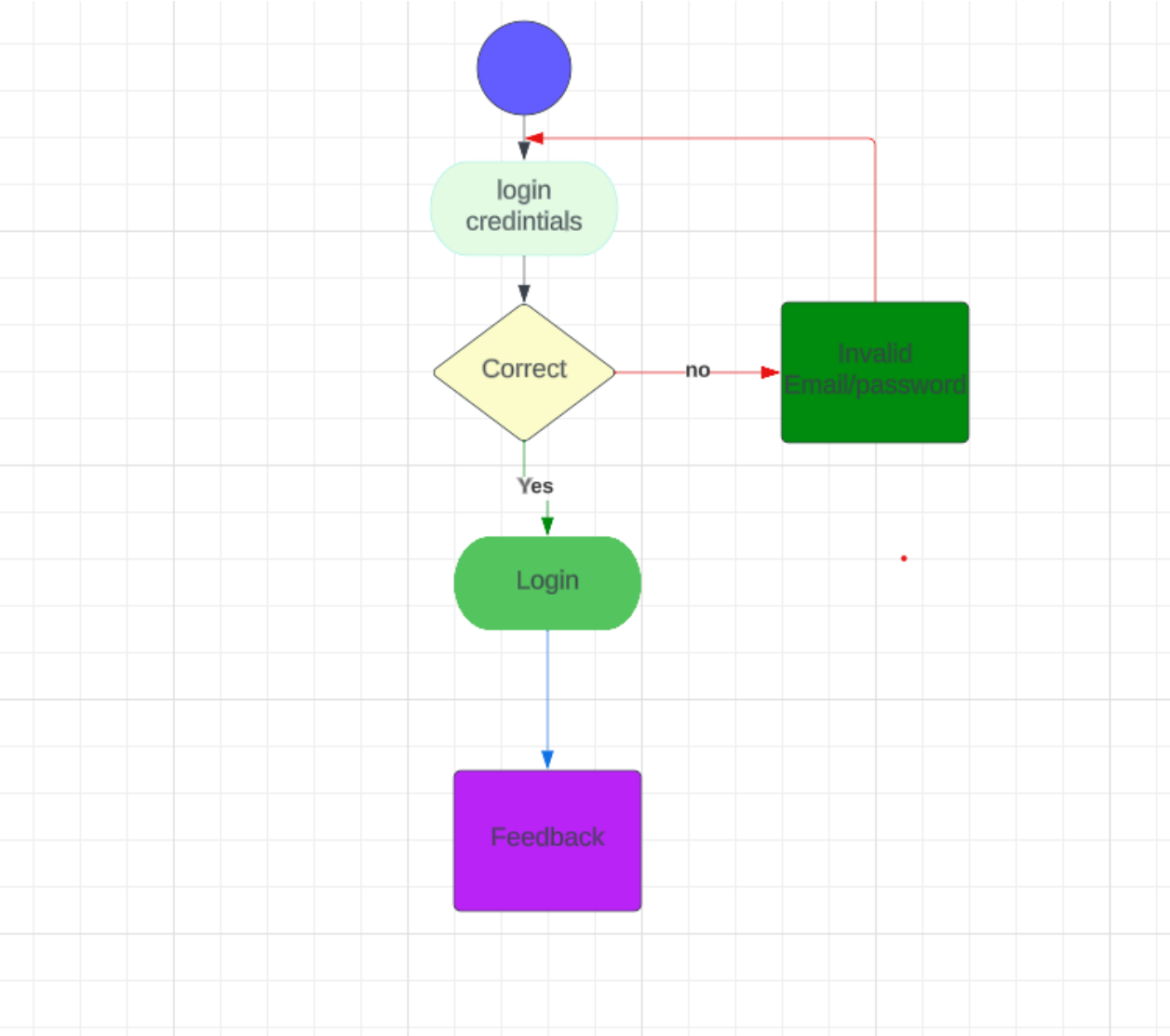


Figure 17 ChatBot

Feedback

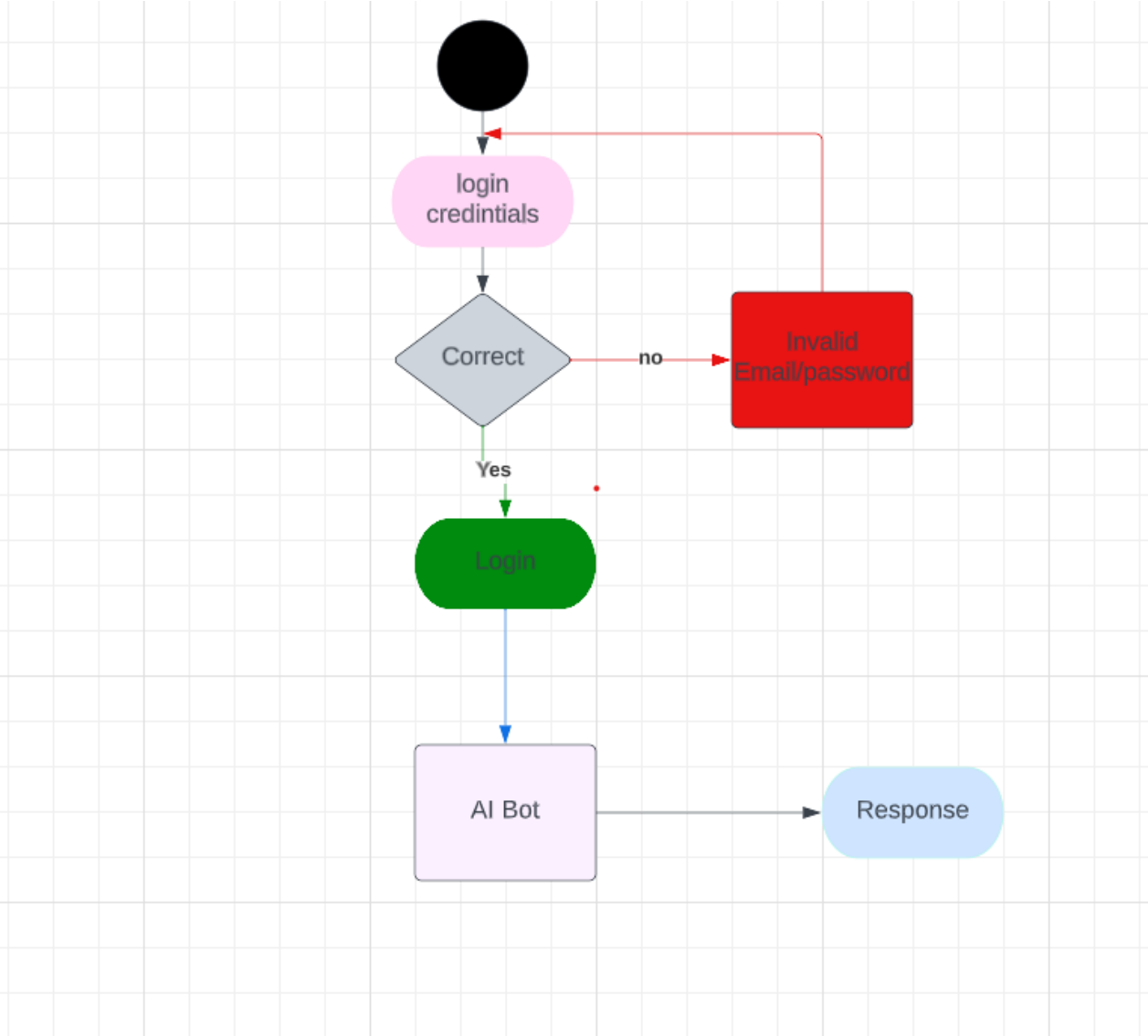


Figure 18 Feedback

4.6 Sequence Diagram

Admin Sequence

Figure shows sequence diagram of admin; the figure describes each module's participation in sequence for admin.

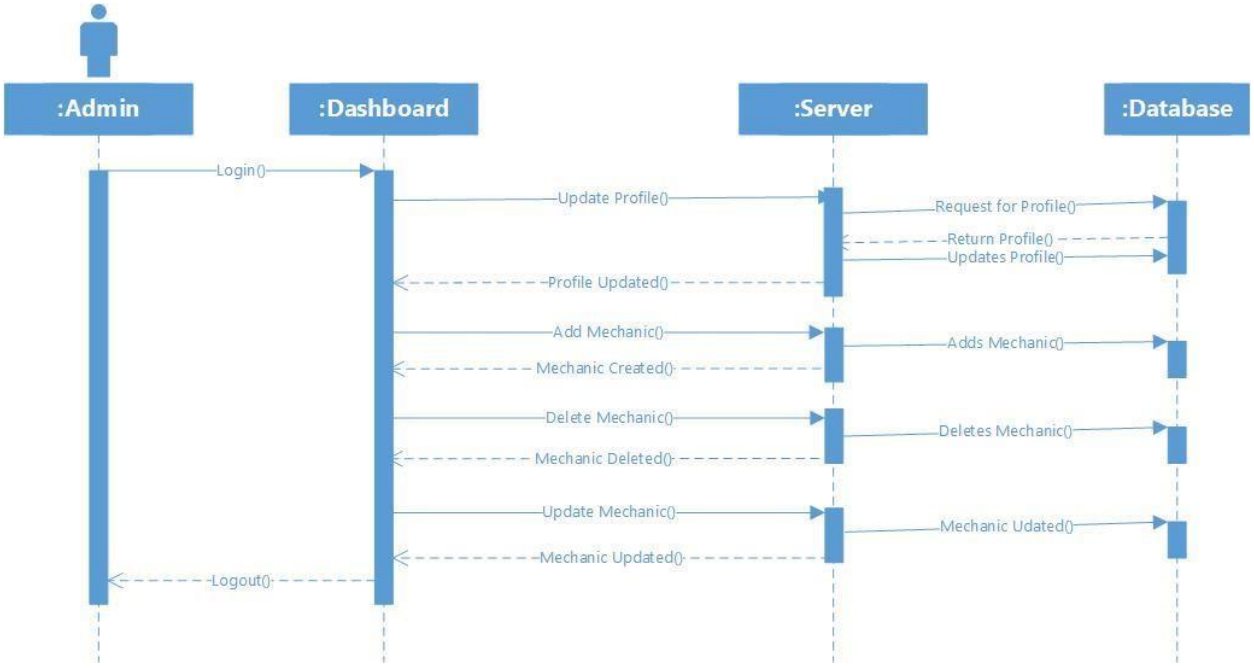


Figure 19 Sequence Diagram Admin

Login Sequence

Figure shows sequence diagram of login; the figure describes each module's participation in sequence for login.

