



**Shri Vile Parle Kelavani Mandal's**  
**Usha Pravin Gandhi College of**  
**Arts, Science and Commerce**



**NAAC Accredited 'A' Grade**

# **ONLINE MECHANIC LOCATOR**

**A Project Report**

Submitted in partial fulfilment of the  
Requirements for the award of the Degree of

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)**

**By**

**Akash Verma**

**53003160110**

**Under the esteemed guidance of**

**Mr. Rajesh K Maurya**

**Assistant Professor**

**DEPARTMENT OF INFORMATION TECHNOLOGY**  
**USHA PRAVIN GANDHI COLLEGE OF ARTS, SCIENCE AND**  
**COMMERCE**

*(Affiliated to University of Mumbai)*

**MUMBAI, 400056**

**MAHARASHTRA**

**2018**

## PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

PNR No.: \_\_\_\_\_

Roll no: \_\_\_\_\_

1. Name of the Student

\_\_\_\_\_

2. Title of the Project

\_\_\_\_\_

3. Name of the Guide

\_\_\_\_\_

4. Teaching experience of the Guide \_\_\_\_\_

5. Is this your first submission?

Yes

☐

No

☐

Signature of the Student

Signature of the Guide

Date: \_\_\_\_\_

Date: \_\_\_\_\_

Signature of the Coordinator

Date: \_\_\_\_\_

**USHA PRAVIN GANDHI COLLEGE OF ARTS, SCIENCE AND  
COMMERCE**

*(Affiliated to University of Mumbai)*

**MUMBAI- MAHARASHTRA-400056**

**DEPARTMENT OF INFORMATION TECHNOLOGY**



**CERTIFICATE**

This is to certify that the project entitled, “**Online Mechanic Locator**”, is bonafide work of **Akash Verma** bearing Seat No:**53003160110** submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from the University of Mumbai.

**Internal Guide**

**Coordinator**

**External Examiner**

**Date:**

**College Seal**

# DECLARATION

I hereby declare that the project entitled, “**Online Mechanic Locator**”.  
done at **Mumbai**, has not been in any case duplicated to submit to any other university for the  
award of any degree. To the best of my knowledge other than me, no one has submitted to  
any other university.

The project is done in partial fulfilment of the requirements for the award of degree of  
**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final  
semester project as part of our curriculum.

**Name and signature of student**

## **ABSTRACT**

The current car repair system is broken and it has become a very frustrating experience for car owners. The car owners have to physically take their car at the garage for repairing and then have to wait for their car to be serviced by mechanic hoping that the car will go in the hands of a good mechanic. The mechanics working in those garages are underpaid when compared to the skill set they have to offer to the industry.

The aim of making this project is to change the current car servicing industry. Through this system people will have the ability to book a mechanic sitting at the comfort of their home and without having to go through the frustration of visiting to the garage to get their car fixed. Mechanics will get connected to various car owners through our system, where he can fix their problem and can get paid. The system also helps Garages to promote their product and services to mechanics and car owners respectively.

# ACKNOWLEDGEMENT

First and foremost I offer my sincerest gratitude to the principal and professors of my College who have supported me throughout my time here, given me valuable knowledge,

Moulded and shaped me into the person I am today.

I'd like to thank our project guide, Prof. Rajesh Maurya, for his patience and for sharing his expertise whilst giving me the space to work in my own way.

I cannot thank my family enough for bringing me up the way they did. The source behind my excellence is you.

I am thankful and fortunate to get constant encouragement, support and guidance from all the teaching staff of the **BSc. I.T. department of Usha Pravin Gandhi College,**

Which helped me in successfully completing my project work.

This list would be incomplete without mentioning all of the developers and education Institutes around the world that share their knowledge, work, and wisdom over the Internet.

**Akash Verma**

# TABLE OF CONTENT

INTRODUCTION .....	12
1.1 Background: .....	13
1.2 Objective: .....	14
1.3 Purpose, Scope and Applicability .....	14
1.3.1 Purpose: .....	14
1.3.2 Scope: .....	15
1.3.3 APPLICABILITY: .....	16
1.4 Achievements .....	16
1.5 Organization of Report .....	17
SURVEY OF TECHNOLOGIES .....	19
2.1 Mobile Application Development .....	19
2.1.1 Native App .....	19
2.1.2 Hybrid App .....	20
2.2 Web Application Development .....	22
2.2.1 Frontend Development .....	22
2.2.2 Backend Development .....	24
2.3 Database .....	26
REQUIREMENT AND ANALYSIS .....	32
3.1 Problem Definition .....	32
3.2 Requirements Specification .....	33
3.3 Planning and Scheduling .....	34
3.4 Software and Hardware Requirements .....	36
3.5 Preliminary Product Description .....	37
3.6 Conceptual Models .....	38
SYSTEM DESIGN .....	39
4.1 Basic Modules .....	39

4.2 Data Design.....	40
4.2.1 Schema Design.....	40
4.2.2 Data Integrity and Constraints .....	43
4.3 Procedural Design.....	48
4.3.1 Logical Diagram .....	48
4.4 User Interface Diagram.....	57
4.5 Security .....	60
4.6 Test Cases .....	61



## Table of Figures

Figure 1.1 Current Situation .....	13
Figure 2.1 JS Framework Comparison .....	24
Figure 3.1 Gantt Chart .....	35
Figure 3.2 ER Diagram .....	38
Figure 3.1 Gantt Chart .....	35
Figure 3.2 ER Diagram .....	38
Figure 4.1 Owner App Use Case .....	48
Figure 4.2 Garage App Use Case.....	49
Figure 4.3 Mechanic App Use Case.....	50
Figure 4.4 Owner Sequence Diagram .....	51
Figure 4.5 Garage Sequence Diagram .....	52
Figure 4.6 Mechanic Sequence Diagram .....	53
Figure 4.7 Garage Activity Diagram.....	54
Figure 4.8 Mechanic Activity Diagram .....	55
Figure 4.9 Owner Activity Diagram .....	56
Figure 4.10 GUI.....	59

## List of Tables

Table 3.1 MySQL VS MongoDB .....	31
Table 3.2 Pros and Cons of SQL and MongoDB.....	31
Table 3.3 Software Requirement .....	36
Table 3.4 Hardware Requirement .....	37
Table 4.1 Test Cases .....	62

## **Online Mechanic Locator**

# **Chapter 1**

## **INTRODUCTION**

We all love our Cars and we want them to be in the same new condition as they came from factory. This sounds good but to make it happen, we require a regular servicing of our Car. You may have encountered a situation wherein you are in need of a mechanic to fix your car but you don't want to give it to a random local mechanic whom you don't know and you're skeptical about his skills or the amount of money he is charging.

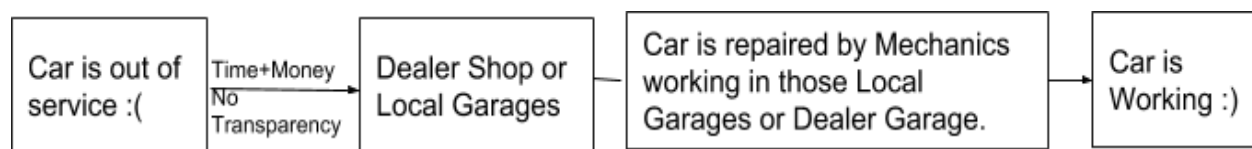
Once your car goes out of warranty you must have faced the hassle of going to your Dealer's Garage for servicing your Car where they will definitely charge you more money than what they used to charge you before when the car was still in warranty. You also have to wait for days or sometimes weeks to get your car back, because they have a large pool of car lined in queue to be repaired and you don't know how long it will take for your car's turn to come for repair.

The system which we are building will make Car Owner's life easy, making car repair and maintenance affordable, convenient, and transparent. Car Owners can find best offers and deals from various different Garages. The system allows Mechanics to showcase their skills and will get online presence where they can solve problem faced by Car Owners and can reach to a larger audience. Hence, making a good sum of money through our system. Our system also allows Garages to increase their sales by showing their services & offers to Car Owners and motor parts listing to Mechanics, directly through the system.

## 1.1 Background:

Dealing with car problems is not only a time-consuming and frustrating experience, it is often expensive. Car owners often lack visibility regarding the quality of mechanics or fairness of the price. At the same time, the mechanics who put in the hard work fixing cars make very little money at repair shops (less than 20% of what consumers pay), and rarely get the recognition they deserve.

The current situation is depicted in Figure 1 below:-



*Figure 1.1 Current Situation*

As you can see in the diagram that when car owners go to the repair shop (Dealer Shop or Local Garage), they don't have any metrics to compare the prices for what they've been charged for. It is also a very time-consuming process and you also have to trust your dealer that whether he is installing genuine motor parts or not. Because it has happened in many cases that dealer install faulty auto parts in your car which will work for some months and then it will again fail so you have to come again for repairing and again the whole cycle goes on and on.

Mechanics who are skilled at their work don't get the recognition they deserve. They often work under some garage where they are paid less.

I believe that the current auto repair system is broken and needs a new system where Car Owners will have better experience and it will empower mechanics to live a better life. It will help garages increase their sales by selling their products and services to the appropriate customer through our system. The system will connect Mechanics, Car Owners and Garages at a single place where they can work together to solve the problem mentioned above.

## 1.2 Objective:

- To help Car Owners find best deals and services from Mechanics and Garages.
- To empower mechanics to live a better life by making his online presence where he can reach to a larger audience and get paid well by fixing the car issue.
- To make Car Owner's life easy by removing all the hassles of going to the service center for repairing the car.
- To boost the sales of products and services of Garages through the system.
- The car owner has the option to select services provided by the specific Garage or specific Mechanic.

## 1.3 Purpose, Scope and Applicability

### 1.3.1 Purpose:

The title itself gives an indirect clue about this project. "Online Mechanic Finder" you can find Mechanics Online to repair your Cars. The project is being executed because there is a need of a system which will allow the users to freely communicate and ask for certain services from the Mechanics.

The Sole purpose is to provide Users a medium through which they can easily, efficiently, securely and comfortably seek help for Car services in a very interactive and friendly User-Interface environment.

Firstly, the Car Owner creates his/her account in the application where he/she will be asked for details about their Cars. When the user experiences any problem in their car, instead of going to a Garage the user can find Mechanics on our system and can chat with him. The Owner can also broadcast his car problem so that it will reach to every mechanic in the system and whoever is ready to solve the Owner's Problem can directly chat with Owner and thus can tell his proposed price quote to the Owner. In this way Owner will receive many price quote by many different

mechanics and Owner can choose between mechanics based on their reviews, ratings, past work history and his proposed price quote.

Secondly, the Mechanic himself has a profile on the system through which they can showcase their skills and can reach to a massive audience, providing their services and thus can earn a good amount of money. They can also book slots in Garage for repair space or can see the garages product (i.e. Auto parts) listing and can contact or chat with Garage for bulk orders at discounted price. Mechanic can chat with Owners through Online Chatting feature provided in the system which is secured and responsive.

Through this system Garage can increase their sales by showcasing their products(Auto parts) and services(towing, repair slots, etc) to Mechanics and by attracting Car Owners through lucrative offers like Free Car wash, Free Inspection and Diagnostic, etc.

