

ProjectReport

ON

On Road Vehicle Breakdown Assistance (ORVBA)

By

Shivani Vijay Bhojak - 1912008
MCA(2019-2022)

INDEX

Chapter No.		Name of Topic	Page No.
1		Introduction	
	1.1	Institute Profile	
	1.2	Abstract	4
	1.3	Existing System and Need for System	4
	1.4	Scope of System	5
	1.5	Operating Environment - Hardware and Software	5
	1.6	Brief Description of Technology Used	7
2		Proposed System	
	2.1	Study of Similar Systems	13
	2.2	Feasibility Study	14
	2.3	Objectives of Proposed System	15
	2.4	Users of System	15
3		Analysis and Design	
	3.1	System Requirements	17
	3.2	Entity Relationship Diagram	19
	3.3	Table Structure	20
	3.4	Use Case Diagrams	24
	3.5	Class Diagram	27
	3.6	Activity Diagram	28
	3.7	Deployment Diagram	29
	3.8	Module Hierarchy Diagram	30
	3.9	Sample Input and Output Screens	31

CHAPTER 1

INTRODUCTION

1.1 Institute Profile

1.2 Abstract

On Road Vehicle Breakdown Assistance (ORVBA) Web App is going to be a good solution for the people who seek help in the remote locations with mechanical issues of their vehicle. Users of the On Road Vehicle Breakdown Assistance will be the registered public and they will be getting connected with the particular mechanic through the trustworthy (ORVBA) app. ORVBA is used to find the nearest mechanic from the current user location via GPS (Google Maps).

1.3 Existing System and need for system:

In an existing system there are users who have their own mechanic database which is very minimal. And also they have no idea if their vehicles are broke down or had any mechanical issue in remote locations or any long distant locations from their known mechanic shops. Users with the contacts of people at the particular place may look for a help from them only if they are ready to do.

It is not possible to find out the suitable mechanic for the desired service at remote locations. The only way they have is to look for any other transportation at the time of issue and then they need to get a mechanic to the particular location at which they have left their vehicle.

1.4 Scope of System

The scope of this system is to develop a platform which improve the efficiency of mechanic and user. This system provide user friendly interface. Users of the On Road Vehicle Breakdown Assistance will be the registered public and they will be getting connected with the particular

mechanic through the trustworthy (ORVBA) app. ORVBA is used to find the nearest mechanic from the current user location via GPS (Google Maps). When user will place a service request, the mechanic will get live location of the user and locate using GPS. User will give review about service.

Also they are under monitoring by the ORVBA system for not charging any extra service fee or any other issue regarding the service as every user is updating their feedback about the availed service through ORVBA system.

1.5 Operating Environment – Hardware and Software

Hardware Requirements:-

Processor : Intel(R) 2.10GHz

Installed memory (RAM) : 4 GB

Hard Disk : 160 GB

Software Requirements: -

Front-End : HTML, CSS, JavaScript and Bootstrap

Back-End : Node.js, MYSQL

1.6 Brief Description of Technology used

1.6.1 Operating system used –

As the application will be stored on server and can be used from any device browser, no specific operating system is required. For the server we will use Windows to host the system

1.6.2 Database

MySQL:

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL).

MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing, for this system we're using a remote MySQL database.

Front End :

We have implemented JavaScript for all the Client side validations. Client side JavaScript is designed to reside inside HTML document & ensure they run properly. It is object based, event driven, platform independent. These are important parts of any Web application to implement Client side Validations and the invalid data is not submitted. The form is not submitted until user fills in correct data. It is extremely useful to restrict mistakes by user.

HTML :

HTML or HyperText Markup Language is the main markup language for creating web pages and other information that can be displayed in a web browser.

HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example . The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, further tags, comments and other types of text-based content.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behaviour of HTML web pages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML.

CSS :CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities or weights are calculated and assigned to rules, so that the results are predictable.

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts.[1] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

Javascript :

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give

structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

Incorporating JavaScript improves the user experience of the web page by converting it from a static page into an interactive one. To recap, JavaScript adds behavior to web pages.

JavaScript is mainly used for web-based applications and web browsers. But JavaScript is also used beyond the Web in software, servers and embedded hardware controls.

Bootstrap :

Bootstrap is a free and open source front end development framework for the creation of websites and web apps. The Bootstrap framework is built on HTML, CSS, and JavaScript (JS) to facilitate the development of responsive, mobile-first sites and apps.

Responsive design makes it possible for a web page or app to detect the visitor's screen size and orientation and automatically adapt the display accordingly; the mobile first approach assumes that smartphones, tablets and task-specific Mobile apps are employees' primary tools for getting work done and addresses the requirements of those technologies in design.

Bootstrap includes user interface components, layouts and JS tools along with the framework for implementation. The software is available precompiled or as source code.

Backend:

Node JS:

As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection, the callback is fired, but if there is no work to be done, Node.js will sleep.

This is in contrast to today's more common concurrency model, in which OS threads are employed. Thread-based networking is relatively inefficient and very difficult to use.

Furthermore, users of Node.js are free from worries of dead-locking the process, since there are no locks. Almost no function in Node.js directly performs I/O, so the process never blocks.

Because nothing blocks, scalable systems are very reasonable to develop in Node.js.

If some of this language is unfamiliar, there is a full article on [Blocking vs. Non-Blocking](#).