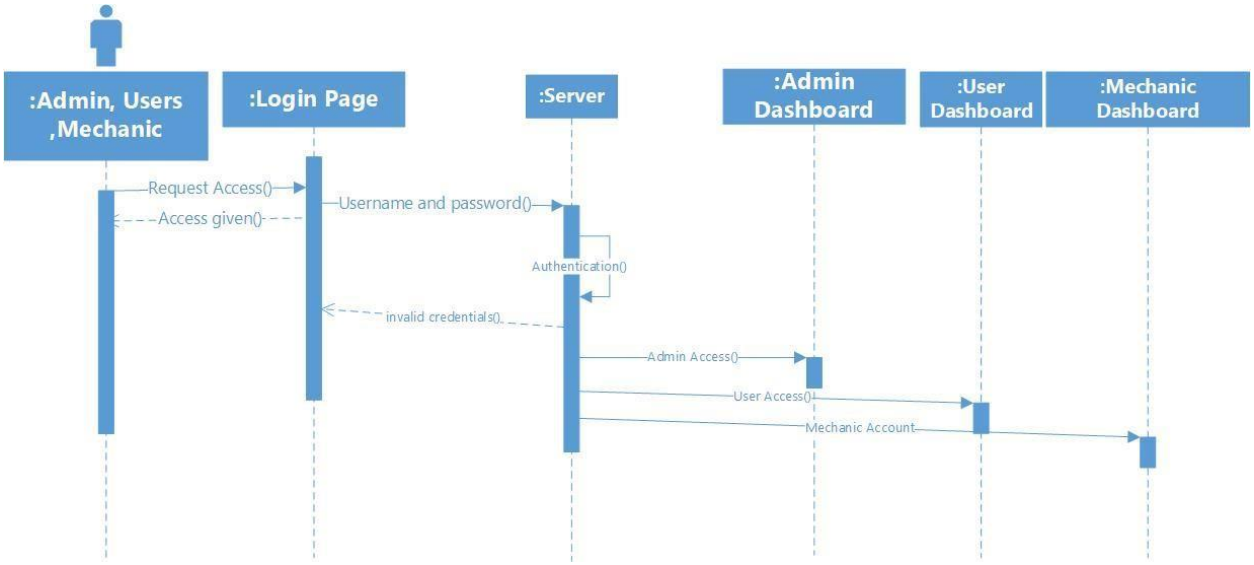


Figure 20 Sequence Diagram Login

System Recommendations

Figure shows the whole interaction of user with the application to perform different task. The figure contains the sequence and the activation of each module against the task.