### 1.3.2 Scope:

- Storing information of Owners, Mechanics & Garages.
- Check validity of information provided by Owners and Mechanics.
- Giving the ability to chat within the system through an interactive and user friendly chatting portal.
- Storing the reviews and ratings provided by car owners.
- Storing the information of garages like offers and products.
- Giving the ability to mechanics to earn through the system by connecting to Owners and Garages.
- Storing owners problem in database

### 1.3.3 APPLICABILITY:

- The System help to connect Car Owners, Mechanics and Garages, where these entities can work together for the benefit of each other.
- This application registers Customer, Mechanics and Garages profiles.
- It has a Mechanic Profile & Garage Profile listing which makes it convenient for the Owner to select and appoint a Mechanic or select a Garage service.
- Different type of Vehicle services are provided to the user.
- Owner can book a slot for his/her vehicle in a specific garage for repair.
- Mechanics can contact garage for purchasing automobile parts or book slot for repairing the car.
- Garage has the ability to share his product listing and services to mechanics and owners respectively. Thus, increasing the sales.

## 1.4 Achievements

This project has helped me in understanding deeply about the concept of how HTTP works and why it is an important protocol. I have learned about Web Services and how they are responsible for sharing data from remote server to application. ReST API is what makes it possible to transfer data to Android Application in JSON format which can be then used to display the data on your application.

I have learned about various tools and technology that can be used to make Android Application in much faster and efficient way. I have learned about various JavaScript and web frameworks which are amazing and can be used to make cross platform application. I have also learned about various design tool to design mockup and prototype at early stage which helps to show the flow of the application.

When I was researching about database i came across Firebase which is NoSQL database and the marketing done by Google Team has initially made me believe that it is the best option available to make application. Though, i agree that firebase is amazing and it has many great features but that is just one side of story. The other side is SQL databases, it's been around from over 30 years



and the features it offers is also amazing. This confusion has given boost to my curiosity to know the difference between the two and I have documented the differences in chapter 2.

I have also developed a good understanding of how to collect important and relevant data from google. This project has helped me in developing my research ability.

I understood the importance of Software Engineering and why it is important to properly plan your project before implementing it. It will save you a bunch of time and conceptual model helps in communicating with other stakeholders of the project.

I believe that there is many more thing to learn and explore that will help me in making this system more efficient and I will surely learn about them when i am at implementation stage.

## 1.5 Organization of Report

1<sup>st</sup> chapter provides a brief introduction about the project, its objectives and goals accomplished. It also focuses on the scope and purpose of the project and how it can be applied in the real world.

2<sup>nd</sup> chapter focuses on the technologies that can be used for making this system. It shows that how mobile application can be developed in much faster and efficient way by making use of cross platform application building tool like Cordova. Later in this chapter we see various other JavaScript libraries and framework that can be used for making UI components that are fully responsive. It also explain in brief the most heated debate among the developer community i.e. SQL vs NoSQL.

3<sup>rd</sup> chapter describes the problem definition of the traditional Auto Repair System. It also outlines the drawbacks of the existing system that lead to requirements to eliminate the problems of the old system. It also covers the requirements of the software and hardware components that I will be using to make this project. Also, the way I have planned to complete this project in stipulated amount of time will be shown with the help of Gantt chart. The Gantt Chart describes how I have scheduled my whole project completion.

In the end of this chapter, I have shown the initial conceptual representation of how the data will be stored and what is the relation among different entities with of help of ER diagram.

4<sup>th</sup> chapter focuses on defining the modules which will be there in the system. It then shows the Schema Design and Constraint which the data has in the system. It will also focus the way data is being used and handled, sent and received, updated and deleted. It will also show a basic user interface that will help to get an idea of how the system will look like to the user. It also discusses security concerns regarding the project and also the testing phases that will help to eliminate errors and mistakes at the initial stages of project implementation that will try to reduce the error in least amount of time and efforts.

## Chapter 2

# SURVEY OF TECHNOLOGIES

Molding an Idea into Reality requires a good code. Code is that ingredient that helps in shaping your idea and bringing it to reality by helping in developing a good system. For making something which is not present in the market, it requires good and deep understanding of the problem. If the problem is properly understood than we can search for the technologies which can be used to solve that problem in faster and much efficient way.

Below are the survey of some technologies which can be used to make the system.

## 2.1 Mobile Application Development

### 2.1.1 Native App

Native application is a software or program which has been developed to perform some specific task on particular environment or platform. Native application built using software development tools (SDK) for a certain software framework, hardware platform or operating system. Like Android app built using Java Development Kit on Java platform, iOS app built using iOS SDK, Swift and Objective C. Similarly, .NET required for Windows platform.

#### **Advantages:**

- Graphical Applications, HD games, intensive animation applications might perform well as native app because Native code is still faster than HTML and JavaScript. WebGL standards helps browser and hybrid app for gaming apps to meet performance but still native has edge.
- Native SDKs allows to access device features without dealing with complexity of native plugins and new device features will be available out of the box along with SDKs.
- Not much dependencies on open source libraries and platforms like Cordova and Ionic

#### **Disadvantages:**

- Separate development effort for each platform which increases the development time.

- Each platform code will have its own release cycle and updates which adds to development time and cost.
- Releasing same feature on all platform at same time always challenging because of different code base.

Different skill set required to develop and maintain the same application on each platform which adds to the cost.

### 2.1.2 Hybrid App

Hybrid apps are native apps only because it can be downloaded from platform's app store like native app. It can get access to all the native platform features. It can have performance close to native app. The major differences are listed below:

- Hybrid apps are built using web technologies like HTML, CSS and JavaScript whereas Native apps built with specific technology and language for specific platform like Java for Android, Swift for iOS.
- Hybrid app runs in webView (A view that displays web pages, uses the same engine of browser but no browser like widgets)
- Native plugins required to access the native features of the platform like camera, mic etc. (Native plugins are like wrapper on top of native libraries or components)
- Hybrid app can be built for any platform from single code base.

#### **Libraries and Frameworks:**

**Cordova** is an open-source mobile development framework. Cordova Plugin helps to access device features. It allows to use standard web technologies for cross-platform development. Applications execute within wrappers targeted to each platform.

**Ionic** is the app platform for web developers. We can build amazing mobile, web, and desktop apps all with one shared code base and open web standards. It uses Cordova behind the scene.

Other cross platform app development frameworks are listed below:

- Framework7
- Titanium Appcelerator
- OnsenUI
- Xamarin (based on C#)

**Advantages:**

- Single code base for all platforms means write once and run anywhere but for native app scenario, we need to build and maintain separate app and code for each platform.
- Same development team can deliver app for any platform including website as well because all required is web technologies.
- Hybrid App is based on web technologies, so same app can be run on browser like any other website or can be run as Progressive Web App(PWA).
- Hybrid app can achieve the same hardware-based performance acceleration as native app.
- Hybrid app can have same and consistent user experience across platform regardless of user moves between different devices or browser.

**Disadvantage:**

- For most applications, performance is same as native app but 3D, HD games, high graphics-oriented apps and other performance centric apps, hybrid approach might not go well.
- Hybrid app can able access all the native device features like touchId, media etc. but dependent on native plugins. Sometime entire new device feature might be not being readily available as native plugin. We can write our own but it adds complexity to the development.

Hybrid app is having dependencies on different libraries and frameworks like Cordova, Ionic which have to be in sync with latest platform version changes and releases.

## 2.2 Web Application Development

### 2.2.1 Frontend Development

#### **HTML**

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

#### **CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css

file, and reduce complexity and repetition in the structural content.

#### **JavaScript**

JavaScript often abbreviated as JS, is a high-level, interpreted Programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

## **Frameworks and Libraries:-**

### **Bootstrap**

Twitter Bootstrap is the most popular front end framework in the recent time. It is sleek, Intuitive, and powerful mobile first front-end framework for faster and easier web development. It uses HTML, CSS and JavaScript. Bootstrap is a free and open-source front-end framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

### **AngularJS**

- Angular is superheroic JavaScript MVVM(Model-View-ViewModel) framework, founded in 2009, which is awesome for building highly interactive web applications.
- Companies that use Angular : Upwork, Freelancer, Udemy, YouTube, Paypal, Nike, Google, Telegram, Weather, iStockphoto, AWS, Crunchbase.

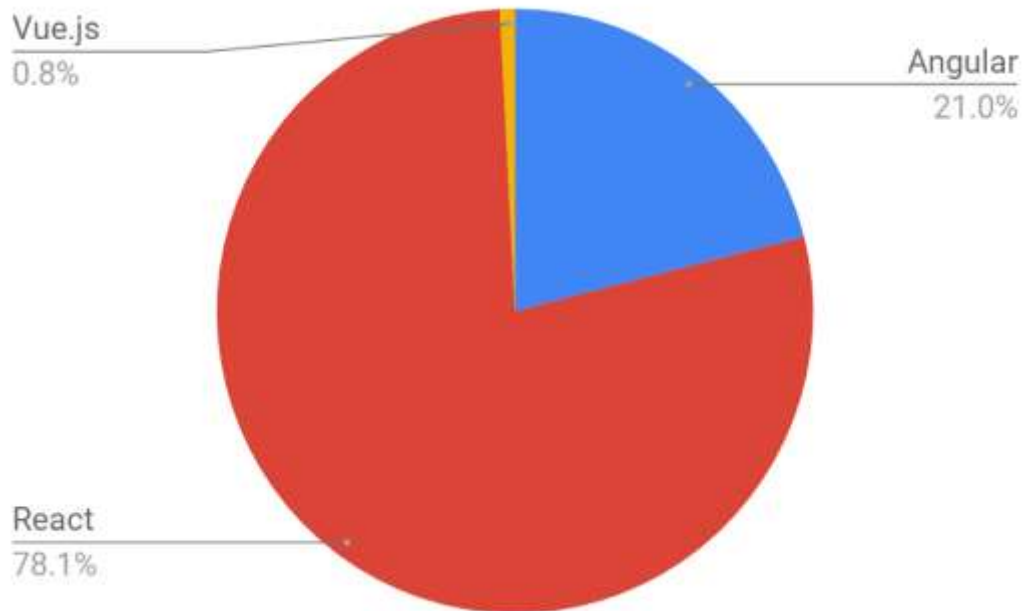
### **ReactJS**

- ReactJS is a JavaScript library, open sourced by Facebook in 2013, which is great for building  
Huge web applications where data is changeable on a regular basis.
- Companies that use ReactJS: Facebook, Instagram, Netflix, New York Times, Yahoo, Khan Academy, Whatsapp, Codecademy, Dropbox, Airbnb, Asana, Atlassian, Intercom, Microsoft.

### **Vue.JS**

- Vue.js is a JavaScript framework, launched in 2013, which perfectly fits for creating highly adaptable user interfaces and sophisticated Single-page applications.
- Companies that use Vue.js: Xiaomi, Alibaba, WizzAir, EuroNews, Grammarly, Gitlab and Laracasts, Adobe, Behance, Codeship, Reuters.

JavaScript frameworks are developing at an extremely fast pace, meaning that today we have frequently updated versions of Angular, ReactJS and another player on this market—Vue.js. We analyzed the number of open positions worldwide that require a specific knowledge of a certain framework. As a source, we took Indeed.com and got the distribution (below) according to more than 60,000 job offers.