Node.js is similar in design to, and influenced by, systems like Ruby's Event Machine and Python's Twisted. Node.js takes the event model a bit further. It presents an event loop as a runtime construct instead of as a library. In other systems, there is always a blocking call to start the event-loop. Typically, behavior is defined through callbacks at the beginning of a script, and at the end a server is started through a blocking call like `EventMachine::run()`. In Node.js, there is no such start-the-event-loop call. Node.js simply enters the event loop after executing the input script. Node.js exits the event loop when there are no more callbacks to perform. This behavior is like browser JavaScript — the event loop is hidden from the user.

HTTP is a first-class citizen in Node.js, designed with streaming and low latency in mind. This makes Node.js well suited for the foundation of a web library or framework.

Node.js being designed without threads doesn't mean you can't take advantage of multiple cores in your environment. Child processes can be spawned by using our `child_process.fork()` API, and are designed to be easy to communicate with. Built upon 8 that same interface is the cluster module, which allows you to share sockets between processes to enable load balancing over your cores.

CHAPTER 2

PROPOSED SYSTEM

2.1 Study of similar System

GoBumpr is a app that offers services like general service, roadside assistance, free diagnostics among others. The services are currently only available only in Chennai .The assistance provided to the user is highly limited to the types of services that is being provided. Assistance through helpline is highly prone to unavailability that makes the users experience worse. The available services to the users are not accessible to the users in terms of locality. This inspired us to create a system that helps to find mechanics easily and quickly.

2.2 Feasibility Study

OPERATIONAL FEASIBILITY

Operational feasibility study tests the operational scope of the software to be developed. The proposed system is fully GUI based that is very user friendly and all inputs to be taken are self-explanatory. All the operations will be well described and easy to access.

Operational feasibility deals with risk factors. It checks the impact of the proposed software on required project or module. This system is developed for the people who seek help in the remote locations with mechanical issues of their vehicle.After acceptance of request the notification will be send to both the user and mechanic which will contain complete details (Name, mob no., Location link).the system is operationally feasible.

TECHNICAL FEASIBILITY

Technical feasibility evaluates the technical complexity of the expert system and often involves determining whether the expert system can be implemented with state-of-the-art techniques and tools.

The technologies and tools are associated with this system are:

- HTML & CSS
- Node Js
- MySQL

Technical feasibility report: Since the users already has the required hardware and a supporting Operating System, it is technically feasible.

ECONOMICAL FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The cost of the hardware and software benefits in the form of reduced costs or fewer costly errors. Since the system is developed as part of project work, there is no manual cost to spend for The the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

2.3 Objectives of proposed system

- To identify the reason for vehicle breakdown.
- To build a common platform that connect with mechanic and user.

The objective of this work is to propose a mobile system which assists a user to search mechanic at any location. We assume that the proposed system works on a smart phone such as Android smart phones and iPhones which needs Internet connection. We intend a user to use our system easily. The goal of the application is to provide a platform where users find solution of vehicle breakdown. This application can be used as a standalone application or it can also be used as a part of a more sophisticated application.

2.4 Users of System

- Admin
- User
- Mechanic

CHAPTER 3

ANALYSIS & DESIGN

3.1 System Requirements

Functional requirement

R1 REGISTRATION

R1.1 Fill personal details

R1.2 Submit information

R1.3 Get login credentials

R2 Service Request

R 2.1 Select Vehicle Type

R 2.2 Select Vehicle problem

R 2.3 Send Service Request

R 2.4 A confirmation notification will be generated automatically and send to the mechanic which will contain user details

R3 Accept Service Request

R 3.1 View User details

R 3.2 Accept service

R 3.3 A confirmation notification will be generated automatically and send to the user which will contain mechanic details