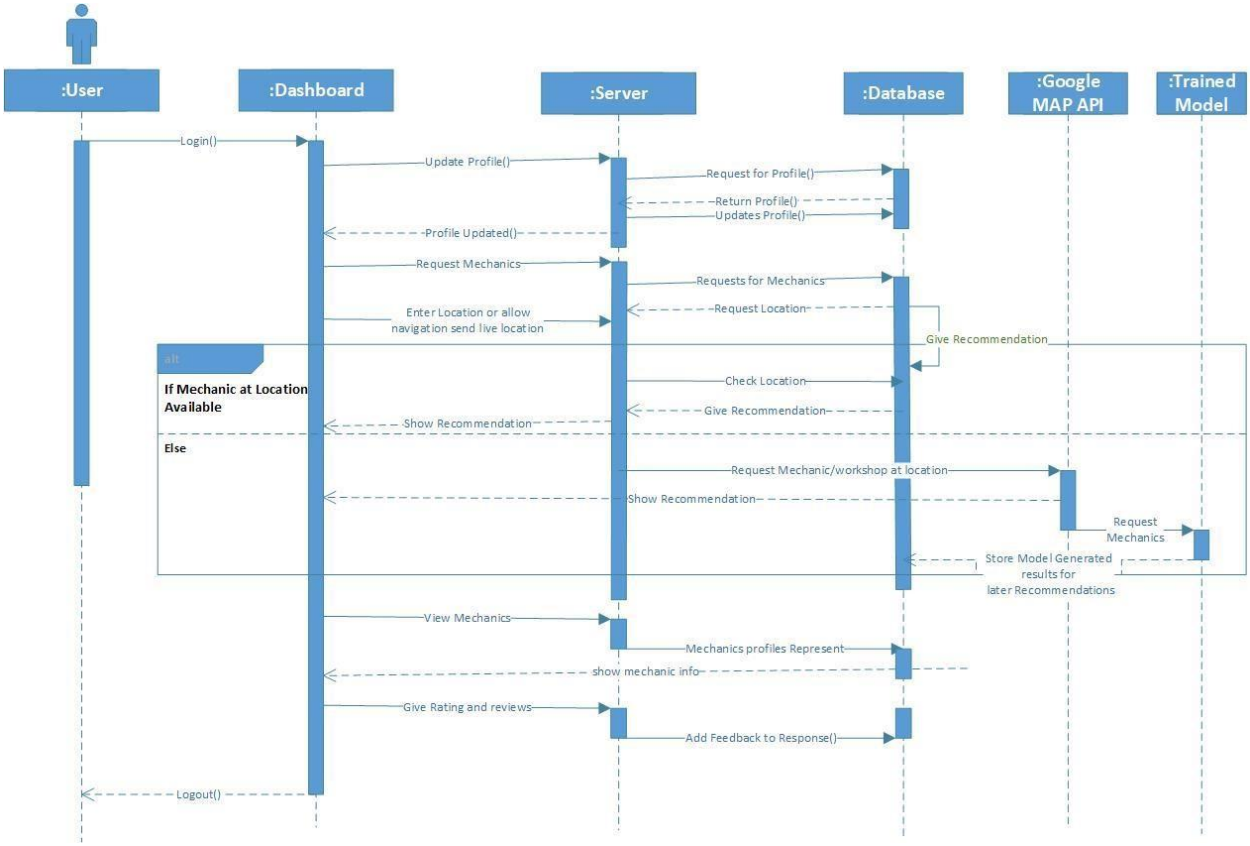


Figure 21 Sequence Diagram System Recommendations

Locate Mechanic

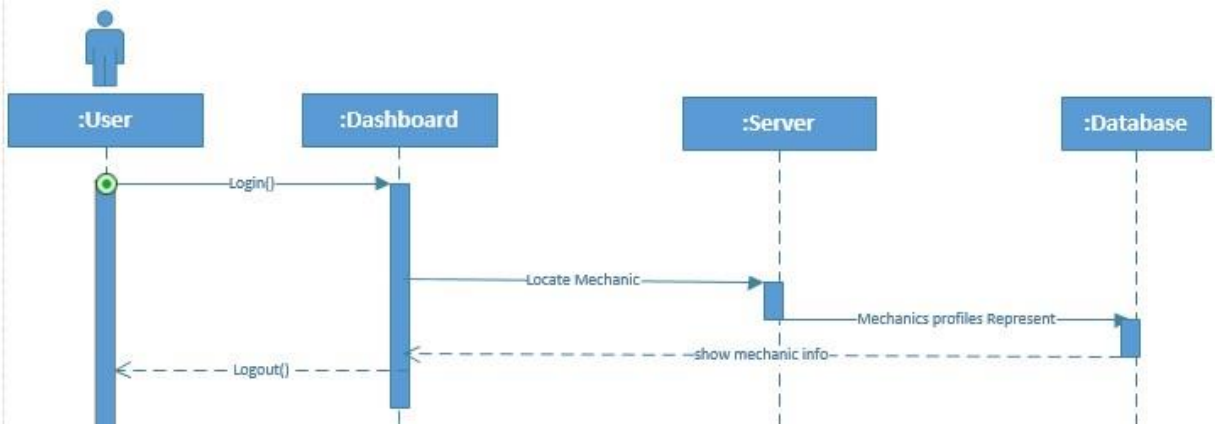


Figure 22Locate Mechanic

Request Mechanic

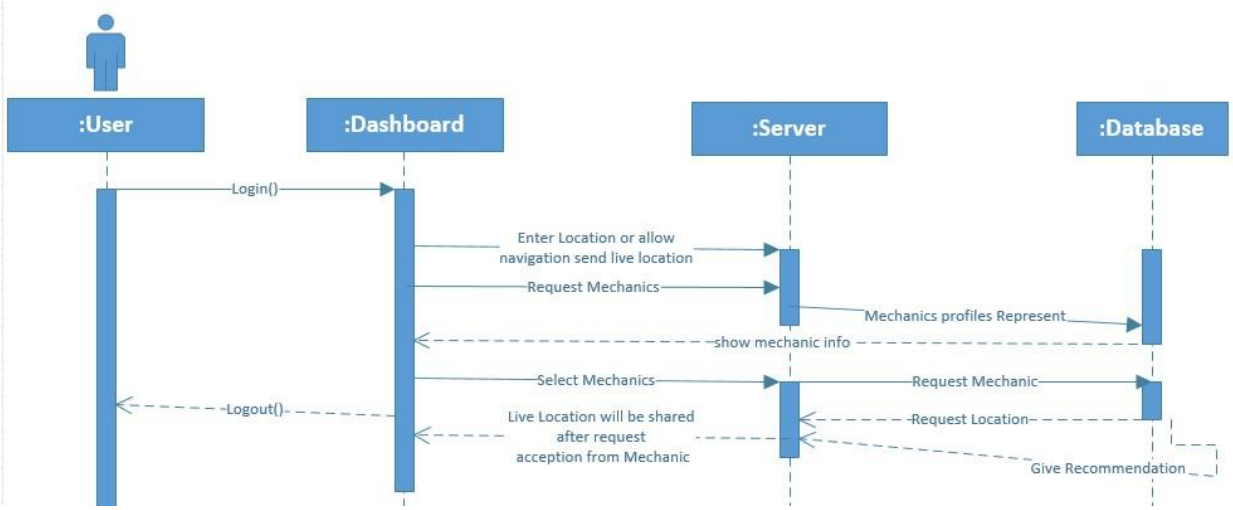


Figure 23 Request Mechanic

Locate Wrecker

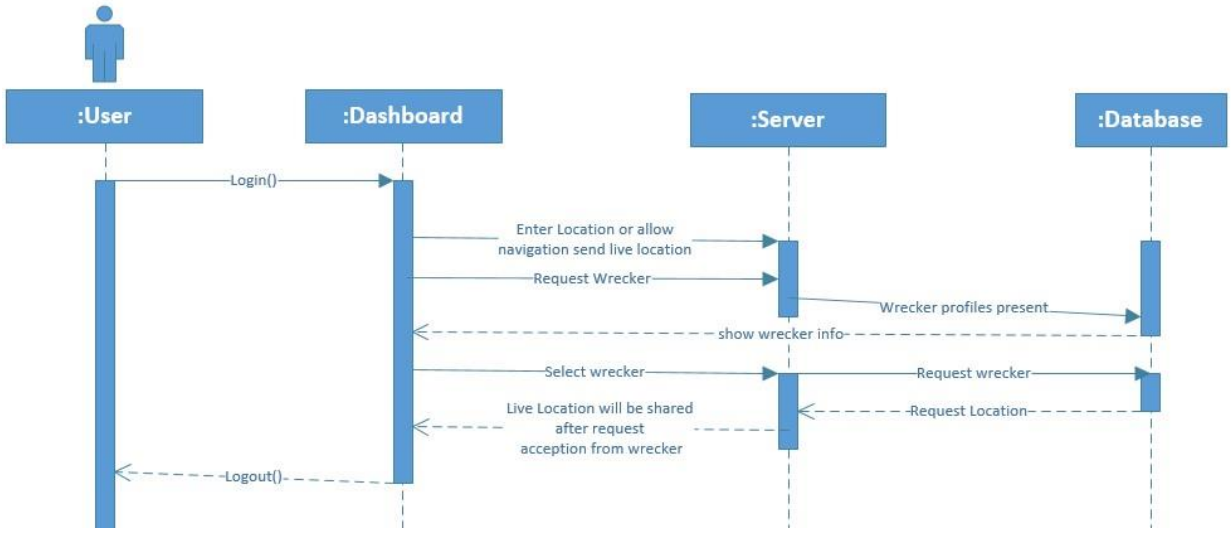


Figure 24 Locate wrecker

Request Petrol

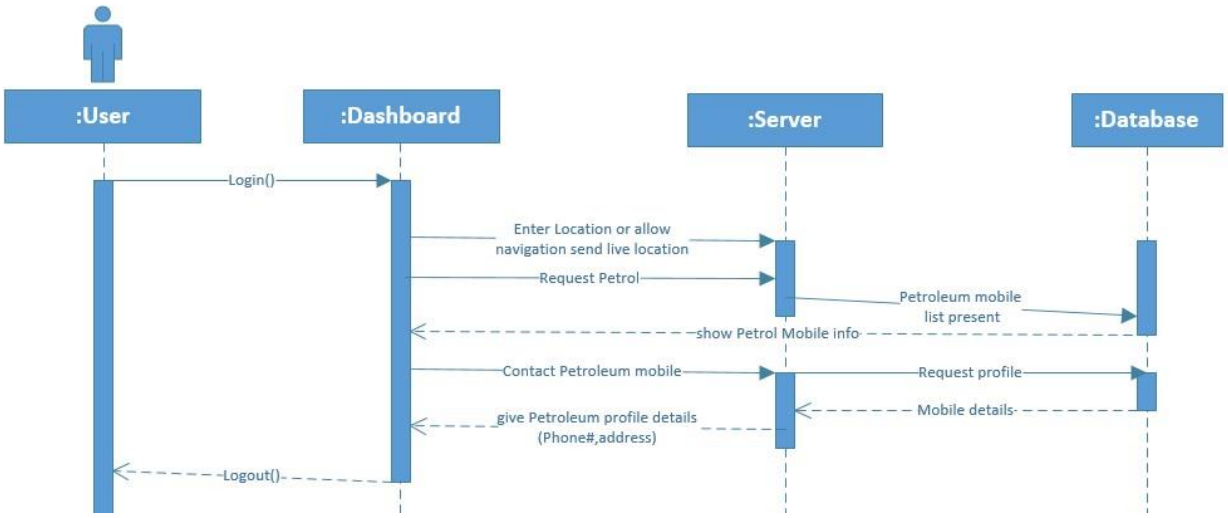


Figure 25 Request Petrol