*Figure 2.1 JS Framework Comparison*

## 2.2.2 Backend Development

### **PHP (Hypertext PreProcessor)**

It is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve. The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.



## **Features of PHP**

It is most popular and frequently used world wide scripting language, the main reason of popularity is; It is open source and very simple.

- Simple
- Faster
- Interpreted
- Open Source
- Case Sensitive
- Simplicity
- Platform Independent
- Security
- Loosely Typed Language

## **Laravel**

**Laravel** is a free, open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller (MVC)

Architectural pattern and based on Symfony.

## **Python**

Python is a beautiful language. It's easy to learn and fun, and its syntax (the rules) is clear and concise. Python is a popular choice for beginners, yet still powerful enough to back some of the world's most popular products and applications from companies like NASA, Google, IBM, Cisco, Microsoft, Industrial Light & Magic among others. One area where Python shines is web development. Python offers many frameworks from which to choose from including bottle.py, Flask, CherryPy, Pyramid, Django and web2py. These frameworks have been used to power some of the world's most popular sites such as Spotify, Mozilla, Reddit, the Washington Post and Yelp.

## **Django**

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

- Ridiculously fast - Django was designed to help developers take applications from concept to completion as quickly as possible.
- Reassuringly secure - Django takes security seriously and helps developers avoid many common security mistakes.

Exceedingly scalable - Some of the busiest sites on the Web leverage Django's ability to quickly and flexibly scale.

## **2.3 Database**

When it comes to choosing a database, one of the biggest decisions is picking a relational (SQL) or non-relational (NoSQL) data structure. While both are viable options, there are certain key differences between the two that must be kept in mind when making a decision.

Here, I break down the most important distinctions and discuss two of the key players in the relational vs non-relational debate: MySQL and MongoDB.

### **The Big Picture Differences**

#### **The Language**

Think of a town - we'll call it Town A - where everyone speaks the same language. All of the businesses are built around it, every form of communication uses it - in short, it's the only way that the residents understand and interact with the world around them. Changing that language in one place would be confusing and disruptive for everyone.

Now, think of another town, Town B, where every home can speak a different language. Everyone interacts with the world differently, and there's no "universal" understanding or set organization. If one home is different, it doesn't affect anyone else at all.

This helps illustrate one of the fundamental differences between SQL relational and NoSQL non-relational databases, and this distinction has big implications. Let's explain:

**SQL databases** use structured query language (SQL) for defining and manipulating data. On one hand, this is extremely powerful: SQL is one of the most versatile and widely-used options available, making it a safe choice and especially great for complex queries. On the other hand, it can be restrictive. SQL requires that you use predefined schemas to determine the structure of your data before you work with it. In addition, all of your data must follow the same structure. This can require significant up-front preparation, and, as with Town A, it can mean that a change in the structure would be both difficult and disruptive to your whole system.

A **NoSQL database**, on the other hand, has dynamic schema for unstructured data, and data is stored in many ways: it can be column-oriented, document-oriented, graph-based or organized as a Key-Value store. This flexibility means that:

- You can create documents without having to first define their structure
- Each document can have its own unique structure
- The syntax can vary from database to database, and
- You can add fields as you go.

## **The Scalability**

In most situations, SQL databases are vertically scalable, which means that you can increase the load on a single server by increasing things like CPU, RAM or SSD. NoSQL databases, on the other hand, are horizontally scalable. This means that you handle more traffic by sharing, or adding more servers in your NoSQL database. It's like adding more floors to the same building versus adding more buildings to the neighborhood. The latter can ultimately become larger and more powerful, making NoSQL databases the preferred choice for large or ever-changing data sets.

## The Structure

SQL databases are table-based, while NoSQL databases are either document-based, key-value pairs, graph databases or wide-column stores. This makes relational SQL databases a better option for applications that require multi-row transactions - such as an accounting system - or for legacy systems that were built for a relational structure.

Some examples of SQL databases include MySQL, Oracle, PostgreSQL, and Microsoft SQL Server. NoSQL database examples include MongoDB, Firebase, BigTable, Redis, RavenDB, Cassandra, HBase, Neo4j and CouchDB.

### SQL vs NoSQL: MySQL vs MongoDB

Now that we've established the key structural differences between SQL and NoSQL databases, let's delve into the key functional differences between the two, looking specifically at MySQL and MongoDB as examples.

### **MySQL: The SQL Relational Database**

The following are some MySQL benefits and strengths:

- **Maturity:** MySQL is an extremely established database, meaning that there's a huge community, extensive testing and quite a bit of stability.
- **Compatibility:** MySQL is available for all major platforms, including Linux, Windows, Mac, BSD and Solaris. It also has connectors to languages like Node.js, Ruby, C#, C++, Java, Perl, Python and PHP, meaning that it's not limited to SQL query language.
- **Cost-effective:** The database is open source and free.
- **Replicable:** The MySQL database can be replicated across multiple nodes, meaning that the workload can be reduced and the scalability and availability of the application can be increased.
- **Sharding:** While sharding cannot be done on most SQL databases, it can be done on MySQL servers. This is both cost-effective and good for business.

## MongoDB : The NoSQL Non-Relational Database

The following are some of MongoDB benefits and strengths:

- **Dynamic schema:** As mentioned, this gives you flexibility to change your data schema without modifying any of your existing data.
- **Scalability:** MongoDB is horizontally scalable, which helps reduce the workload and scale your business with ease.
- **Manageability:** The database doesn't require a database administrator. Since it is fairly user-friendly in this way, it can be used by both developers and administrators.
- **Speed:** It's high-performing for simple queries.
- **Flexibility:** You can add new columns or fields on MongoDB without affecting existing rows or application performance.

### Side-by-Side Comparison of MySQL and MongoDB:

	MySQL	MongoDB
Written in	C++, C	C++, C and JavaScript
Type	RDBMS	Document-oriented
Main points	<ul style="list-style-type: none"><li>● Table</li><li>● Row</li><li>● Column</li></ul>	<ul style="list-style-type: none"><li>● Collection</li><li>● Document</li><li>● Field</li></ul>
Schemas	Strict	Dynamic

Scaling	Vertically	Horizontally
<b>Key features</b>	<ul style="list-style-type: none"> <li>● Full-text searching and indexing</li> <li>● Integrated replication support</li> <li>● Triggers</li> <li>● SubSELECTs</li> <li>● Query caching</li> <li>● SSL support</li> <li>● Unicode support</li> <li>● Different storage engines with various performance characteristics</li> </ul>	<ul style="list-style-type: none"> <li>● Auto-sharding</li> <li>● Native replication</li> <li>● In-memory speed</li> <li>● Embedded data models support</li> <li>● Comprehensive secondary indexes</li> <li>● Rich query language support</li> <li>● Various storage engines support</li> </ul>
<b>Best used for</b>	<ul style="list-style-type: none"> <li>● Data structure fits for tables and rows</li> <li>● Strong dependence on multi-row transactions</li> <li>● Frequent updates and modifications of large volume of records</li> <li>● Relatively small datasets</li> </ul>	<ul style="list-style-type: none"> <li>● High write loads</li> <li>● Unstable schema</li> <li>● Your DB is set to grow big</li> <li>● Data is location based</li> <li>● HA (high availability) in unstable environment is required</li> <li>● No database administrators (DBAs)</li> </ul>
<b>Examples</b>	NASA, US Navy, Bank of Finland, UCR, Walmart, Sony, S2 Security	Expedia, Bosch, Otto, eBay, Gap, Forbes, Foursquare, Adobe, Intuit,

	Corporation, Telenor, Italtel, iStock, Uber, Zappos, Booking.com, Twitter, Facebook, others.	Metlife, BuzzFeed, Crittercism, CitiGroup, the City of Chicago, others.
--	--	---

*Table 3.1 MySQL VS MongoDB*

**Pros and Cons Table:**

MySQL pros	MongoDB pros
<ul style="list-style-type: none"> <li>• Atomic transactions support</li> <li>• JOIN support</li> <li>• Mature solution</li> <li>• Privilege and password security system</li> </ul>	<ul style="list-style-type: none"> <li>• Document validation</li> <li>• Integrated storage engines</li> <li>• Shortened time between primary failure and recovery</li> </ul>
MySQL cons	MongoDB cons
<ul style="list-style-type: none"> <li>• Tough scaling</li> <li>• Stability concerns</li> <li>• Isn't community-driven development</li> </ul>	<ul style="list-style-type: none"> <li>• Not the best option for apps with complex transactions</li> <li>• Not a snap-in replacement for legacy solutions</li> <li>• Young solution</li> </ul>

*Table 3.2 Pros and Cons of SQL and MongoDB*

# **Chapter 3**

## **REQUIREMENT AND ANALYSIS**

### **3.1 Problem Definition**

When the car is in warrantee, the servicing is usually provided by Manufacturer's Official Repair Centre but when the car is out of warranty and we want to service our cars, we all have faced the problems like waiting in a long queue for your turn to come for repairing, there's lot of hassle in dealing with the Dealer who you know is charging you more money than what he is supposed to be charging. You will eventually end up paying more than the actual work done on the car. In today's world where time plays an important role in everyone's life, it is very hard to go to dealer shop for car servicing because of your busy schedule. You also don't know anything about your mechanics like his experience, his ratings and reviews because there is no such system to give rating and reviews. The other cost-effective way is to go to a local garage. They will charge you less than the Dealer but still the problem of visiting there by making some time out of your busy schedule still exists. Also the Mechanics working in these types of the Local Garage are making way lower income than what other mechanics of same expertise are getting in Dealers or some Official Manufacturer's service center. India is a place where people with high skills in these types of work (e.g. servicing) are getting paid lower than industry standards because they don't have any platform to reach the masses and they end up working in some local garages, where they are underpaid.

So I think that the current system for automobile repair is broken and need some change. By using my skills which I have acquired in IT, I want to change this whole system by providing an infrastructure for mechanics to get to the mass audience (here, car owner) also wants to provide a medium for car owner to find and book mechanics.



### Existing Applications similar to Online Mechanic Finder:

- YourMechanic  
YourMechanic was founded in 2012 with the goal to make car repair and maintenance, affordable, convenient, and transparent.
- SuperCheapAuto  
SuperCheapAuto is a thriving specialty retail business, specializing in automotive parts and accessories. It is an Australian based car repair company

Though these startups are solving the same problem which we are solving but the way of solving the problem is completely different. They provide mechanics under their brand so mechanic themselves could not create their own brand while using their system. This limitations is solved by our system where mechanics can create a good reputation and their own brand by using our system which let them connect to massive user base. We are also keeping the garages in our system giving them the ability to reach to massive audience and thus increasing their sales. Hence, we are creating an online platform where Owners can be benefitted by getting a good mechanic to repair his car when he needs, Mechanics can increase their earnings and can also book services and products from garages and lastly garages can increase their sales and create online presence through our system.

## 3.2 Requirements Specification

The system provides Car Owners to choose a highly skilled good mechanic who can either take your car to any nearby garage or will take it to his own place, will repair it and then will drop it to your place or you can drop your car to the mechanics place and he'll use his skills to work perfectly on your car and eventually fix it.

Because this system is changing the whole automobile repair system, it cannot happen overnight. Hence, we are connecting the current local garage to our system where we are focusing on increasing the sales and services provided by those garages. For e.g. Mechanic can purchase the parts provided by these Local Garages. Car Owners can avail some special offer like free Car Inspection, free Car Wash, Towing Services, Repair Services provided by these Local Garages.