R 3.4 Track user location

R4 Review

R4.1 Give Review

Non-Functional requirements

- Performance requirement

The website is up 24/7 for registration and booking of services

- Safety requirement

Database backup is taken once in a week without hampering website execution

Every user has security login that is password and user ID

- Software quality attributes

Portability

User-friendly UI

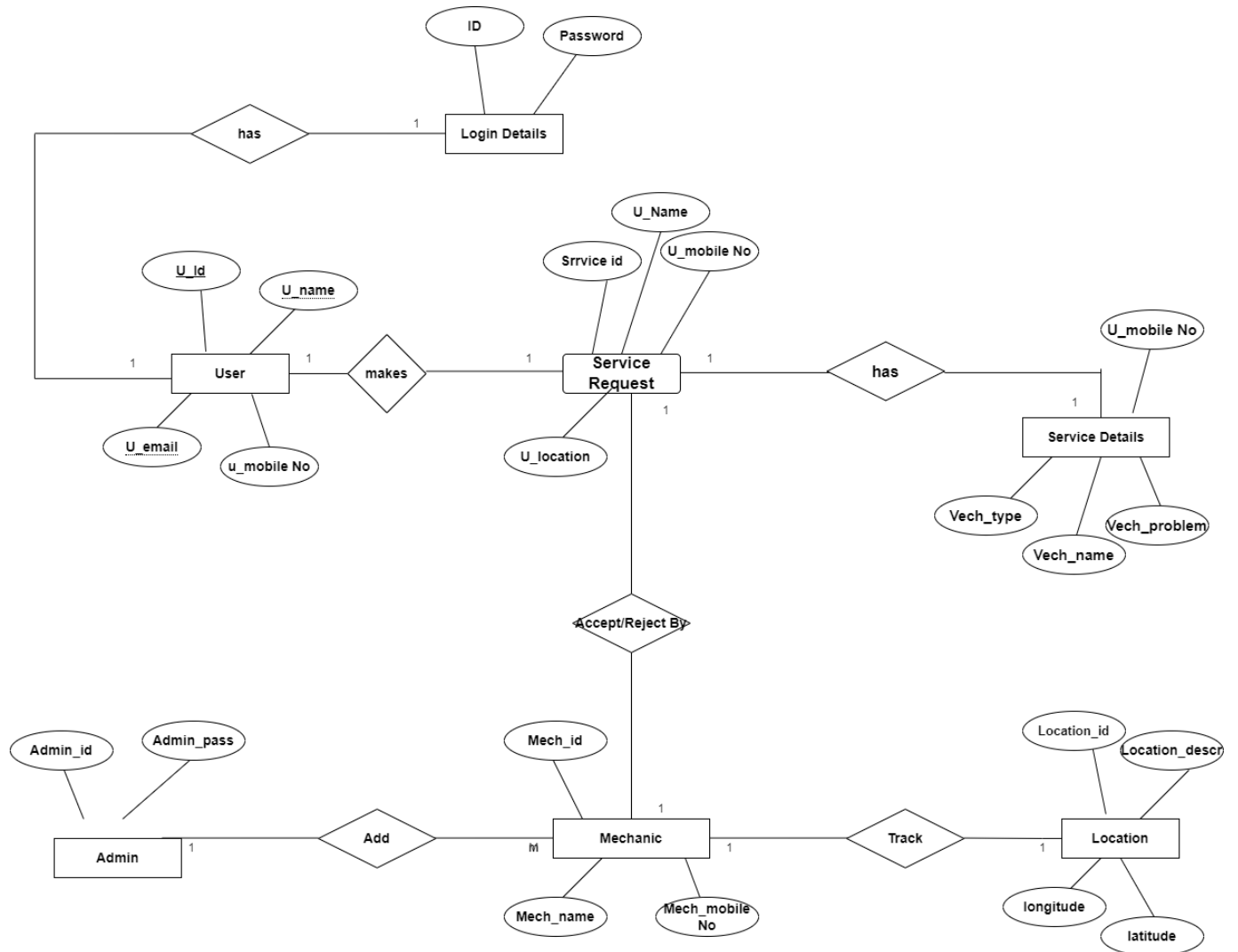
Flexibility

- Business rules

Achieving the functionality

All users need a valid email address

3.2 EntityRelationshipDiagram (ERD)



3.3 Table Structure

USER TABLE			
Field	Type	length	Description
<u>U_Id</u>	Number	10	Primary key
U_Name	String	20	Not Null
U_Email	String	20	Not Null
U_Mobile no.	Number	10	Not Null

LOGIN TABLE			
Field	Type	length	Description
User_Id	Number	10	Foreign key
User_Pass	String	20	Not Null

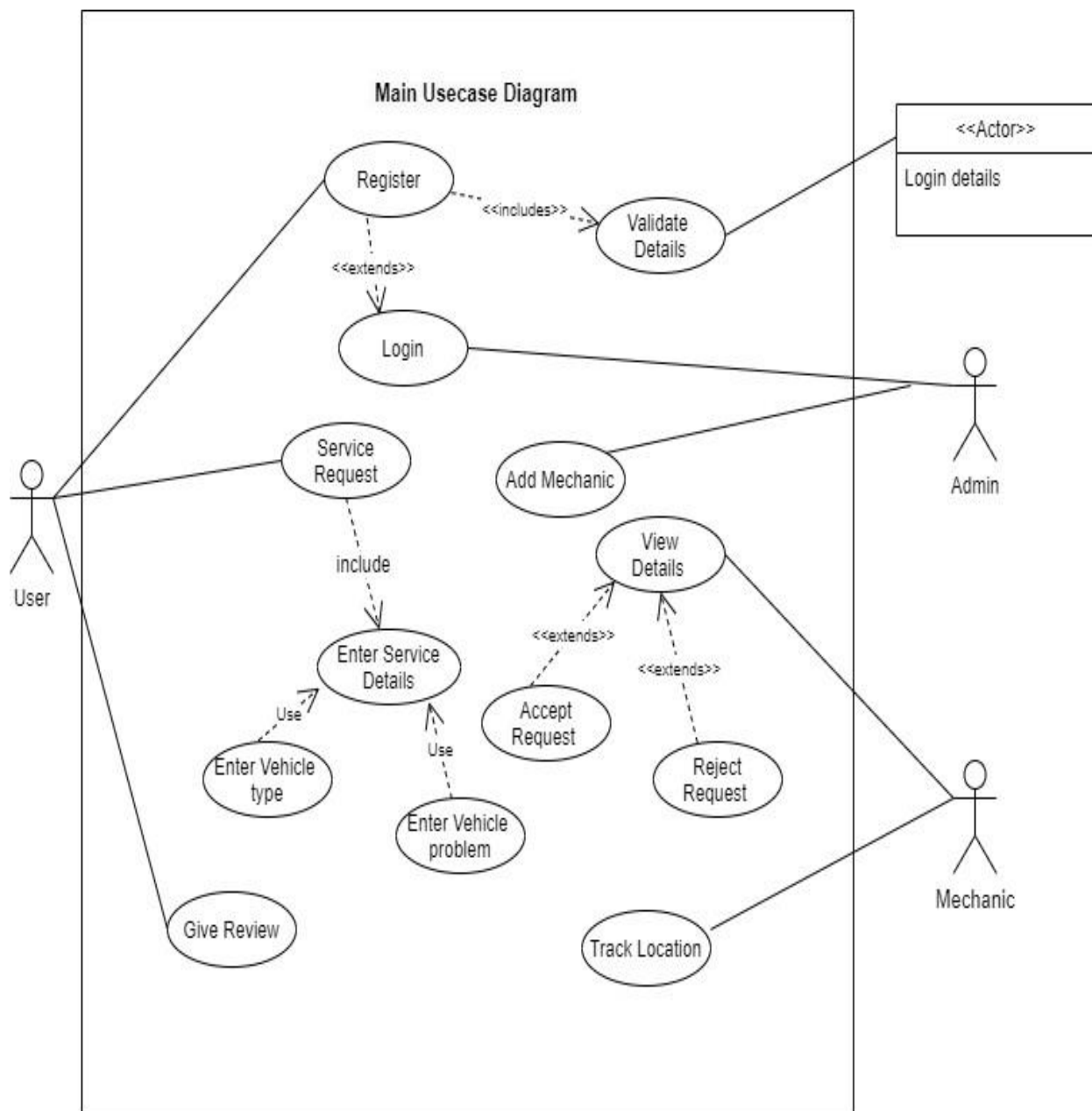
MECHANIC TABLE			
Field	Type	length	Description
<u>Mech_Id</u>	Number	10	Primary key
Mech_Name	String	20	Not Null
Mech _MobileNo.	Number	20	Not Null