View, Update User Profile

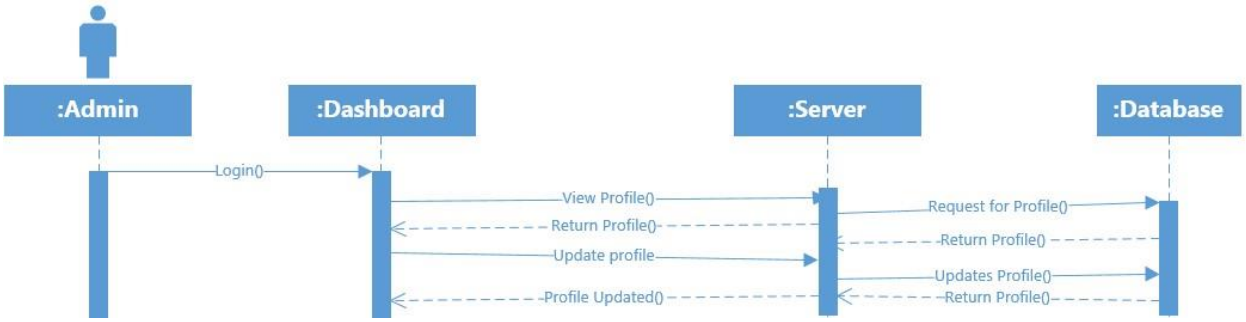


Figure 26 Vuew, Update User profile

Feedback

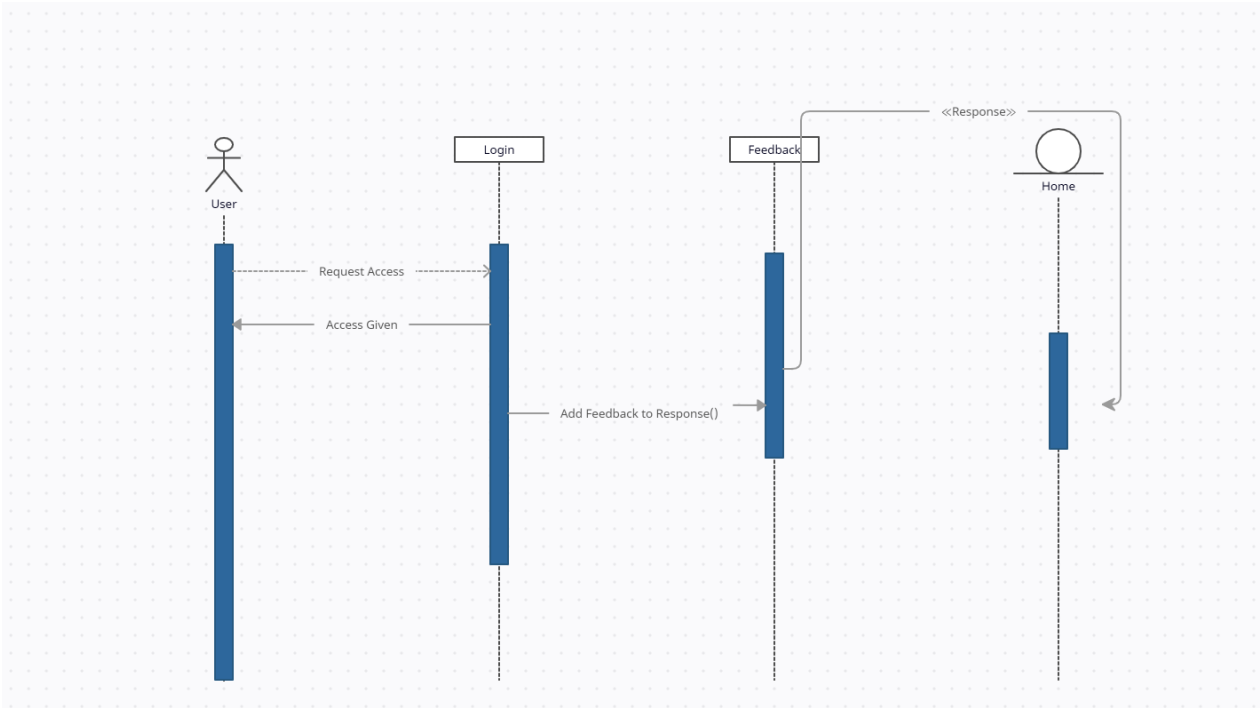


Figure 27 Feedback

Chatbot

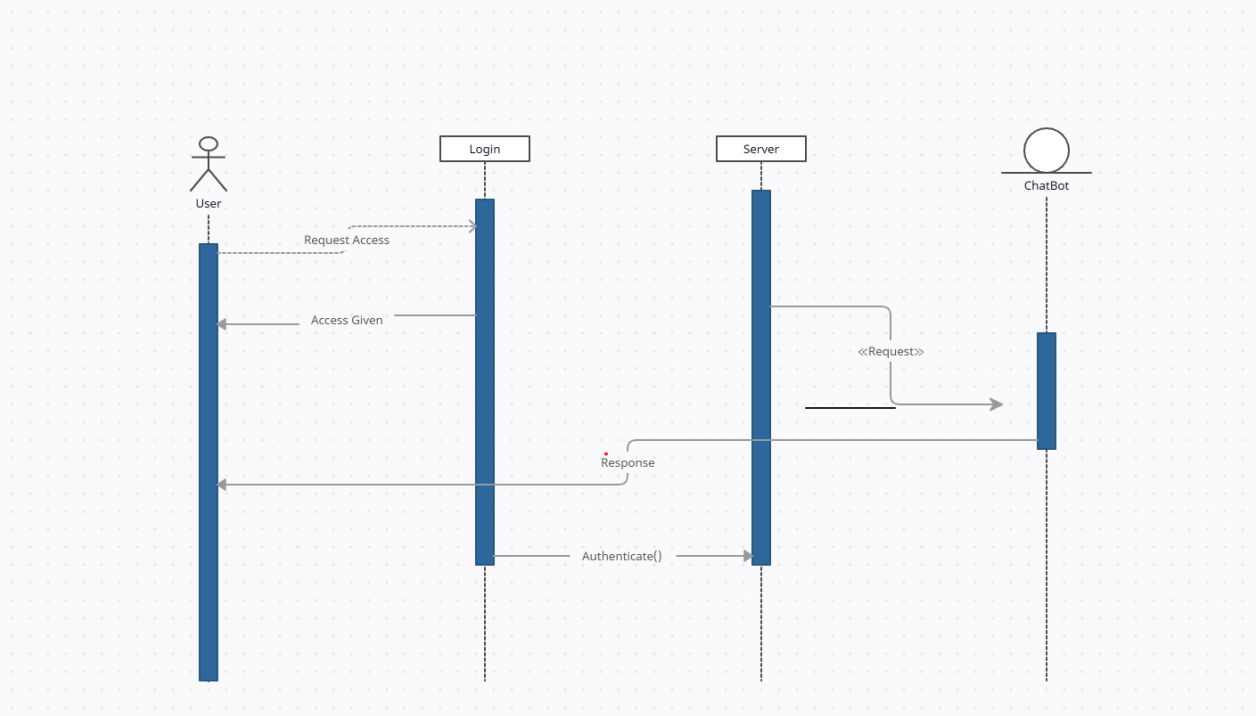


Figure 27 Sequence Diagram Chatbot

4.7 Collaboration Diagram

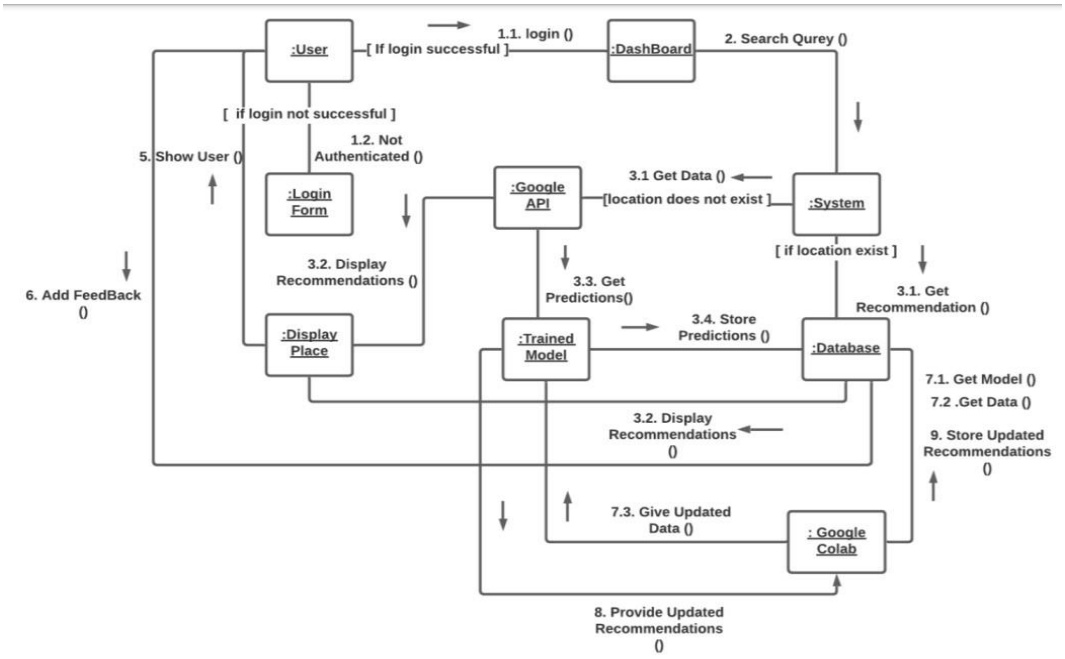
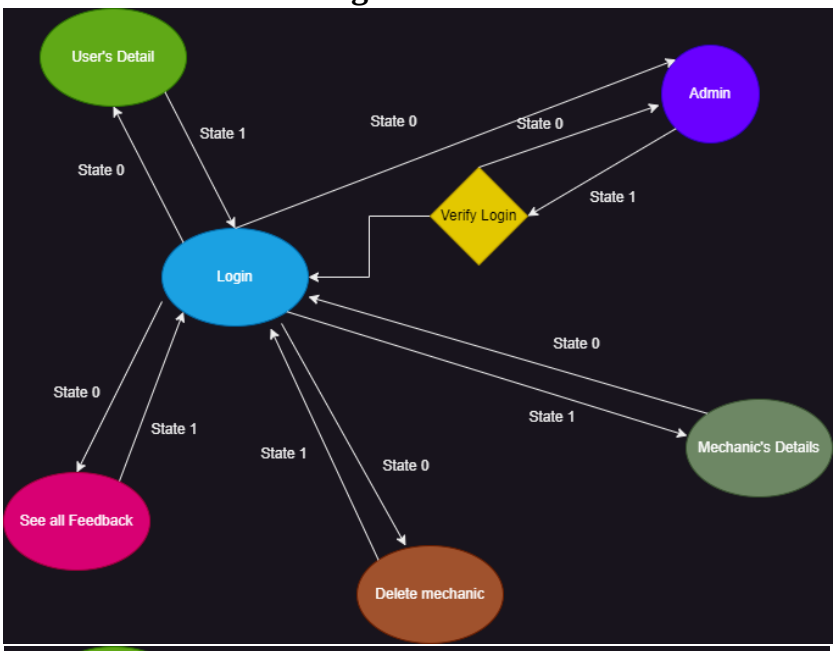
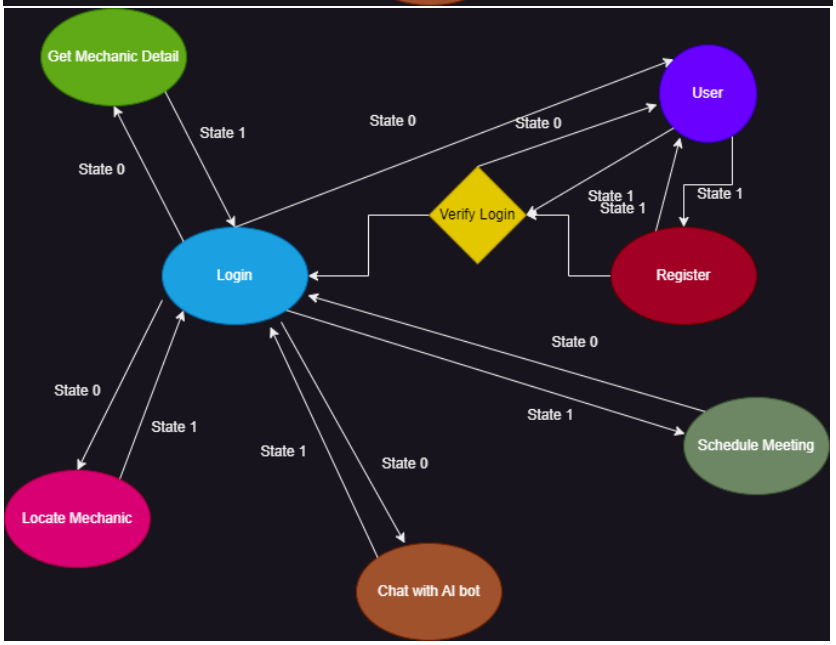