This will help boost the sales of these Garages and so will also make sure that our system is not completely demolishing the current system and therefore, our system will grow and change the current workflow of how these Local Garage operates.

We are on a mission to change the current Automobile Repair System by interconnecting Car Owner, Garages and Mechanics. This system has the potential to change the current automobile repair industry.

### 3.3 Planning and Scheduling

#### **Software development model:**

This project will be developed using the Iterative Model of Software Development process. An **Iterative Life Cycle Model** does not start with a full specification of requirements. In this model, the development begins by specifying and implementing just part of the software, which is then reviewed in order to identify further requirements. Moreover, in iterative model, the iterative process starts with a simple implementation of a small set of the software requirements, which iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed. Each release of Iterative Model is developed in a specific and fixed time period, which is called iteration.

Iterative model can be used in the following scenario:-

- When the requirements of the complete system are clearly defined and understood.
- The major requirements are defined, while some functionalities and requested enhancements evolve with the process of the development process.
- A new technology is being used and is being learnt by the development team, while they are working on the project.
- If there are some high risk features and goals, which might change in the future.
- When the resources with needed skill sets are not available and are planned to be used on contract basis for specific iterations.

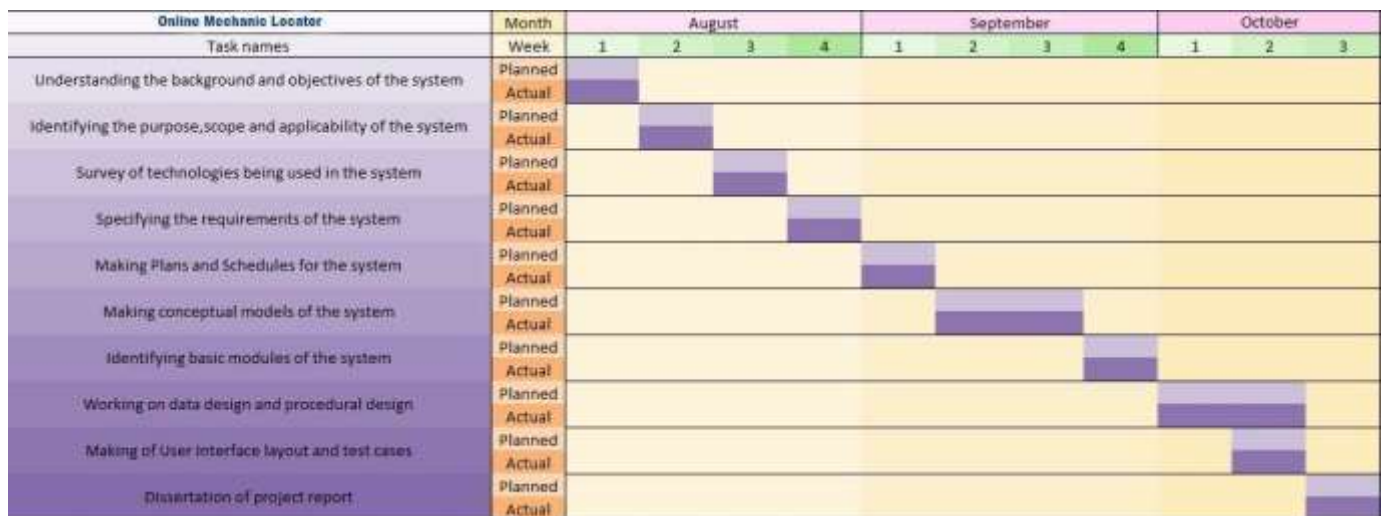
The above characteristics makes iterative model suitable for developing this System where we want to develop a Minimum Viable Product (MVP) of Online Mechanic Locator and deploy

in the market to get reviews and report from the market to incorporate in the system and release the next version with extended features and functionalities.

### **Process of Iterative model:**

The process of Iterative Model is cyclic, unlike the more traditional models that focus on a rigorous step-by-step process of development. In this process, once the initial planning is complete, a handful of phases are repeated again and again, with the completion of each cycle incrementally improving and iterating on the software.

### **Gantt Chart:**



*Figure 3.1 Gantt Chart*

### 3.4 Software and Hardware Requirements

#### Software Requirements:

The following are the recommended requirements for the smooth functioning of the entire system:

Software	Requirements
<b>Operating system Platform</b>	<b>Windows 10</b>
<b>Browser</b>	<b>Any of Chrome, Mozilla, Opera etc.</b>
<b>Development Environment</b>	<b>Sublime Text 3, Django Server, Cordova or PhoneGap, Git, GitHub</b>
<b>Development Tools</b>	<b>HTML, CSS, Bootstrap, JavaScript Framework, Django</b>
<b>Database</b>	<b>MySQL</b>

*Table 3.3 Software Requirement*

## Hardware Requirements:

The following are the minimum requirements for the smooth functioning of the entire system:

Hardware	Requirements
<b>Processor</b>	<b>Intel CORE i3 7th Gen</b>
<b>RAM</b>	<b>500 MB or Higher</b>
<b>Hard Disk</b>	<b>2 GB or Higher</b>
<b>Internet</b>	<b>High-speed Internet connection</b>

*Table 3.4 Hardware Requirement*

## 3.5 Preliminary Product Description

The system is very innovative and has the potential to change how people do servicing of their car. It has the potential to provide employment to many mechanics who have the skills but don't have an exposure to the market.

### Following are Preliminary Product Description:

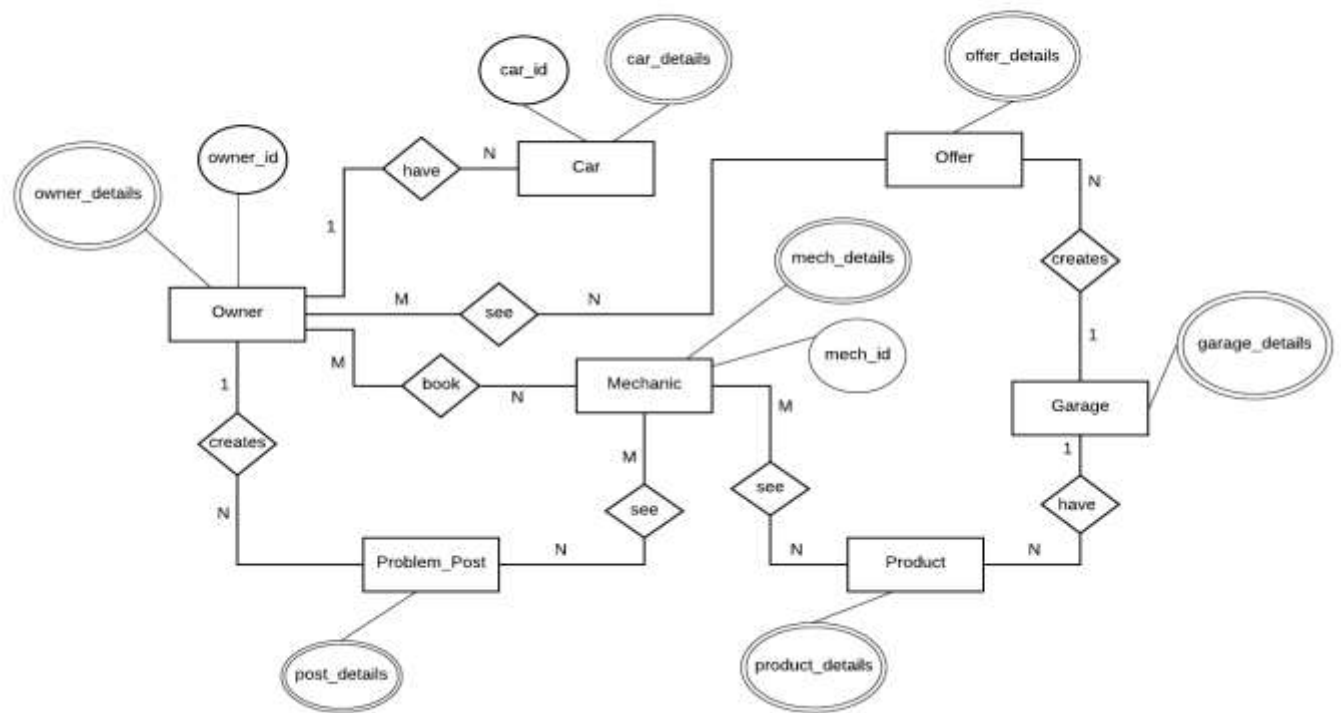
- Car Owners can find and book Mechanics through this system at the comfort of sitting at their home without having to go to garage for repairing their cars.
- Car Owners can also find good Deals and Offers from various different Garage through this system.
- Through this system, Mechanic can earn money by getting connected to huge community of Car Owners where he can provide his services.
- Garages can provide offers and product listing through this system thus increasing their sales.

## 3.6 Conceptual Models

### Entity Relationship Diagram:

The ER model defines the conceptual view of a database. It works around real-world **entities** and the associations among them. At view level, the ER model is considered a good option for designing databases.

### **ER Diagram:**



*Figure 3.2 ER Diagram*

# CHAPTER 4

## SYSTEM DESIGN

### 4.1 Basic Modules

There will be three modules in this system i.e. Owner, Mechanic, Garage.

The details of these modules is given below.

#### 1. Owner

- Login/Register.
- Edit their Profile.
- Can see and locate Mechanics Profiles, Garages offers and services.
- Can broadcast their car problem, it can be seen by all the Mechanics in the system.
- Can tell problem to a particular Mechanic, it can only be seen by that Mechanic.
- Can chat with Mechanics.
- Can give ratings and reviews to Mechanics and Garages.
- Can report Garages and Mechanics.

#### 2. Mechanic

- Login//Register.
- Edit their Profiles.
- Can see the broadcasted problems and confirm or ignore them. If confirmed then can chat with Owner and solve his/her problems.
- Can see his personalized question asked by User and if agrees than can chat with Owner.
- Can see various Garages nearby and see their services like motor parts, slots for repairing is open or not.
- Can chat or call Garage for booking slots or purchasing motor parts.
- Can Report Garages.

### 3. Garages

- Login/Registration.
- Can sell their services (e.g. Car Wash, Free Inspection of Car) to Owners.
- Can sell motor parts and slot booking facility to Mechanics.
- Can chat with Mechanics.
- Can report Mechanics.