SERVICE DETAILS TABLE			
Field	Type	length	Description
<u>Vehicle_problem</u>	String	20	Not Null
Vehicle_type	String	20	Not Null
Vehicle_name	String	20	Not Null
U_Mobile no.	Number	10	Not Null

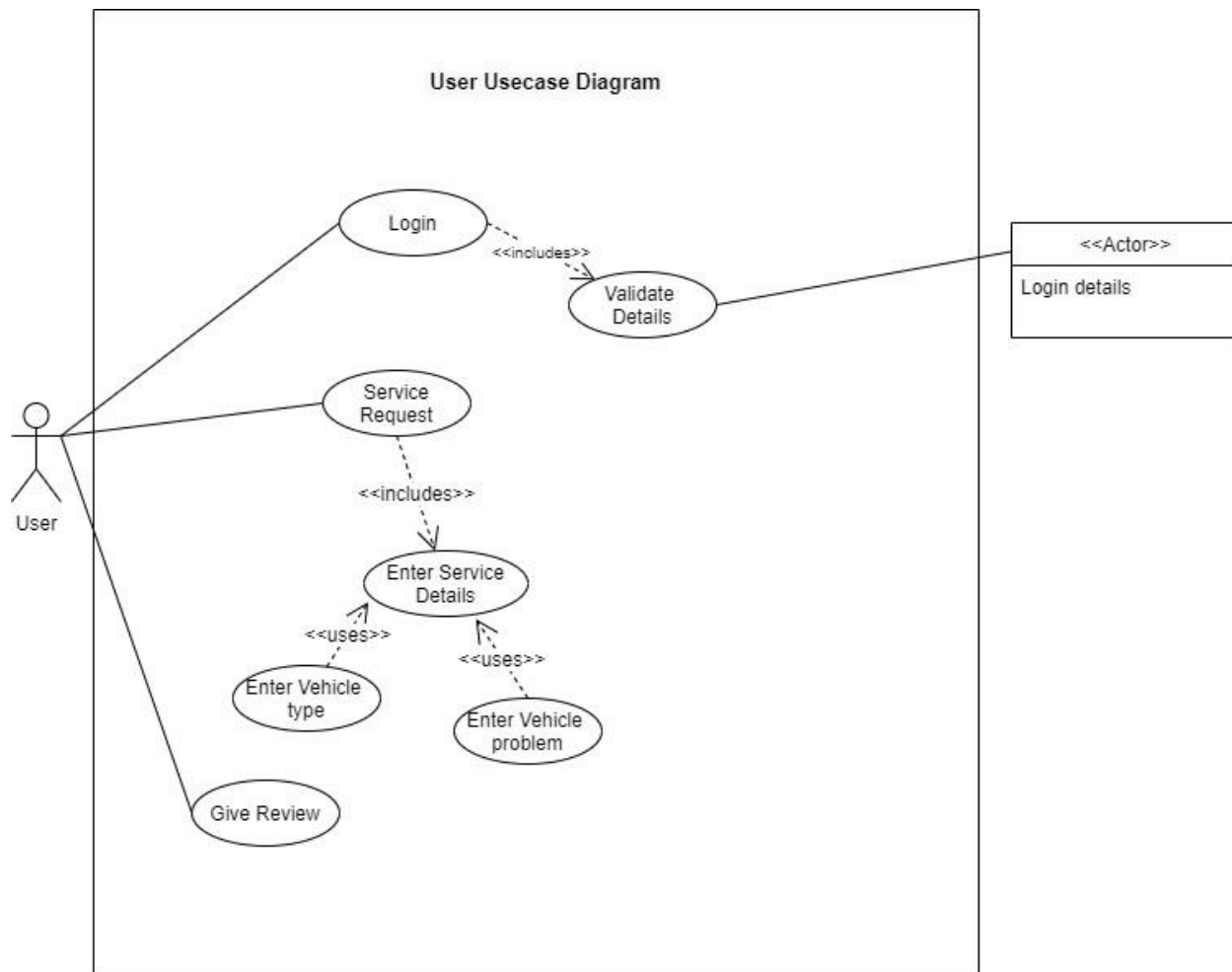
ADMIN TABLE			
Field	Type	length	Description
Admin_Id	Number	10	Primary
Admin_Pass	String	20	Not Null

SERVICE REQUEST TABLE			
Field	Type	length	Description
<u>Service_Id</u>	Number	10	Primary key
User_Name	String	20	Not Null
User_location	String	20	Not Null
User_email	String	20	Not Null