Figure 28 Collaboration Diagram

4.8 State Transition Diagram





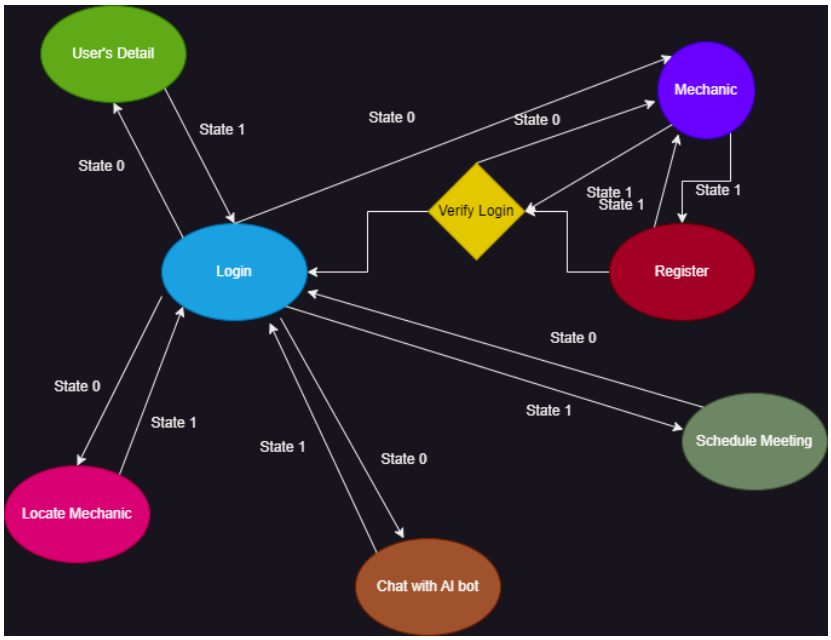
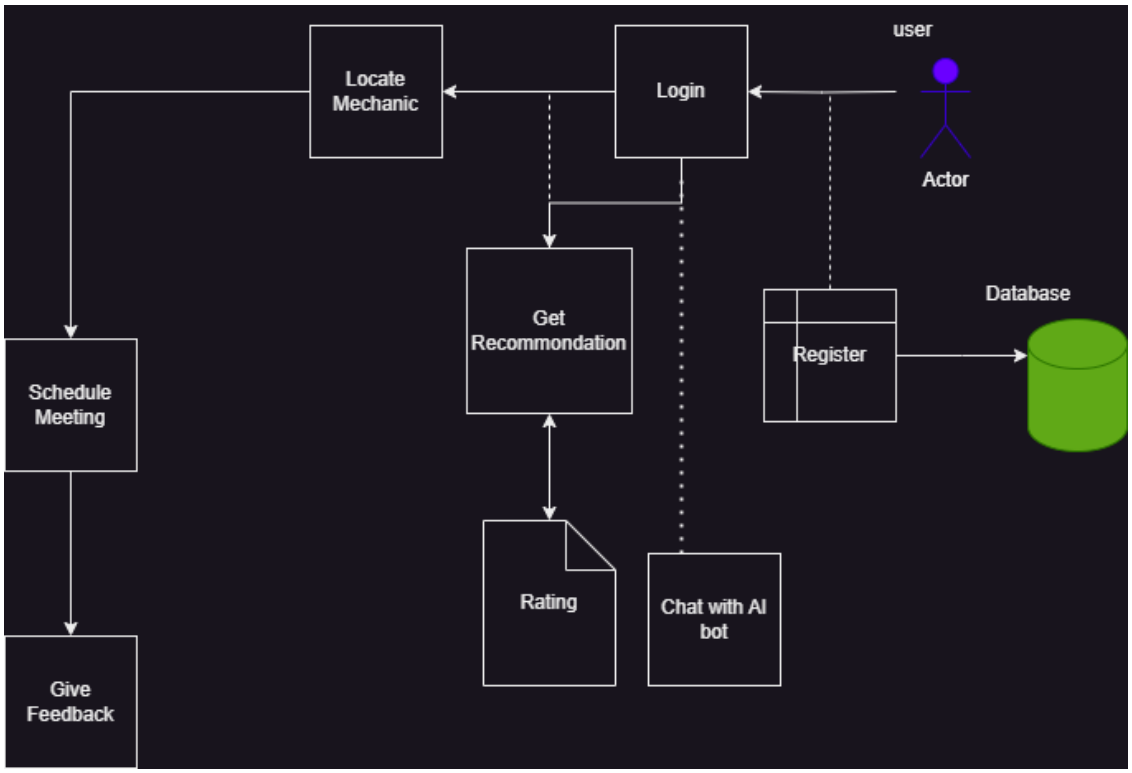


Figure 9 State Transition Diagram

4.9 Component Diagram



4.10 Deployment Diagram

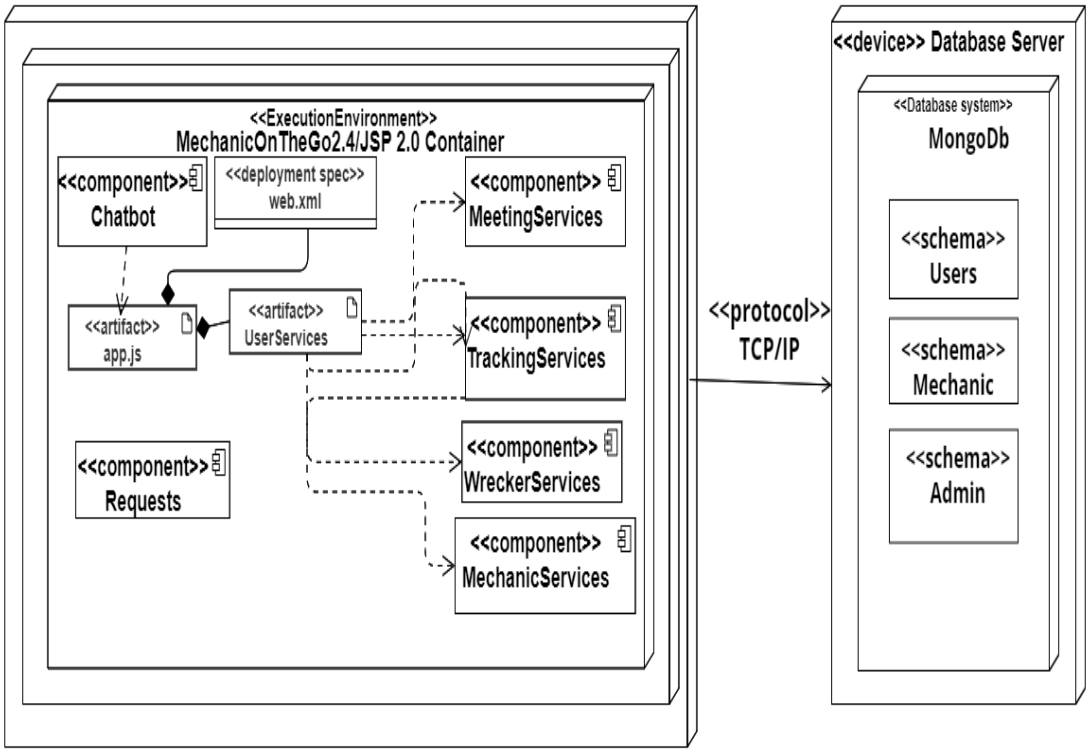


Figure31 Component Diagram

Chapter 5: Testing

5.1 Test Case Specifications

Positive Test Case	
ID	TC_01
Priority	High
Description	To verify user authentication to system.
Reference	FR_01
Users	Driver, Mechanic.
Pre-requisites	A System is online. B User must have active login credentials. C User has internet access.
Steps	A Open application. B Enter login email id . C Enter Password. D Press Login.
Input	Login email id and password
Expected result	Successfully enters the system and main home page opens.
Status	Tested, passed.

Negative Test Case	
ID	TC_01
Priority	High
Description	To verify user authentication to system.
Reference	FR_01
Users	Driver, Mechanic.
Pre-requisites	A System is online. B User must have active login credentials provided by system administrator. C User has internet access.
Steps	A Open the application. B Enter login email id. C Enter Password. D Press Login.
Input	Incorrect Login email id or password or deactivated credentials.
Expected result	Does not allow access to system features and notifies the error.
Status	Tested, passed.

Positive Test Case	
ID	TC_02
Priority	High
Description	To register user into the application.
Reference	FR_02
Users	Driver, Mechanic.
Pre-requisites	A System is online. B Users must fill all fields. C User has internet access.
Steps	A Open the application. B Fill all fields. C Press Signup.
Input	First name, Last name, Email, Phone Number, Address, City, Password, Role
Expected result	Successfully create user account.
Status	Tested, passed.

Negative Test Case	
ID	TC_02
Priority	High
Description	To register user into the application.
Reference	FR_02
Users	Driver, Mechanic.
Pre-requisites	A System is online. B Users must fill all fields. C User has internet access.
Steps	A Open the application. B Fill all fields. C Press Signup.
Input	First name, Last name, Email, Phone Number, Address, City, Password, Role
Expected result	Does not create account for user.
Status	Tested, passed.

Positive Test Case	
ID	TC_03
Priority	Medium
Description	To locate a mechanic using the application.
Reference	FR_03
Users	Driver
Pre-requisites	A System is online. B User is logged in C User has internet access.
Steps	A Open the application. B Locate a nearby mechanic. C Initiate communication with the mechanic.
Input	Driver's request for assistance.
Expected result	Mechanic receives the request and responds appropriately.
Status	Tested, passed.

Negative Test Case	
ID	TC_03
Priority	Medium
Description	To locate a mechanic using the application.
Reference	FR_03
Users	Driver
Pre-requisites	A System is online. B User is logged into the application. C User has internet access. D Mechanic is not available in the system.
Steps	A Open the application. B Click on the "Locate Mechanic" option. C Verify the map displays available mechanics. D No mechanic is shown on the map.
Input	Mechanic location selection.
Expected result	Displays a message indicating that no mechanics are available at the moment.
Status	Tested, passed.

Positive Test Case	
ID	TC_04
Priority	High
Description	To request assistance from a mechanic.
Reference	FR_04
Users	Driver.
Pre-requisites	A System is online. B User is logged into the application. C User has internet access. D Mechanic is available and located.
Steps	A Open the application. B Click on the "Request Assistance" option. C Confirm the request.
Input	Request for mechanic assistance.
Expected result	Notifies the selected mechanic and initiates the assistance request.
Status	Tested, passed.

Negative Test Case	
ID	TC_04
Priority	High
Description	To request assistance from a mechanic.
Reference	FR_04
Users	Driver.
Pre-requisites	A System is online. B User is logged into the application. C User has internet access. D Mechanic is not available or cannot be located.
Steps	A Open the application. B Click on the "Request Assistance" option. C Attempt to confirm the request.
Input	Request for mechanic assistance.
Expected result	Displays an error message indicating that the selected mechanic is unavailable.
Status	Tested, passed.

Positive Test Case	
ID	TC_05
Priority	Medium
Description	To locate a wrekcer using the application.
Reference	FR_03
Users	Driver
Pre-requisites	A System is online. B User is logged in C User has internet access.
Steps	A Open the application. B Locate a nearby wrekcer. C Initiate communication with the wrekcer.
Input	Driver's request for assistance.
Expected result	wrekcer receives the request and responds appropriately.
Status	Tested, passed.