## 4.2 Data Design

### 4.2.1 Schema Design

Owner Table:

Field Name	Datatype	Field Length	Description	Example
owner_id	INTEGER	5	Owner ID	11
first_name	VARCHAR	10	First Name	Ajay
middle_name	VARCHAR	10	Middle Name	Anand
last_name	VARCHAR	10	Last Name	Sharma
email	VARCHAR	20	Email ID	ajayasharma@gmail.com
address	VARCHAR	30	Residential Address	Santacruz, Mumbai
pincode	INTEGER	6	Pin Code	400054
phone	VARCHAR	10	Contact Number	9768876321



Car Table:

Field Name	Datatype	Field Length	Description	Example
car_id	INTEGER	5	Car ID	12
car_make	VARCHAR	12	Car Make	Mahindra
car_model	VARCHAR	12	Model Number	Scorpio
year	INTEGER	4	Car is of which year	2017

Problem\_Post Table:

Field Name	Datatype	Field Length	Description	Example
post_id	INTEGER	5	Post ID	13
post	TEXT	80	Post	Problem Posts
date_time	DATETIME	-	Date and Time of Post creation	15/06/2018 11:15:09

Mechanic Table:

Field Name	Datatype	Field Length	Description	Example
mechanic_id	INTEGER	5	Mechanic ID	15
first_name	VARCHAR	10	First Name	Narendra
middle_name	VARCHAR	10	Middle Name	Ashok
last_name	VARCHAR	10	Last Name	Mishra
email	VARCHAR	20	Email ID	narendramishra@gmail.com
address	VARCHAR	30	Mechanic Address	Vile Parle, Mumbai
pincode	INTEGER	6	Pin Code	400056
phone	INTEGER	10	Contact Number	8976102554

Garage Table:

Field Name	Datatype	Field Length	Description	Example
garage_id	INTEGER	5	Garage ID	25
garage_name	VARCHAR	20	Garage Name	Hero Auto Parts
garage_owner	VARCHAR	16	Full Name of Garage Owner	Siven Singh
registration_number	VARCHAR	20	Garage Registration Number	267XJ123W8
address	VARCHAR	30	Garage Address	Santacruz, Mumbai
pincode	INTEGER	6	Pin Code	400054
email	VARCHAR	20	Email ID	sivensingh123@gmail.com
phone	INTEGER	10	Contact Number	9768876321

Product Table:

Field Name	Datatype	Field Length	Description	Example
product_id	INTEGER	5	Product ID	34
car_product_make	VARCHAR	12	Product is of which Car company	Mahindra
car_product_model	VARCHAR	12	Product is of which model	Thar
product_name	VARCHAR	16	Product Name	Brake Pad
product_price	FLOAT	6	Product Price	680

Offer Table:

Field Name	Datatype	Field Length	Description	Example
offer_id	INTEGER	5	Offer ID	41
offer_title	VARCHAR	12	Offer Title	Free Car Wash
offer_details	TEXT	30	Offer Details	Get your car wash for free

## 4.2.2 Data Integrity and Constraints

Owner Table:

Field Name	Allow Nulls	Unique	Constraint	Key
owner_id	No	Yes	Auto-Increment	Primary
car_id	-	-	-	Foreign
post_id	-	-	-	Foreign
first_name	No	No	-	-
middle_name	Yes	No	-	-
last_name	No	No	-	-
email	No	Yes	Check valid email	-
address	No	No	-	-
pincode	No	No	= 6	-
phone	No	Yes	= 10	-

Car Table:

Field Name	Allow Nulls	Unique	Constraint	Key
car_id	No	Yes	Auto-Increment	Primary
car_make	No	No	-	-
car_model	No	No	-	-
year	No	No	>2005 and < current year	-

Problem\_Post Table:

Field Name	Allow Nulls	Unique	Constraint	Key
post_id	No	Yes	Auto-Increment	Primary
post	No	No	-	-
date_time	No	No	Auto Generated	-

Mechanic Table:

Field Name	Allow Nulls	Unique	Constraint	Key
mechanic_id	No	Yes	Auto-Increment	Primary
first_name	No	No	-	-
middle_name	Yes	No	-	-
last_name	No	No	-	-
reviews	Yes	No	Set by Car Owner	-
ratings	Yes	No	Set by Car Owner	-
email	No	Yes	Check valid email	-
address	No	No	-	-

pincode	No	No	=6	-
phone	No	Yes	=10	-

Garage Table:

Field Name	Allow Nulls	Unique	Constraint	Key
garage_id	No	Yes	Auto-Increment	Primary
offer_id	-	-	-	Foreign
product_id	-	-	-	Foreign
garage_name	No	No	-	-
garage_owner	No	No	-	-
registration_number	No	Yes	-	-
reviews	Yes	No	Set by Car Owner	-
ratings	Yes	No	Set by Car Owner	-
address	No	No	-	-
pincode	No	No	=6	-
email	No	Yes	Check valid email	-
phone	No	Yes	=10	-

Product Table:

Field Name	Allow Nulls	Unique	Constraint	Key
product_id	No	Yes	Auto_increment	Primary
car_product_make	No	No	-	-
car_product_model	Yes	No	-	-

product_name	No	No	-	-
product_price	No	No	-	-

Offer Table:

Field Name	Allow Nulls	Unique	Constraint	Key
offer_id	No	Yes	Auto-Increment	Primary
offer_title	No	No	-	-
offer_details	No	No	-	-

Mapping tables:

Owner-Mechanic-Mapping Table:

Field Name	Allow Nulls	Unique	Constraint	Key
mapping_id	No	Yes	Auto-Increment	Primary
owner_id	-	-	-	Foreign
mechanic_id	-	-	-	Foreign

Owner-Offer-Mapping Table:

Field Name	Allow Nulls	Unique	Constraint	Key
mapping_id	No	Yes	Auto-Increment	Primary
owner_id	-	-	-	Foreign
offer_id	-	-	-	Foreign

Mechanic-Problem\_Post-Mapping Table:

Field Name	Allow Nulls	Unique	Constraint	Key
mapping_id	No	Yes	Auto-Increment	Primary
mechanic_id	-	-	-	Foreign
post_id	-	-	-	Foreign

Mechanic-Product-Mapping Table:

Field Name	Allow Nulls	Unique	Constraint	Key
mapping_id	No	Yes	Auto-Increment	Primary
mechanic_id	-	-	-	Foreign
product_id	-	-	-	Foreign