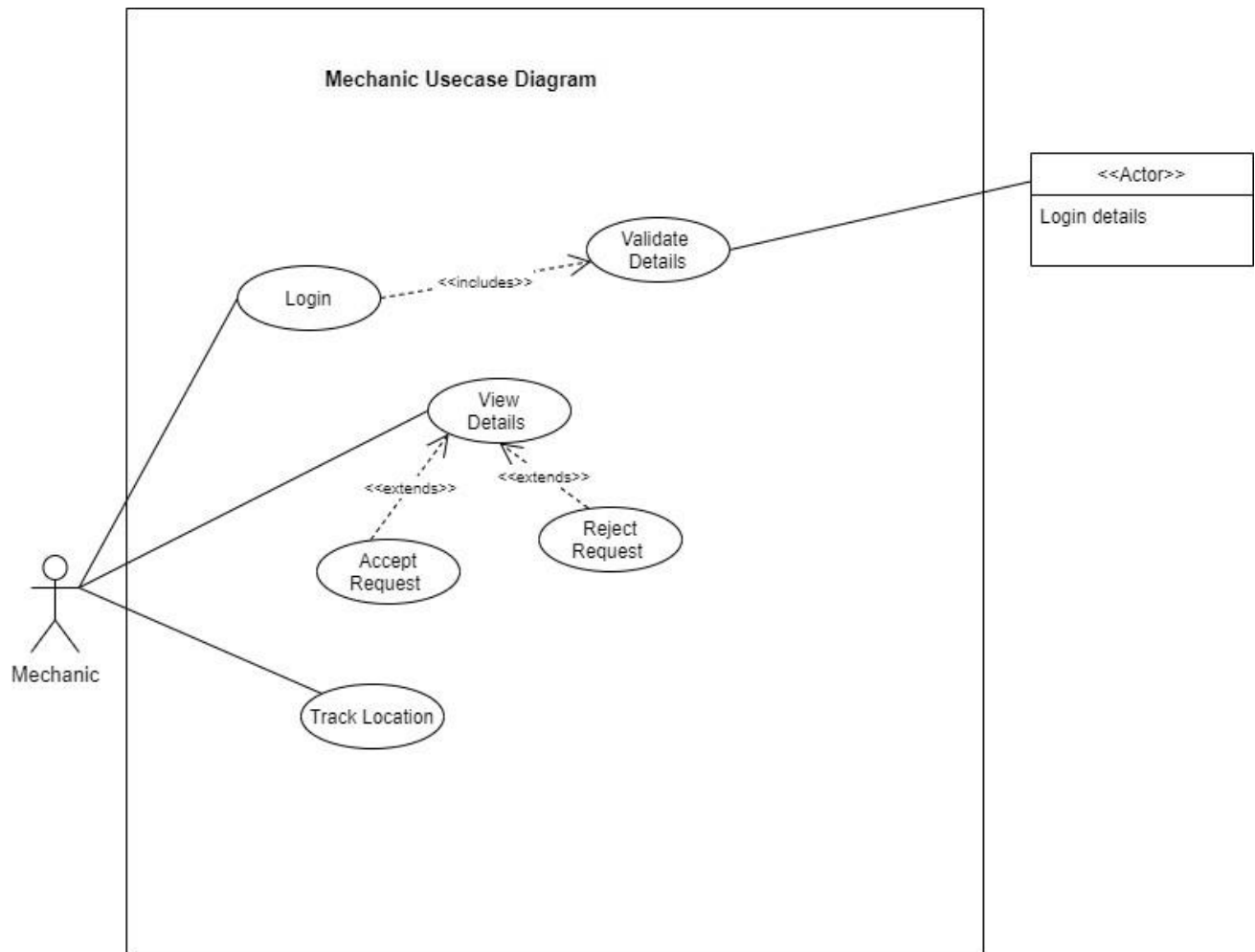
3.4 Usecase Diagrams



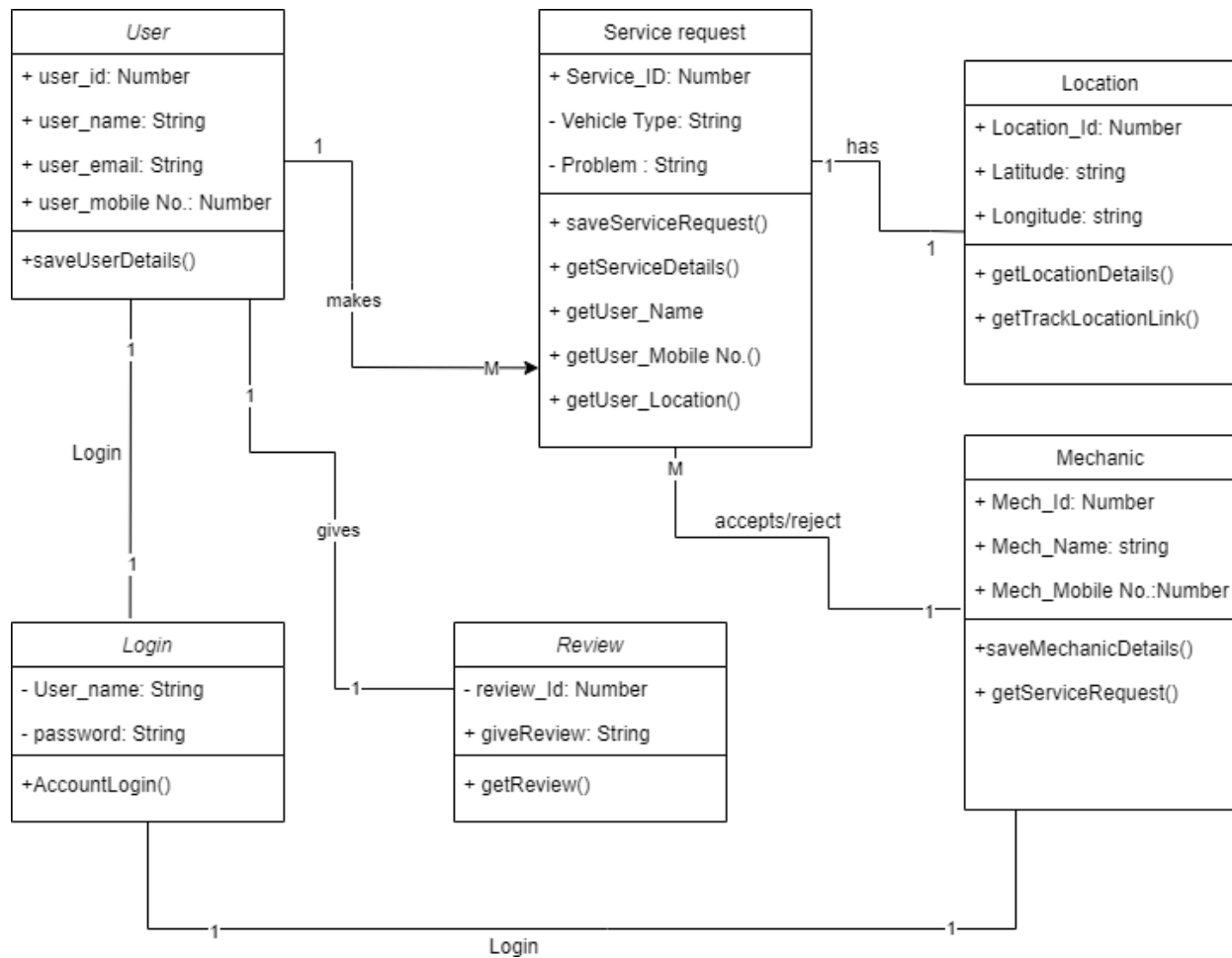
3.4.1 User Usecase Diagram



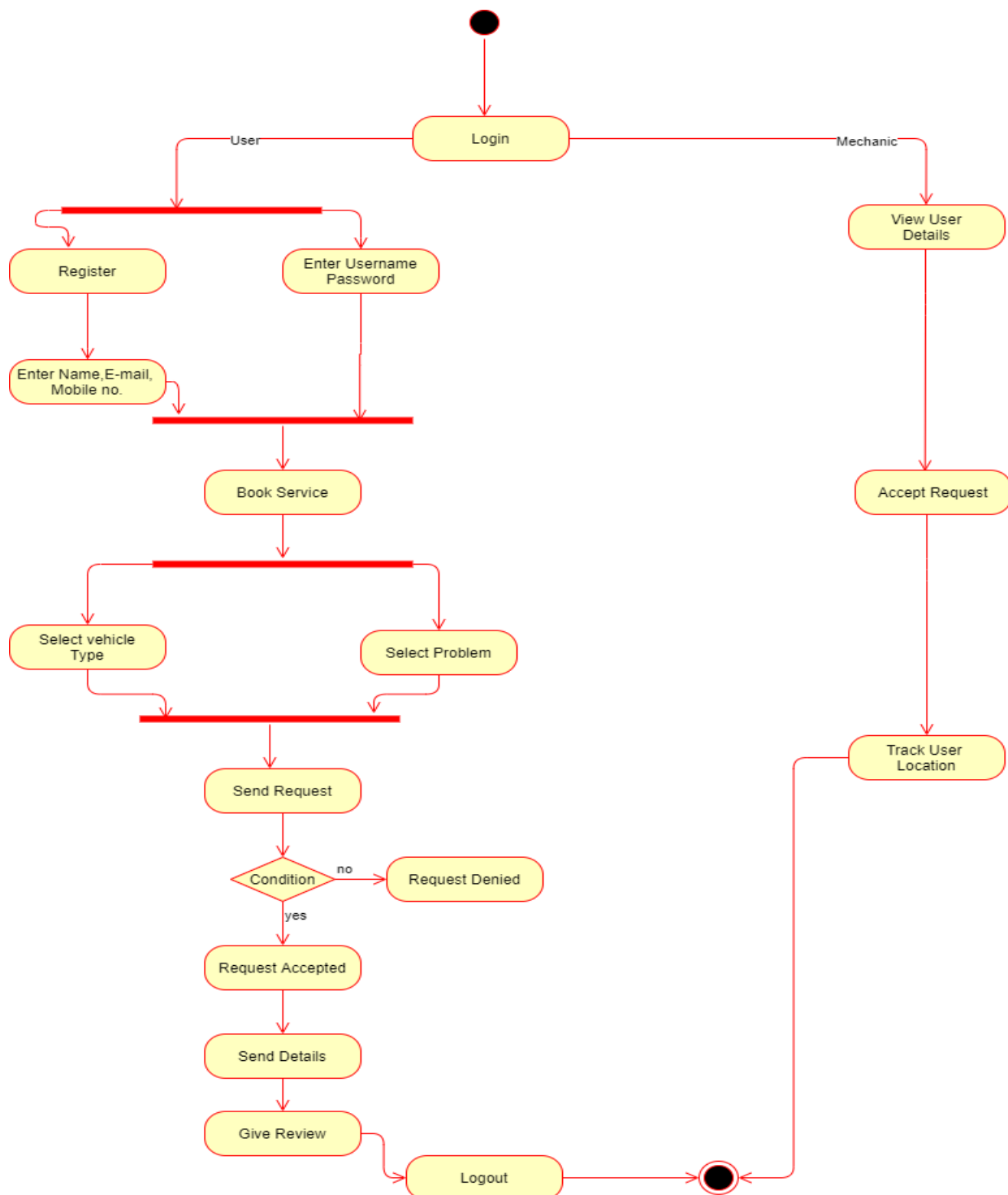
3.4.2 Mechanic Usecase Diagram



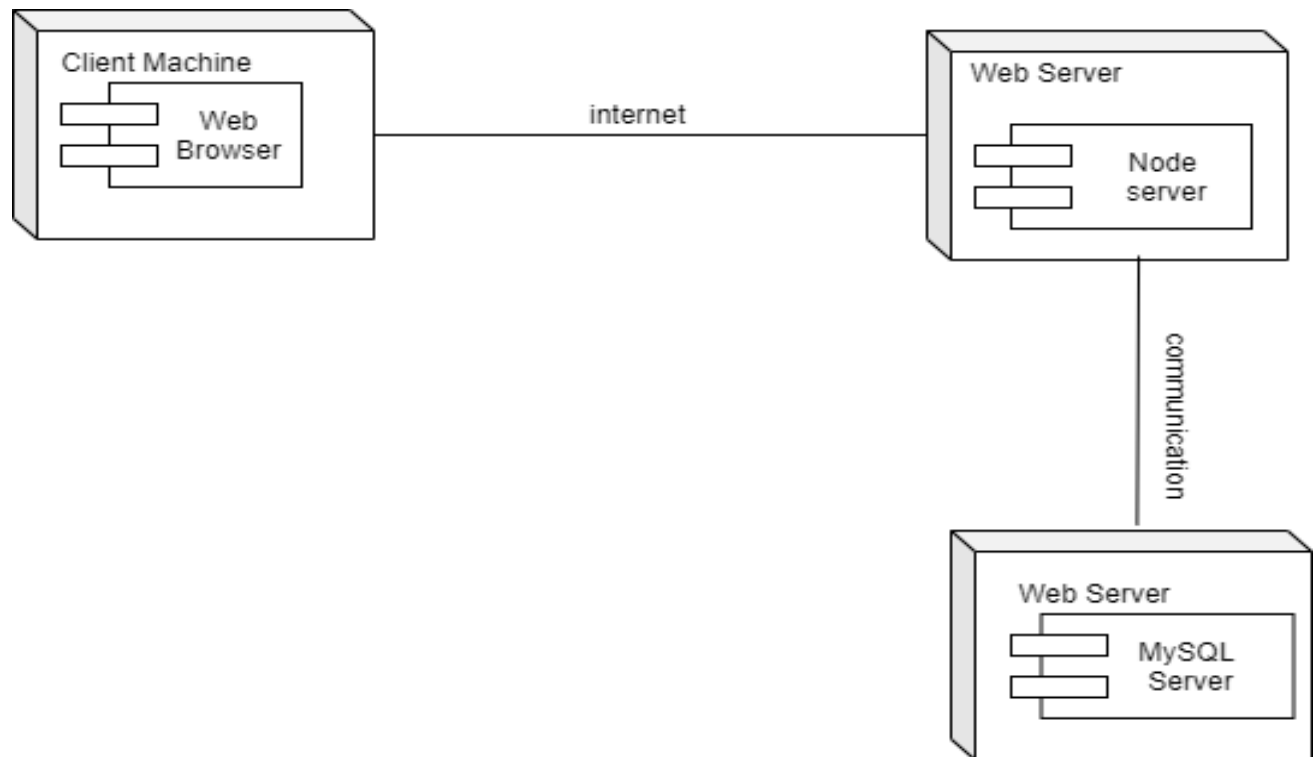
3.5 Class Diagram



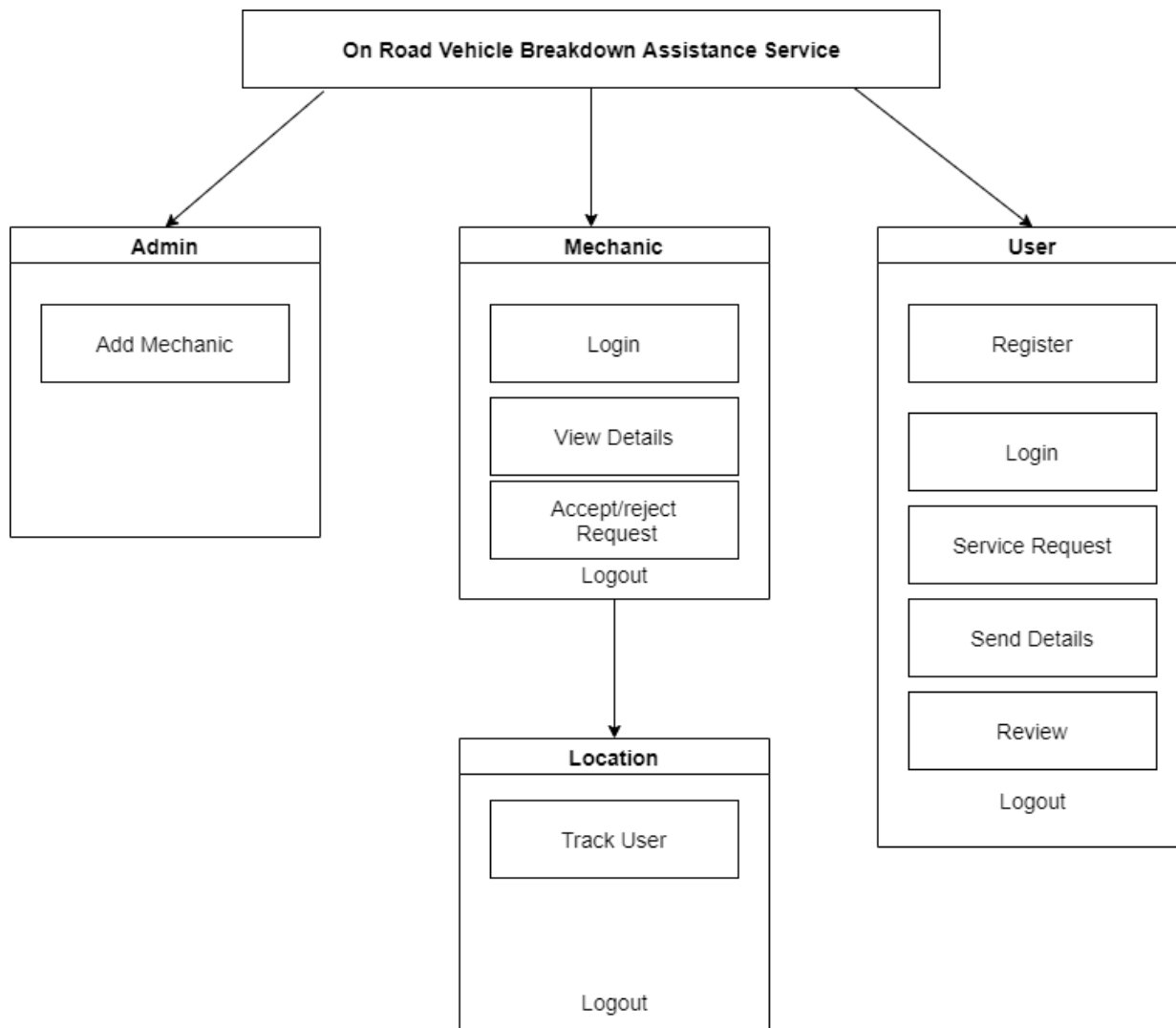
3.6 Activity Diagram



3.7 Deployment Diagram