Negative Test Case	
ID	TC_05
Priority	Medium
Description	To locate a wrekcer using the application.
Reference	FR_05
Users	Driver
Pre-requisites	A System is online. B User is logged into the application. C User has internet access. D wrekcer is not available in the system.
Steps	A Open the application. B Click on the "Locate wrekcer " option. C Verify the map displays available wrekcer. D No wrekcer is shown on the map.
Input	wrekcer location selection.
Expected result	Displays a message indicating that no wrekcer are available at the moment.
Status	Tested, passed.

Positive Test Case	
ID	TC_06
Priority	High
Description	To request assistance from a wrekcer.
Reference	FR_04
Users	Driver.
Pre-requisites	A System is online. B User is logged into the application. C User has internet access. D wrekcer is available and located.
Steps	A Open the application. B Click on the "Request Assistance" option. C Confirm the request.
Input	Request for wrekcer assistance.
Expected result	Notifies the selected wrekcer and initiates the assistance request.
Status	Tested, passed.

Negative Test Case	
ID	TC_06
Priority	High
Description	To request assistance from a wrekcer.
Reference	FR_04
Users	Driver.
Pre-requisites	A System is online. B User is logged into the application. C User has internet access. D wrekcer is not available or cannot be located.
Steps	A Open the application. B Click on the "Request Assistance" option. C Attempt to confirm the request.
Input	Request for wrekcer assistance.
Expected result	Displays an error message indicating that the selected wrekcer is unavailable.
Status	Tested, passed.

Positive Test Case	
ID	TC_07
Priority	Medium
Description	To locate a petrol station using the application.
Reference	FR_03
Users	Driver
Pre-requisites	A System is online. B User is logged in C User has internet access.
Steps	A Open the application. B Locate a nearby petrol station. C Initiate communication with the petrol station owner or operator.
Input	Driver's request for assistance.
Expected result	petrol station receives the request and responds appropriately.
Status	Tested, passed.

Negative Test Case	
ID	TC_07
Priority	Medium
Description	To locate a petrol station using the application.
Reference	FR_03
Users	Driver
Pre-requisites	A System is online. B User is logged into the application. C User has internet access. D petrol station is not available in the system.
Steps	A Open the application. B Click on the "Locate petrol station " option. C Verify the map displays available mechanics. D No petrol station is shown on the map.
Input	petrol station location selection.
Expected result	Displays a message indicating that no petrol station are available at the moment.
Status	Tested, passed.

Positive Test Case	
ID	TC_08
Priority	High
Description	To request assistance from a petrol station.
Reference	FR_04
Users	Driver.
Pre-requisites	A System is online. B User is logged into the application. C User has internet access. D petrol station is available and located.
Steps	A Open the application. B Click on the "Request Assistance" option. C Confirm the request.
Input	Request for petrol station assistance.
Expected result	Notifies the selected petrol station and initiates the assistance request.
Status	Tested, passed.

Negative Test Case	
ID	TC_08
Priority	High
Description	To request assistance from a petrol station.
Reference	FR_04
Users	Driver.
Pre-requisites	A System is online. B User is logged into the application. C User has internet access. D petrol station is not available or cannot be located.
Steps	A Open the application. B Click on the "Request Assistance" option. C Attempt to confirm the request.
Input	Request for petrol station assistance.
Expected result	Displays an error message indicating that the selected petrol station is unavailable.
Status	Tested, passed.

Black Box Test Cases

Black box testing also known as Behavioral Testing, is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behavior or performance errors
- Initialization and termination errors

5.1.1 Equivalence Partitions (EP)

Equivalence class partitioning (EP) is a very widely used method to decrease the number of possible test cases that are required to test a system.

Equivalence Partitions for Login

Variables	Valid Classes	Invalid Classes
Email	1. Contains '@' character and .domain. 2. Case sensitive. 3. Compulsory field.	1. Doesn't contain '@' character and .domain 2. Empty Field.
Password	1. Length should be greater than 6 characters. 2. May contain symbols, alphabets [a-z A-Z] and digits	1. Length less than 6 characters. 2. Empty field.

Equivalence Partitions for Signup

Variables	Valid Classes	Invalid Classes
First Name	1. Contains alphabets [a-z A-Z] only. 2. Compulsory field.	1. Shouldn't contain Numbers. 2. Shouldn't contain Symbols. 3. Empty field.
Last Name	3. Contains alphabets [a-z A-Z] only. 4. Compulsory field.	4. Shouldn't contain Numbers. 5. Shouldn't contain Symbols. 6. Empty field.
Email	1. Contains '@' character and. domain. 2. Case sensitive. 3. Compulsory field.	1. Doesn't contain '@' character and .domain 2. Empty Field.
Phone Number	1. Contains numbers only. 2. Compulsory field.	1. Shouldn't contain alphabets. 2. Shouldn't contain Symbols. 3. Empty field.
Password	1. Length should be greater than 6 characters. 2. May contain symbols, alphabets [a-z A-Z] and digits [0-9].	1. Length less than 6 characters. 2. Empty field.

Equivalence Partitions for Review

Variables	Valid Classes	Invalid Classes
Stars	1. Minimum input 1 star. 2. Maximum input 5 stars.	1. More than 5 stars are not allowed.
Comment	1. May contain symbols, alphabets [a-z A-Z] and digits [0-9].	-

Equivalence Partitions for Real-time Tracking

Variables	Valid Classes	Invalid Classes
Tracking Information	Accurate tracking	Inaccurate tracking

Equivalence Partitions for Service Completion:

Variables	Valid Classes	Invalid Classes
Service Status	Successfully completed	Incomplete service

Equivalence Partitions for Payment Processing

Variables	Valid Classes	Invalid Classes
Payment Status	Successful payment	Payment failure

Equivalence Partitions for Service Selection

Variables	Valid Classes	Invalid Classes
Selected Service	Valid service type	Service not offered

Equivalence Partitions for Mechanic Profiles

Variables	Valid Classes	Invalid Classes
Mechanic Ratings	Highly rated mechanics	Low-rated mechanics

5.1.2 Boundary Value Analysis

A boundary value is an input or output value on the border of an equivalence partition, includes minimum and maximum values at inside and outside boundaries. Normally Boundary value analysis is part of stress and negative testing.

Boundary Value Analysis for login

Variables	Minimum	Maximum	Boundary Value
Email	1. Doesn't contain '@' character and .domain 2. Empty Field.	1. Doesn't contain '@' character and .domain 2. Empty Field.	1. Contain '@' character and .domain
Password	1. Length less than 6 characters. 2. Empty field.	1. Length greater than 30 characters. 2. Empty field.	1. Length should be greater than 6 and less than 30 characters. 2. May contain symbols, alphabets [a-z A-Z] and digits [0-9].

Boundary Value Analysis for Signup

Variables	Minimum	Maximum	Boundary Value
First Name	1. Empty field.	1. Shouldn't contain Numbers. 2. Shouldn't contain Symbols.	1. Contains alphabets [a-z A-Z] only.
Last Name	2. Empty field.	3. Shouldn't contain Numbers. 4. Shouldn't contain Symbols.	2. Contains alphabets [a-z A-Z] only.
Email	1. Doesn't contain '@' character and .domain 2. Empty Field.	1. Doesn't contain '@' character and .domain 2. Empty Field.	1. Contain '@' character and .domain
Password	1. Length less than 6 characters. 2. Empty field.	1. Length greater than 30 characters. 2. Empty field.	1. Length should be greater than 6 and less than 30 characters. 2. May contain symbols, alphabets [a-z A-Z] and digits [0-9].

Boundary Value Analysis for Search

Variables	Minimum	Maximum	Boundary Value
Search	1. Empty Field.	1. Empty Field	1. May contain symbols, alphabets [a-z A-Z] and digits [0-9].

Boundary Value Analysis for Location Selection

Variables	Minimum	Maximum	Boundary Value
Location of need	1. Latitude and Longitude within valid ranges.	1. N/A	1. Latitude and Longitude within specified valid ranges.

Boundary Value Analysis for Mechanic Request

Variables	Minimum	Maximum	Boundary Value
Request Time	1. Current time.	1. N/A	1. Current time.

Boundary Value Analysis for Mechanic'S Arrival

Variables	Minimum	Maximum	Boundary Value
Completion Time	1. Current time.	1. N/A	1. Current time.

Boundary Value Analysis for Reviews

Variables	Minimum	Maximum	Boundary Value
User Rating	1. Minimum 1 star rating.	1. Maximum 5 stars rating.	1 star rating.

Boundary Value Analysis for Reviews

Variables	Minimum	Maximum	Boundary Value
User Rating	1. Minimum 1 star rating.	1. Maximum 5 stars rating.	1 star rating.

5.1.3 Decision Table Testing

Conditions	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Valid email	0	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1
Email in database	0	0	1	0	0	1	0	0	1	1	0	1	1	0	1	1
User is blocked	0	0	0	1	0	0	1	0	1	0	1	1	0	1	1	1
Valid and correct password	0	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1
Action	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Error message	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
Login password	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Conditions	1	2	3	4	5	6	7
Valid email	0	1	1	1	1	1	1
Email in database	0	0	1	0	1	1	1
User is blocked	0	0	0	0	1	0	1
Valid and correct password	-	0	0	1	0	1	1
Action	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Error message	1	1	1	1	1	0	1
Login password	0	0	0	0	0	1	0

5.1.4 State transition Testing

State Transition testing, a black box testing technique, in which outputs are triggered by changes to the input conditions or changes to 'state' of the system. In other words, tests are designed to execute valid and invalid state transitions.