## 4.3 Procedural Design

### 4.3.1 Logical Diagram

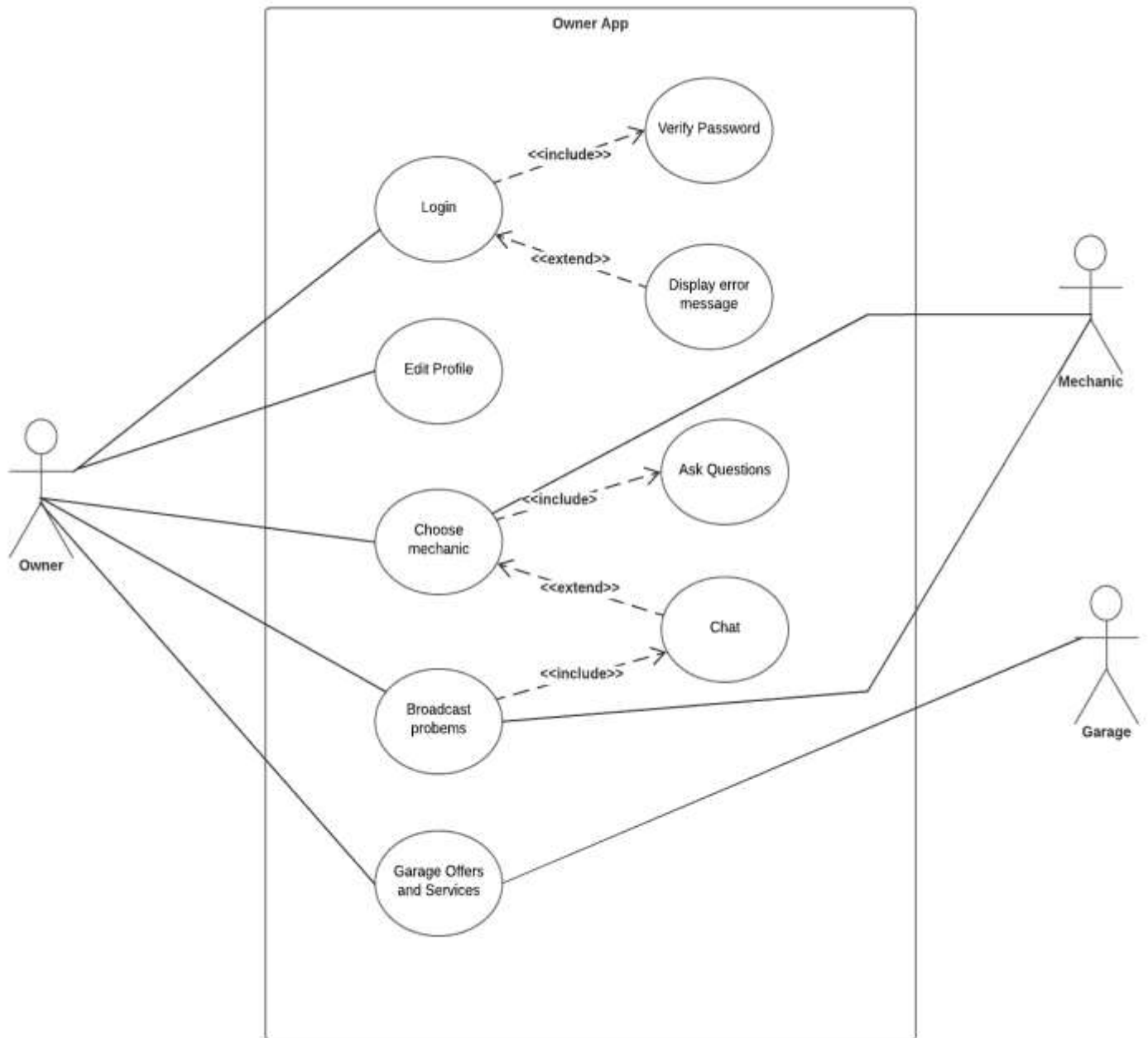


Figure 4.1 Owner App Use Case



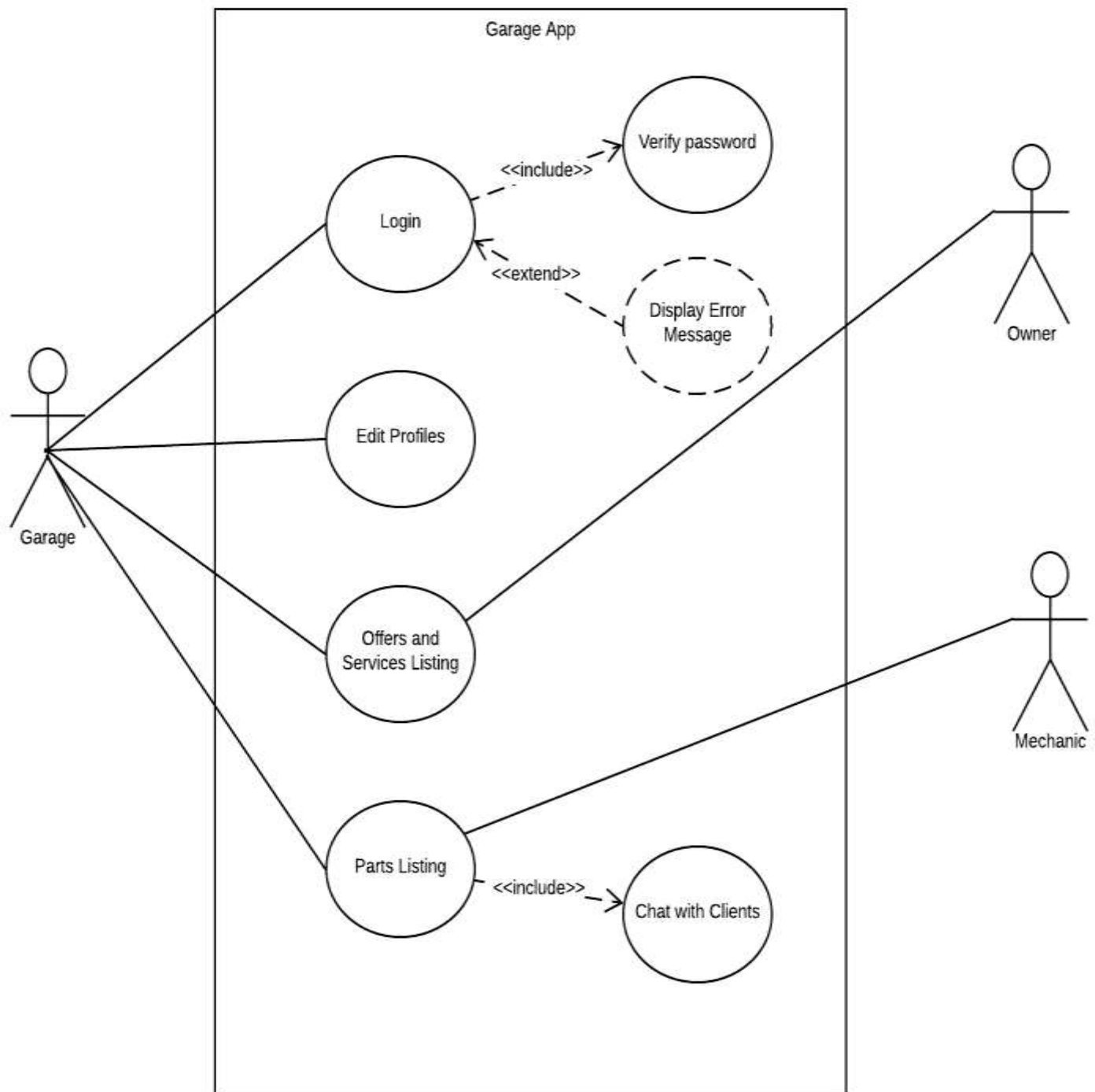


Figure 4.2 Garage App Use Case



Figure 4.3 Mechanic App Use Case

## Sequence Diagram:

### Owner Sequence Diagram

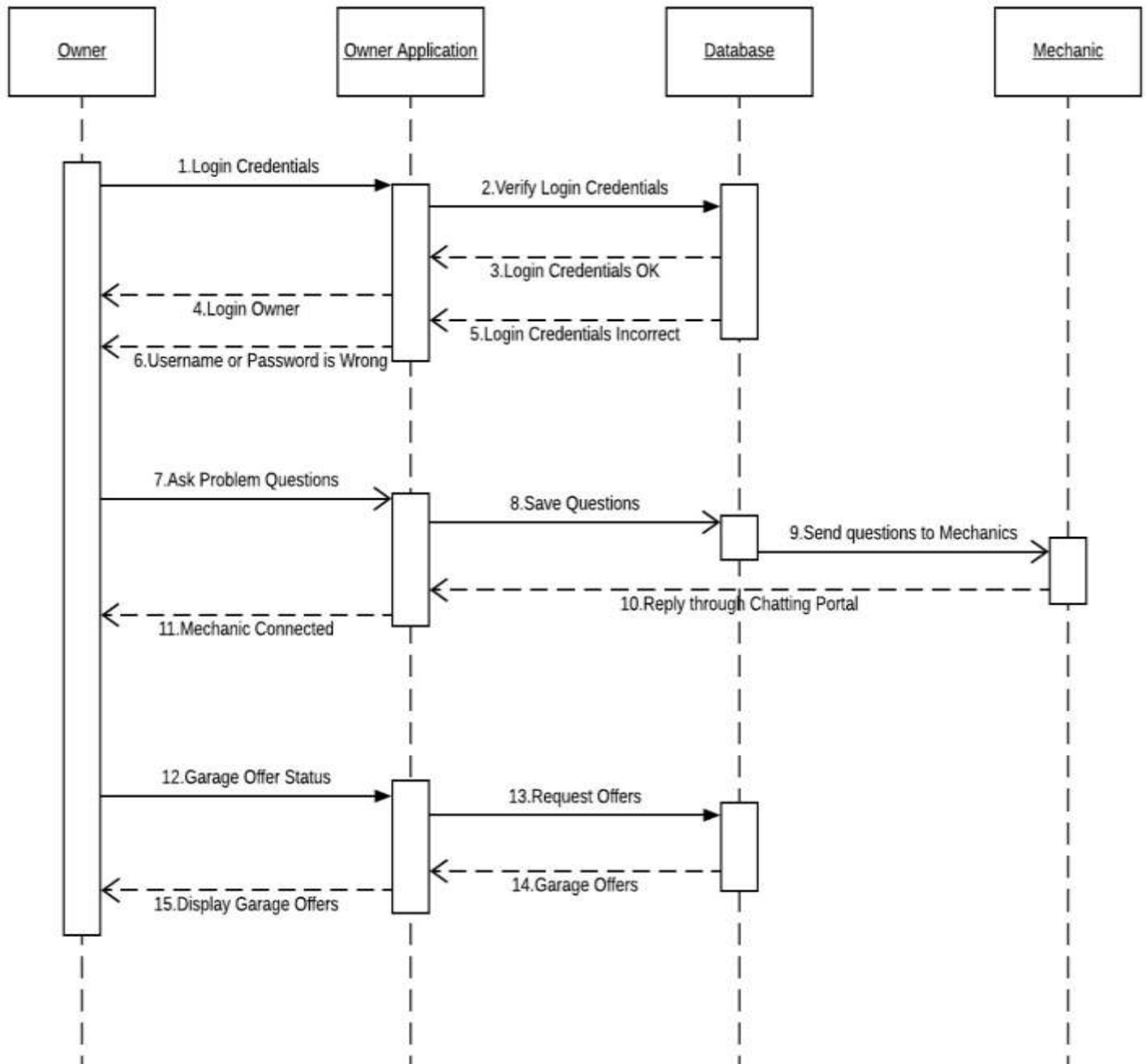


Figure 4.4 Owner Sequence Diagram