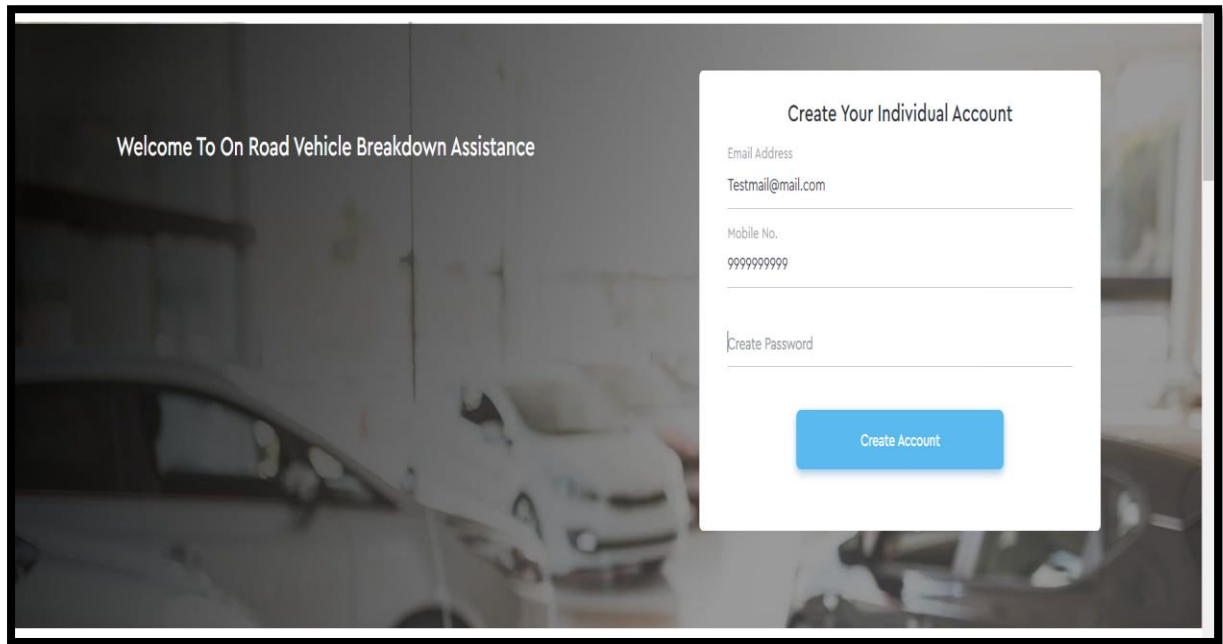


3.8 Module Hierarchy Diagram



3.9 Sample Input and Output Screens

1) Registration



The image shows a registration form titled "Create Your Individual Account" overlaid on a blurred background of a car. The form includes fields for "Email Address" (containing "Testmail@mail.com"), "Mobile No." (containing "9999999999"), and "Create Password". A blue "Create Account" button is at the bottom of the form. The background text "Welcome To On Road Vehicle Breakdown Assistance" is visible on the left.

Welcome To On Road Vehicle Breakdown Assistance

Create Your Individual Account

Email Address
Testmail@mail.com

Mobile No.
9999999999

Create Password

Create Account

2) Log In

Login to Your Account

Enter Email Address

Testmail@mail.com

Enter Password

Continue

Create Account Create Admin Account

3) Service Request Page

Service Request Details

Full Name: User1

Mobile Number: 7225083625

Email Address: Testmail@mail.com

Other Details

Vehicle type : ☐ BIKE ☒ CAR

Service Location: Chandrapur, Bazar Ward, Chandrapur, Maharashtra, India

Book Service