States:

1. NotLoggedIn
2. LoggedIn
3. LocationSelected
4. Service Selected
5. Mechanic/wrecker Requested
6. Mechanic/wrecker Accepted
7. Ride In Progress
8. Payment Processing
9. Ride Completed
10. RatingProvided

Transitions:

Login: NotLoggedIn -> LoggedIn

Logout: LoggedIn -> NotLoggedIn

Select Location: LoggedIn -> LocationSelected

Select a Mechanic/wrecker: Mechanic/wrecker Selected -> Mechanic/wrecker Selected

Accept Driver’s Request: Mechanic/wrecker assistance Requested -> Mechanic/wreckerAccepted

Start Ride: Driver Accepted -> Ride In Progress

Complete Ride: Ride In Progress -> Payment Processing

Process Payment: Payment Processing -> Ride Completed

Provide Rating: Ride Completed -> Rating Provided

5.1.5 Use Case Testing

Use Case Testing is a functional black box testing technique that helps testers to identify test scenarios that exercise the whole system on each transaction basis from start to finish.

- **Use Case Testing for Signup**

Description:

This use case describes the process when a user is trying to register for the first time.

Actors:

- A: User
- B: System

Preconditions:

- The user has not registered account.
- The user's location services are enabled.

Main Flow:

1. A: Enter First Name
2. A: Enter Last Name
3. A: Enter Email
4. A: Enter Phone Number
5. A: Enter password
6. A: Address
7. A: Click on register account button
8. S:Validate details
9. S:Register the account

Postcondition:

- The user is successfully registered.

- **Use Case Testing for Login**

Description:

This use case describes the process when a user is trying to login to the system.

Actors:

- A: User
- B: System

Preconditions:

- The user has active internet.
- The user's location services are enabled.

Main Flow:

1. A: Enter email
2. A: Enter password
3. A: Click Login button
4. S: Validate details
5. S: Allows access
6. Email or password not valid.
   S: Display error message
7. Empty field
   S: Display error message

Postcondition:

- The user is successfully login.

- **Use Case Testing for Mechanic / Wrecker Request**

Description:

This use case describes the process when a user is trying to request a mechanic/ wrecker.

Actors:

- A: User
- B: System

Preconditions:

- The user has a registered account and is logged into the app.
- The user's location services are enabled.
- There are available mechanic/ wrecker in the vicinity.

Main Flow:

1. **User Sets Pickup Location:**
    - User selects the option to request a mechanic/wrecker.
    - User sets the pickup location by either manually entering it or using their current location.
2. **User Chooses Service Type:**
    - User selects the type of service they want (e.g., mechanic, wrecker).
3. **User Requests a Service:**
    - User confirms the service request.
4. **System Searches for mechanic/wreckers:**
    - The system identifies available mechanic/wreckers in the vicinity.
5. **System Assigns a mechanic/wreckers:**
    - The system assigns the request to an available mechanic/wreckers.
6. **User Receives Driver Information:**
    - User receives information about the assigned mechanic/wreckers, including name, vehicle details, and estimated arrival time.

7. **Mechanic/wreckers Accepts the Ride:**
    - The assigned mechanic/wreckers accepts the service request.
8. **User Receives Confirmation:**
    - User receives confirmation that the mechanic/wreckers has accepted the request.

Postcondition:

The ride is in progress.

- **Use Case Testing for review**

Description:

This use case describes the process when a user is trying to give review.

Actors:

- A: User
- B: System

Preconditions:

- The user has active internet.
- The user has completed a request for mechanic/wrecker

Main Flow:

1. A: Enter Feedback message.
2. B: Validate details
3. B: Submit.
4. Internet connection lost. B: Display error message.
5. Feedback fields empty. B: Display error message.

Postcondition:

- The user has successfully given a review.

- **Use Case Testing for Request Cancellation by mechanic/wrecker**

Description:

This scenario tests how the system handles a situation where the assigned mechanic/wrecker cancels the ride.

Actors:

- A: mechanic/wrecker
- B: System

Preconditions:

User has requested a ride, and a mechanic/ wrecker has accepted it.

Main Flow:

The assigned mechanic/ wrecker cancel the request.

Postcondition:

Verify that the system promptly informs the user about the cancellation and proceeds to assign an alternate mechanic/wrecker.

5.2 White Box Test Cases

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

5.2.1 Cyclometric complexity

Cyclometric complexity is a source code complexity measurement that is being correlated to a number of coding errors. It is calculated by developing a Control Flow Graph of the code that measures the number of linearly-independent paths through a program module.
Lower the Program's cyclometric complexity, lower the risk to modify and easier to understand.

5.3 Performance testing

Performance testing, a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage.

1. Load Testing:

Description:

This test evaluates how well the system performs under normal and peak loads.

Scenarios:

1. Simulate a gradual increase in the number of simultaneous ride requests to evaluate the system's response to increasing load.
2. Simulate a sudden surge in ride requests to test the system's ability to handle peak loads.
3. Evaluate the performance of the system when multiple users are accessing various features simultaneously (e.g., requesting rides, viewing maps, providing feedback).

Metrics:

- Response time for ride requests.
- Throughput (number of successful ride requests per unit of time).
- System resource utilization (CPU, memory, network).

2. Scalability Testing:

Description:

This test assesses the app's ability to scale horizontally by adding more resources or servers.

Scenarios:

1. Gradually increase the number of server instances or resources and monitor the system's performance.
2. Test the app's performance when running on multiple servers concurrently.

Metrics:

- Response time as the system scales.
- System resource utilization as the number of servers increases.

3. Stress Testing:

Description:

This test evaluates the system's robustness by subjecting it to extreme conditions.

Scenarios:

1. Simulate a high number of ride requests exceeding the system's capacity.
2. Increase the load beyond the expected peak capacity to identify the breaking point.
3. Introduce network latency or packet loss during ride requests.

Metrics:

- Identify the point of failure.
- Response time and system stability under stress conditions.

4. Endurance Testing:

Description:

This test assesses the system's performance over an extended period to identify potential memory leaks or degradation.

Scenarios:

1. Simulate continuous ride requests over an extended time.
2. Monitor system performance for memory leaks or resource exhaustion.

Metrics:

- System stability over time.
- Memory usage trends.

5. Network Performance Testing:

Description:

This test evaluates the app's performance under various network conditions.

Scenarios:

1. Test the app's performance on different network speeds (3G, 4G, and Wi-Fi).

2. Introduce network fluctuations (latency, packet loss) during ride requests.

Metrics:

- Response time under different network conditions.
- Reliability of the app under poor network conditions.

6. Concurrency Testing:

Description:

This test evaluates the system's performance when multiple users perform different actions simultaneously.

Scenarios:

1. Simulate multiple users simultaneously requesting services, providing feedback, and using other app features.
2. Evaluate how the system handles overlapping activities.

Metrics:

- Response time for concurrent activities.
- Throughput under simultaneous usage.

7. Geographical Load Balancing:

Description:

This test evaluates how well the system distributes load across different geographical regions.

Scenarios:

1. Simulate service requests from different geographical locations.
2. Evaluate the load balancing mechanism to ensure efficient resource utilization.

Metrics:

- Response time for service requests from different regions.
- Distribution of resources based on geographical load.

5.4 Stress Testing

Stress testing a Non-Functional testing technique that is performed as part of performance testing. During stress testing, the system is monitored after subjecting the system to overload to ensure that the system can sustain the stress.

The recovery of the system from such phase (after stress) is very critical as it is highly likely to happen in production environment.

5.5 System Testing

System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.

System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and NonFunctional testing.

5.6 Regression Testing

Regression testing a black box testing technique that consists of re-executing those tests that are impacted by the code changes. These tests should be executed as often as possible throughout the software development life cycle. Types of Regression Tests:

Final Regression Tests

A "final regression testing" is performed to validate the build that hasn't changed for a period of time. This build is deployed or shipped to customers.

Regression Tests

A normal regression testing is performed to verify if the build has NOT broken any other parts of the application by the recent code changes for defect fixing or for enhancement.

5.6.1 Selecting Regression Tests

- Requires knowledge about the system and how it affects by the existing functionalities.
- Tests are selected based on the area of frequent defects.
- Tests are selected to include the area, which has undergone code changes many a times.
- Tests are selected based on the criticality of the features.

5.6.2 Regression Testing Steps

Regression tests are the ideal cases of automation which results in better **R**eturn **on I**nvestment (ROI).

- Select the Tests for Regression.
- Choose the apt tool and automate the Regression Tests
- Verify applications with Checkpoints
- Manage Regression Tests/update when required
- Schedule the tests
- Integrate with the builds
- Analyze the results

Chapter 6: Tools and Techniques

6.1 Languages and Frameworks

- React Native
- Node Js
- Express Js
- MongoDB

6.2 APIs

Javascript framework Node Js and MongoDB database integration. Also deployed on server.

6.3 Libraries

```
{
  "name": "mechanic-on-the-go",
  "version": "0.0.1",
  "private": true,
  ▷ Debug
  "scripts": {
    "android": "react-native run-android",
    "ios": "react-native run-ios",
    "start": "react-native start",
    "test": "jest",
    "lint": "eslint . --ext .js,.jsx,.ts,.tsx"
  },
  "dependencies": {
    "@react-native-async-storage/async-storage": "^1.15.5",
    "@react-native-community/masked-view": "^0.1.11",
    "@react-navigation/bottom-tabs": "^5.11.11",
    "@react-navigation/material-top-tabs": "^5.3.15",
    "@react-navigation/native": "^5.9.4",
    "@react-navigation/stack": "^5.14.5",
    "@reduxjs/toolkit": "^1.5.1",
    "@thecodingmachine/redux-toolkit-wrapper": "^2.0.1",
    "axios": "^0.23.0",
    "react": "17.0.2",
    "react-native": "0.67.2",
    "react-native-gesture-handler": "^1.10.3",
    "react-native-image-picker": "^4.7.3",
    "react-native-pager-view": "^5.4.7",
    "react-native-restart": "^0.0.22",
    "react-native-safe-area-context": "^3.3.2",
    "react-native-screens": "^3.8.0",
    "react-native-splash-screen": "^3.3.0",
    "react-native-svg": "^12.1.1",
    "react-native-tab-view": "^3.1.1",
    "react-native-track-player": "^1.2.7",
    "react-redux": "^7.2.5",
    "redux": "^4.1.0"
  },
```

Chapter 7: Summary and Conclusion

Our mobile application consists of a Home, Client, List of all mechanics, List of all recommended Mechanic, Wrecker Service, Petroleum Service, Maps and a drawer Screens. The travelers traveling on motorway got traveling emergency facilities easily and precisely but in local areas like subways, GT road, and other local connected roads lack of those on way facilities, our app is working to provide those services to those needy people. As our app has emergency services (motorway, highway police, fueling station) and can provide the mechanic from well-reputed companies or user can seek help from the closest private mechanics through this application.