### Garage Sequence Diagram

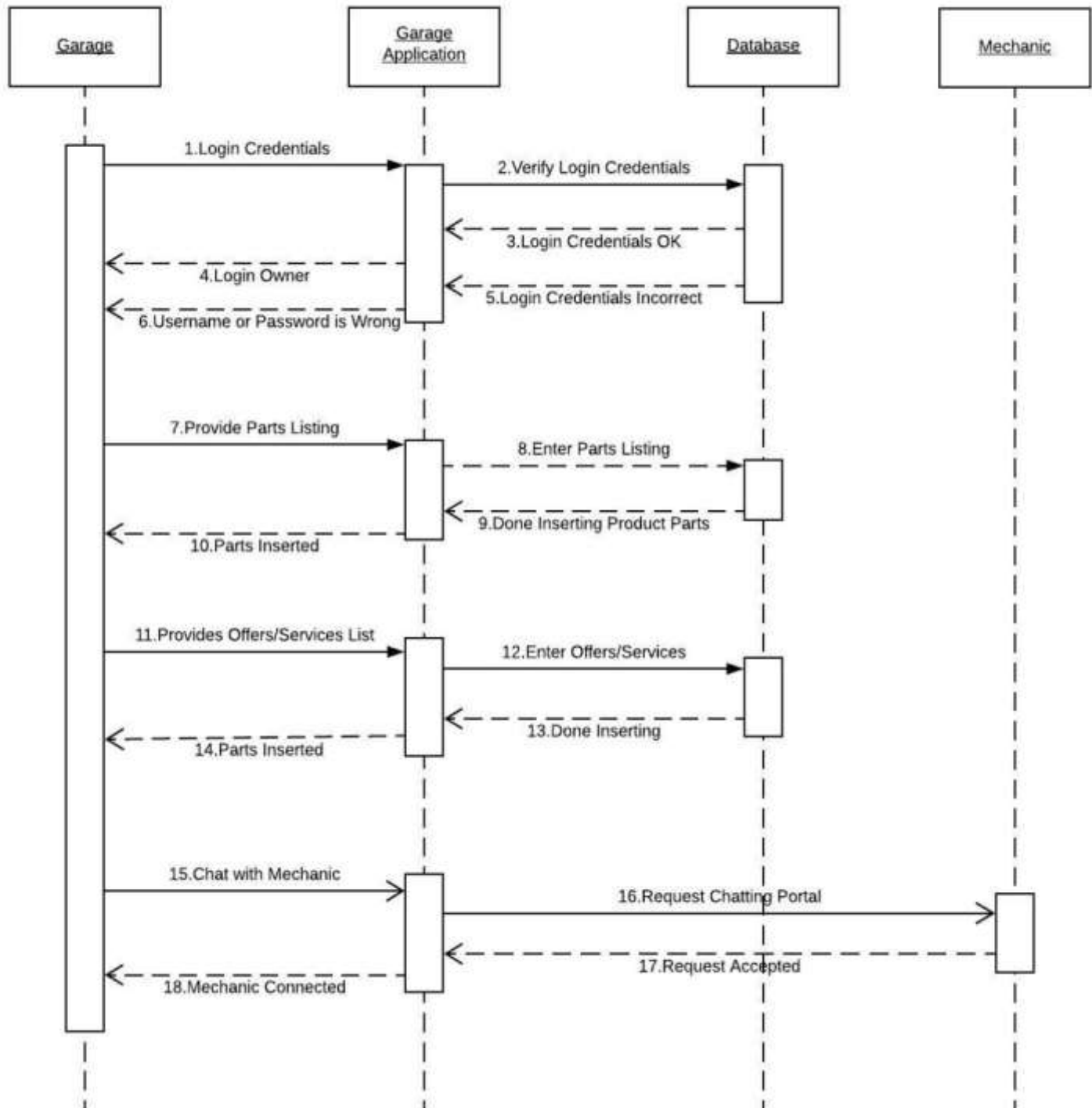


Figure 4.5 Garage Sequence Diagram

### Mechanic Sequence Diagram

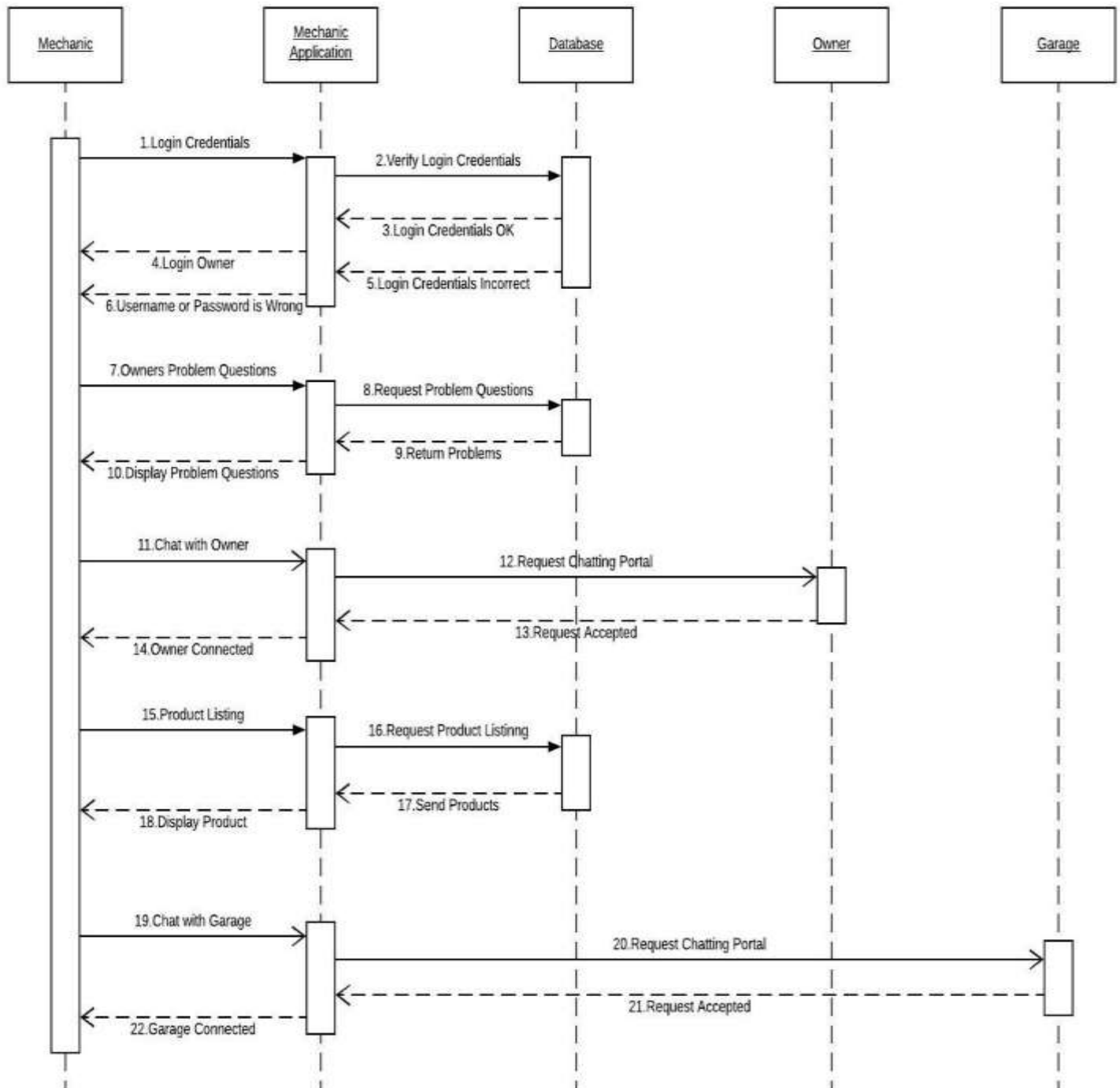


Figure 4.6 Mechanic Sequence Diagram

## Activity Diagram:

### Garage Activity Diagram:

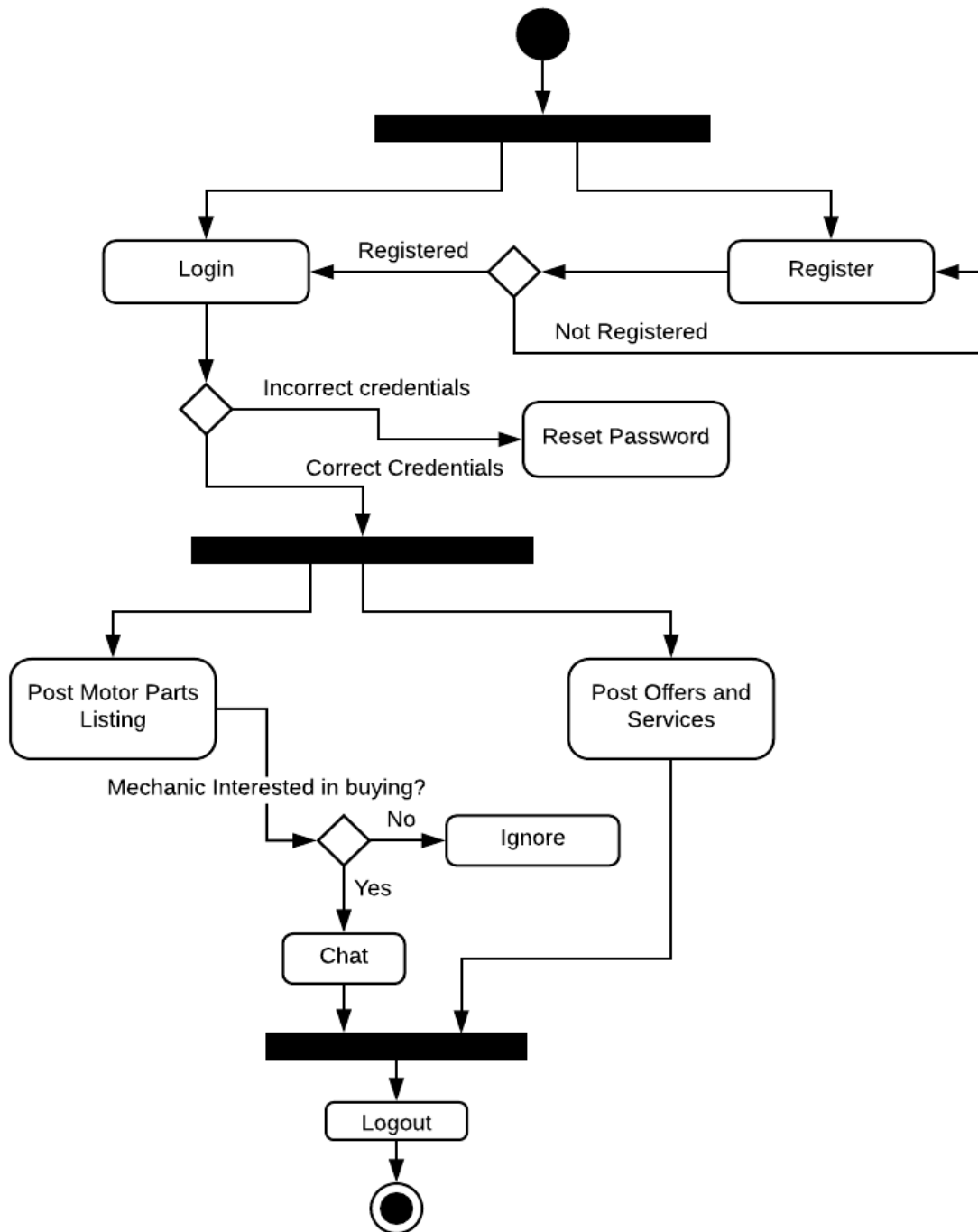
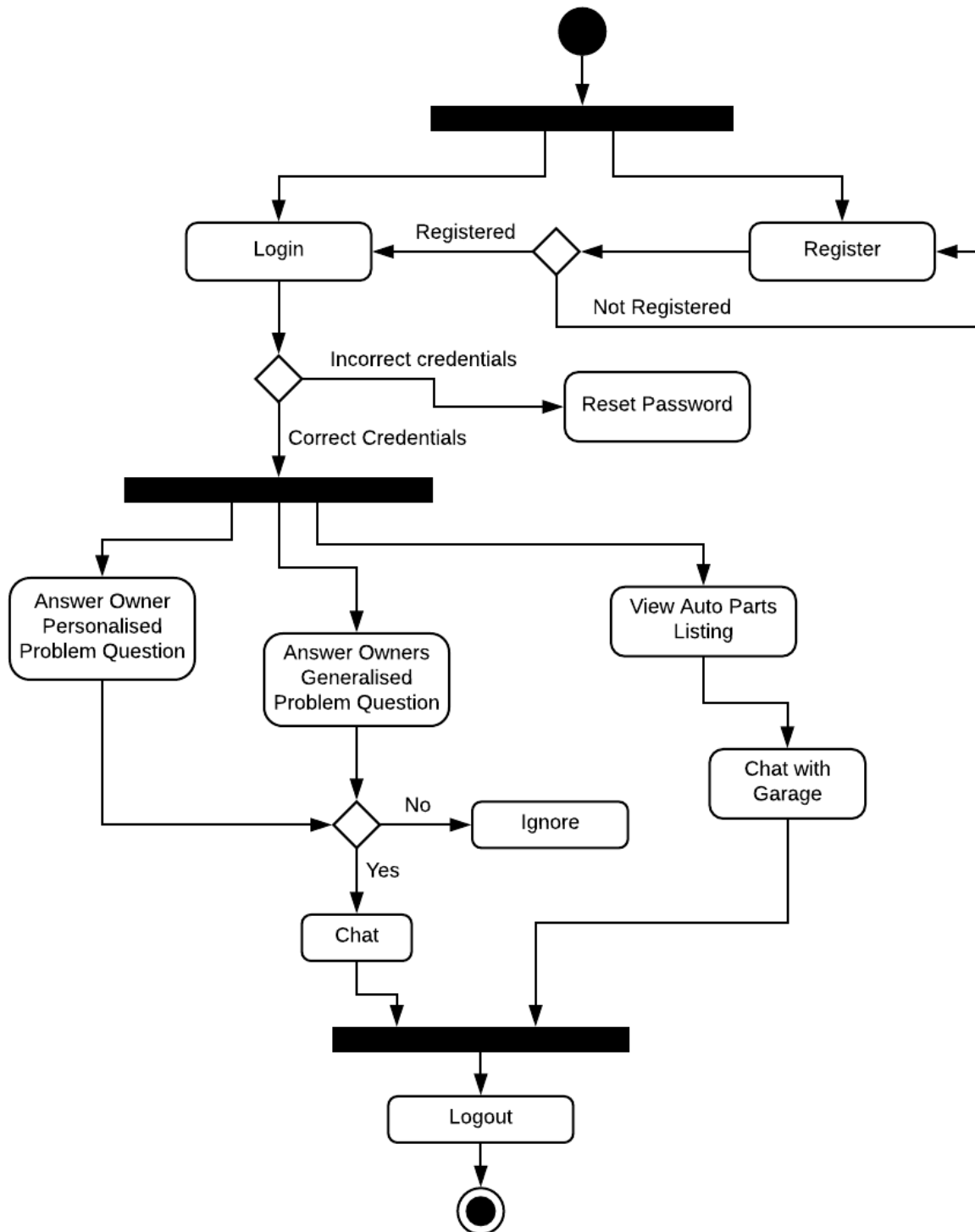