You can get wrecker and mechanic by clicking our app and notifying that you simply need a mechanic or wrecker. Our app will then ask you for your current location and locate a mechanic or wrecker who is nearby. You will be asked about your car problem if you know about your problem then you can fill the blank form otherwise you can say that you are not familiar with the problem. Then you will be given a data for the mechanics after that if you confirm that mechanic then his car/bike registration number will be provided to you to sure that you get the right guy. Once mechanic have received your location, information regarding your problem and your vehicle (like vehicle registration number, name, company name and model).

Our app will send you the notification that the mechanic is on his way to you and will show you his live location so you will be able to know when he will come to you. After getting services form the mechanic app will ask you to rate the experience. Same scenario works for wrecker you will locate the wrecker and the wrecker will come to your location and will take you to the closest mechanic.

Conclusion

The main thing that we concluded form our project is how the technology is getting evolved day by day to provide better user experience to the end user. Our project shows the importance of advance technologies and features such as requesting a mechanic / wrecker in need of emergency

Implementing such advance features requires a lot of research and resources. As the project proceed, we come to know that it is a board area in research field as well having a lot of use cases in various fields such as education, entertainment and industries.

It is due to the guidance of our supervisor that we could achieve our goal within the required time constraints and each of our group members have learn a new technology which will help us in getting better opportunities in the future.

Chapter 8: User Manual

Table of Contents

    - If you haven't created an account, follow the "User Registration" instructions in Section 1.
- Login and Authentication
    - Open the application and tap on "Login."
    - Enter your registered email address and password.
    - Tap on "Login" to access your account.
- Account Recovery (if applicable)
    - If you forget your password, tap on "Forgot Password."
    - Follow the instructions to reset your password and regain access to your account.

4. Requesting Mechanic Services

- Open the Application
    - Launch the application on your device.
- Select "Mechanic" from the Main Menu
    - Tap on the "Mechanic" option on the main screen.
- Choose "Request Mechanic"
    - Select the "Request Mechanic" option to request a mechanic for your vehicle.
- Provide Your Current Location (GPS or Manual Entry)
    - Allow the application to access your device's location services or enter your location manually.
- Describe the Vehicle Problem (Optional)
    - If you know the specific issue with your vehicle, provide a brief description in the designated field.
- Confirm if You're Familiar with the Problem
    - Indicate whether you are familiar with the problem or not.
- Select "Find Mechanic"

- Tap on the "Find Mechanic" button to locate mechanics near your location.

- View Nearby Mechanics and Their Prices
    - The application will display a list of nearby mechanics along with their prices and ratings.
- Choose a Mechanic Based on Reviews, Ratings, and Price
    - Review the information provided and select a mechanic based on reviews, ratings, and price that suits your preferences.
- Confirm the Mechanic Selection
    - Confirm your selection and proceed to the next step.
- Receive Confirmation with Mechanic Details (Registration Number, Name)
    - Once you confirm, you will receive a confirmation screen with the mechanic's details, including registration number and name.
- Track Mechanic's Live Location and Estimated Time of Arrival (ETA)
    - The application will provide a real-time location map of the mechanic and their estimated time of arrival (ETA) at your location.

5. Getting Mechanic Services

- Await Mechanic's Arrival
    - Wait for the mechanic to arrive at your location based on the provided ETA.
- Confirm Mechanic's Identity with Registration Number
    - Verify the mechanic's identity by matching their registration number with the one provided by the application.
- Discuss and Show the Vehicle Problem to the Mechanic
    - Communicate the issue with your vehicle to the mechanic and show them the specific problem areas if necessary.
- Mechanic Performs the Necessary Repairs or Service
    - Allow the mechanic to carry out the required repairs or service on your vehicle.
- Stay Updated on the Progress of the Service

    - The application may provide updates on the progress of the service, such as completion percentage or estimated completion time.
- Verify Completion of Service
    - Once the mechanic completes the service, review the work done and ensure the problem is resolved to your satisfaction.
- Pay the Mechanic (In-app or Cash, as per App Policy)
    - Depending on the app's payment options, pay the mechanic either through the application or with cash, following the app's policy.
- Provide Feedback and Rate the Mechanic's Service
    - Provide feedback and rate the mechanic's service through the application to help improve the overall user experience.

6. Requesting Wrecker Services

- Open the Application
    - Launch the application on your device.
- Select "Wrecker" from the Main Menu
    - Tap on the "Wrecker" option on the main screen.
- Choose "Request Wrecker"
    - Select the "Request Wrecker" option to request a wrecker service.
- Provide Your Current Location (GPS or Manual Entry)
    - Allow the application to access your device's location services or enter your location manually.
- Confirm the Request
    - Confirm your request for a wrecker service.
- Receive Confirmation with Wrecker Details (Registration Number, Name)
    - Once confirmed, you will receive a confirmation screen with the wrecker's details, including registration number and name.
- Track Wrecker's Live Location and Estimated Time of Arrival (ETA)

    - The application will provide a real-time location map of the wrecker and their estimated time of arrival (ETA) at your location.
- Follow the Instructions Provided by the Wrecker for Safe Vehicle Loading and Transportation
    - When the wrecker arrives, follow any instructions provided by the wrecker operator to ensure the safe loading and transportation of your vehicle.

7. Requesting Petrol Services

- Open the Application
    - Launch the application on your device.
- Select "Petrol" from the Main Menu
    - Tap on the "Petrol" option on the main screen.
- Choose "Request Petrol"
    - Select the "Request Petrol" option to request petrol services.
- Provide Your Current Location (GPS or Manual Entry)
    - Allow the application to access your device's location services or enter your location manually.
- Confirm the Request
    - Confirm your request for petrol services.
- Receive Confirmation with Delivery Details (Vehicle Type, Registration Number, Name)
    - Once confirmed, you will receive a confirmation screen with the delivery details, including vehicle type, registration number, and name.
- Track the Petrol Delivery Vehicle's Live Location and Estimated Time of Arrival (ETA)
    - The application will provide a real-time location map of the petrol delivery vehicle and their estimated time of arrival (ETA) at your location.
- Ensure Your Vehicle is Ready for Petrol Refuelling

    - Make sure your vehicle is accessible and ready for petrol refuelling when the delivery vehicle arrives.
- Make Payment for the Petrol Refuelled (In-app or Cash, as per App Policy)
    - Depending on the app's payment options, pay for the refuelled petrol either through the application or with cash, following the app's policy.
- Provide Feedback and Rate the Petrol Service
    - Provide feedback and rate the petrol service through the application to help improve the overall user experience.

8. Frequently Asked Questions (FAQs)

- Common Questions and Answers
- Troubleshooting Tips

Chapter 9: Lessons Learnt and Future Work

During the course of the entire project, we faced a lot of problems. We have discussed the problems and their possible solutions below.

Integration of Geo Location Api:

The potential for geolocation to transform apps is almost limitless. As a central feature of delivery, fleets and an advertiser's dream, mobile location data is now part of everyday life.

But getting quality location data without discrepancies that could severely alter the user's experience is not a guarantee. GPS can be inaccurate and battery-draining, making it impossible to target users at the right time. For app developers, location features must be done right.

Where location features are fundamental to app functionality such as delivery, logistics or fitness, inaccurate tracking leads to a sub-par or unusable user experience. For marketers targeting users based on their location, inaccurate location data severely compromises campaigns.

But why is location data inaccurate, and what can we do to improve it?

Environment, device state, and satellite position all impact accuracy and precision. Dense urban environments usually mean less accurate location data. Buildings, walls, trees or any high obstacle can cause signal blocks, as satellites struggle to send signals to a user without interference. Equally, if there are not enough satellites, GPS receivers struggle to calculate location accurately. To best combat this, it's essential to connect an app to all of the available data sources on a device, including GPS, cell towers, Wi-Fi, Bluetooth, and the device's hardware.

Future Work

We are continuously working to enhance and improve the user experience of our mechanic service application. Here are some potential future updates and features that we are considering:

1. **Integration with Vehicle Maintenance Records:** We are working on integrating the application with vehicle maintenance records. This will allow users to maintain a digital log of their vehicle's service history, track previous repairs, and receive reminders for upcoming maintenance tasks.
2. **Emergency Assistance:** In critical situations such as accidents or breakdowns, we aim to provide an emergency assistance feature. This feature will enable users to quickly request immediate help, including emergency towing services or on-site repairs.
3. **User Community and Reviews:** We plan to introduce a user community section within the application where users can share their experiences, provide recommendations, and exchange information. This will help create a collaborative environment and allow users to make more informed decisions when selecting service providers.
4. **Integration with Payment Platforms**: To offer greater convenience, we are considering integrating the application with popular payment platforms. This will provide users with additional options for seamless and secure payment transactions.

References

Websites

[1] "Uber," Uber, [Online]. Available: http://www.uber.com

[2] "Careem," Careem, [Online]. Available: https://www.careem.com/

[3] "Star Of Service," Star Of Service, [Online]. Available: https://www.starofservice.pk/

[4] "GoCurb," Curb, [Online]. Available: https://gocurb.com/about/

[5] "Wrench," Wrench, [Online]. Available: https://wrench.com/

[6] "Your Mechanic," Your Mechanic, [Online]. Available: https://www.yourmechanic.com/

[7] "Look UP," LookUp, [Online]. Available: https://www.lookup.pk/dynamic/categories.aspx?class_id=1