Figure 4.7 Garage Activity Diagram

**Mechanic Activity Diagram:**



*Figure 4.8 Mechanic Activity Diagram*

### Owner Activity Diagram:

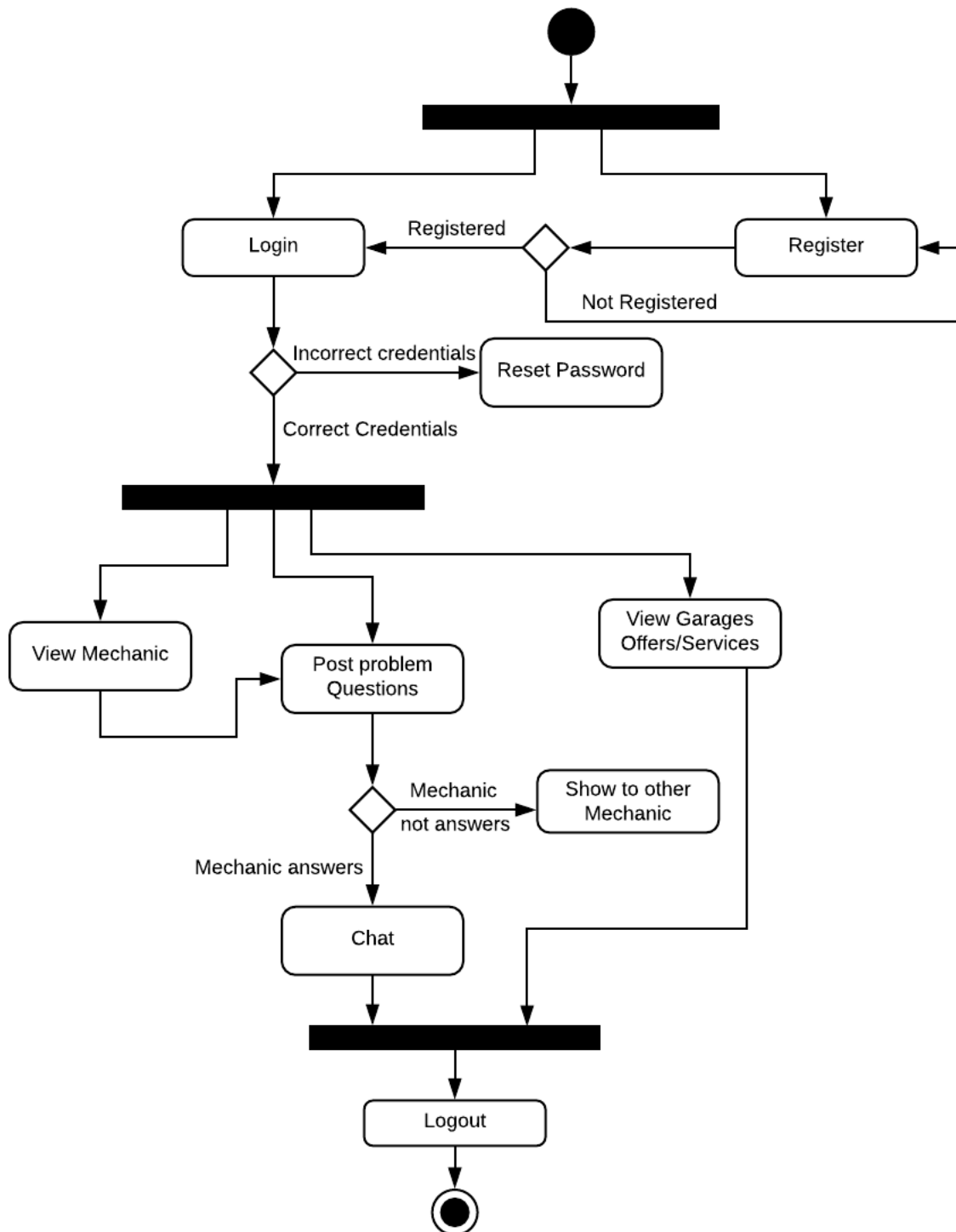


Figure 4.9 Owner Activity Diagram

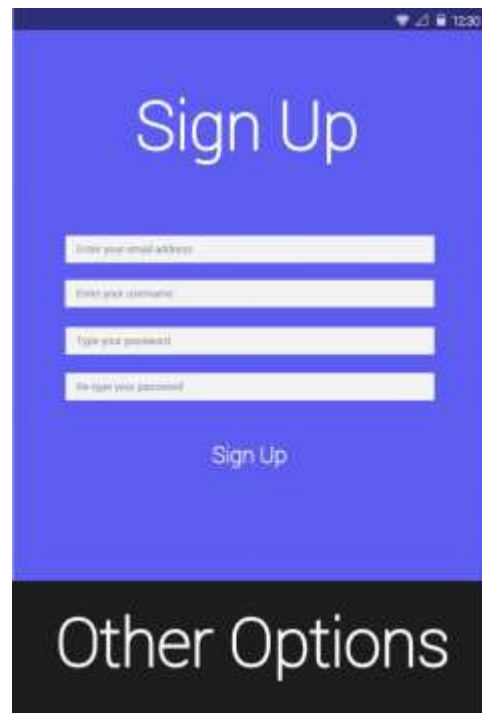


## 4.4 User Interface Diagram



A mobile app login screen with a red background. At the top, the word "Login" is displayed in white. Below it, there are two input fields: "Username" and "Password", each with a small eye icon to toggle visibility. A "Login" button is centered below the fields. At the bottom, a black bar contains the text "Forget Password" in white.

Login Page



A mobile app sign-up screen with a blue background. At the top, the words "Sign Up" are displayed in white. Below it, there are four input fields: "Enter your email address", "Enter your username", "Type your password", and "Re-type your password". A "Sign Up" button is centered below the fields. At the bottom, a black bar contains the text "Other Options" in white.

Sign up Page



A mobile app home screen with a dark background. It features three horizontal cards, each with a profile icon, a name, and a status. At the bottom, there is a navigation bar with three items: a profile icon, the text "Broadcast", and the text "Garage".

Home Page

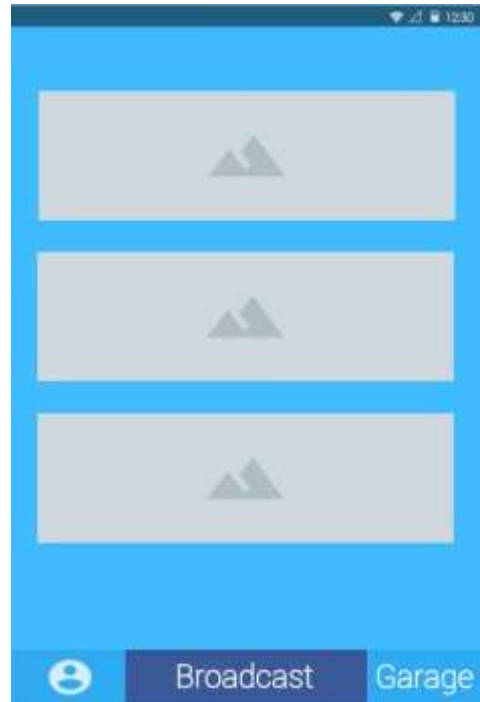


A mobile app profile screen with a light blue background. At the top, there is a circular profile picture placeholder. Below it, there are several input fields for personal information: "Name", "Address", "Car Model", "Car manufacture year", "Phone number", "Email address", and "Bio".

Profile Page



Broadcast Problem  
Question Page



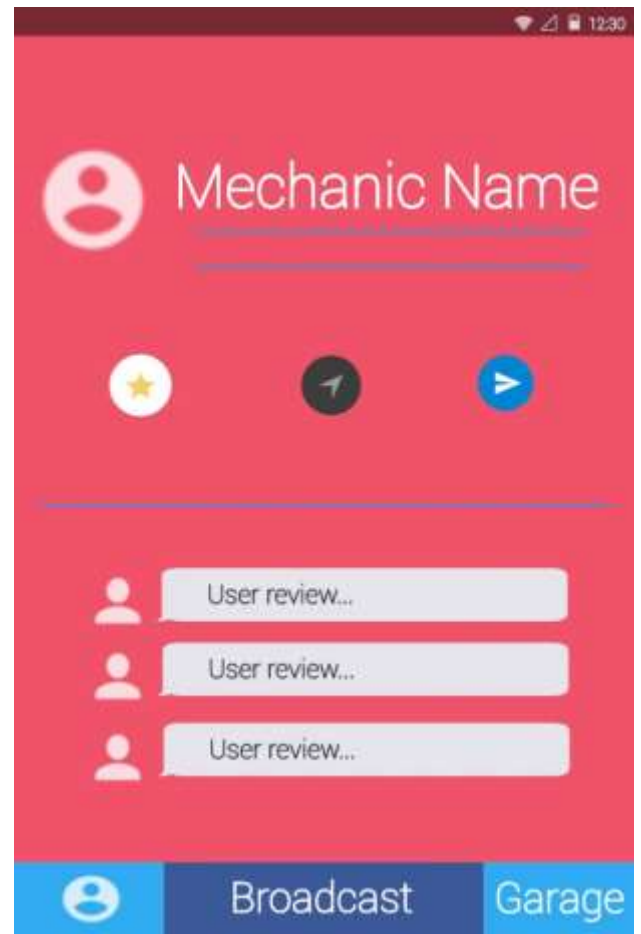
Garage Offers



Garage UI



Mechanic UI



Mechanic Profile

*Figure 4.10 GUI*

## 4.5 Security

### Authentication

- Wrong Username: If the Username is wrong user won't be able to enter the application. The username is checked from the Authentication.
- Wrong Password: If the Password is wrong user won't be able to enter the application. The username is checked from the Authentication.

### Profile

- Car Owner Data: The Car owner Data is very important. When the car owner enters the profile, the information is stored in the database. When the car owner enters the app and selects profile, car owner must retain his own data. If the car owner wants to update his profile the data should be updated into database.
- Mechanic Data: When the mechanic enters his profile, the information is stored in the database. When mechanic enters the app and selects profile, mechanic must retain his own data. If the mechanic wants to update his profile the data should be updated into database.
- Garage Owner Data: When the Garage Owner enters his profile, the information is stored in the database. When the garage owner enters the app and selects profile, garage owner must retain his own data. If the garage owner wants to update his profile the data should be update into the database.

### Broadcasting Question Problem

- Unauthorized Access: Only the car owners is allowed to broadcast any question problem and the same would be uploaded into the database. Only the car owner is allowed to edit or delete his/her broadcasted question problem. No other person have the access to delete them or edit them.
  - The Mechanic has the access only to answer those broadcasted question.
  - The Mechanic cannot chat the car owner until both the party have a deal.

## Garage Offers

- Unauthorized Access: Only the garage owner is allowed to upload any offers of their garage into the database.
  - The Mechanic has the access only to check the offers
  - The Mechanic cannot edit the offers only the Garage owners has access to it.

## 4.6 Test Cases

ID	Name	Description	Steps	Expected outcome
TC1	Login test case	To test the Functionality Of the user	<ol style="list-style-type: none"><li>1. Login with the Registered user in the Database</li><li>2. Login with partially Correct information</li><li>3. Login with no information.</li></ol>	The application Should successfully Login the user when Both email and password Are correct otherwise Display appropriate Error message to the user
TC2	Reset Password Test case	To test Weather user can reset password if the user forgets it	<ol style="list-style-type: none"><li>1. Enter email address and the application should click on reset password, be able to send reset</li><li>2. Open user email and password email to click on specified link. The user and update</li><li>3. Input the new password. The password upon</li><li>4. Login with the new password rest. Password.</li></ol>	The application should be able to send reset password email to the user and update the password up on password reset.

TC3	Change Password Test case	To test whether user can change password if user wishes to change	<ol style="list-style-type: none"> <li>1. Login using the user credentials.</li> <li>2. Open user profile and open change password dialog.</li> <li>3. Input current password and new password twice.</li> <li>4. Logout and try to login with new password</li> </ol>	The application will be able to change user's password if the user knows it's current password if the user knows it's current password, the same should reflect when user tries to login next time.
TC4	Broadcasting Question Problem Test case	To test whether the user can broadcast a problem question	<ol style="list-style-type: none"> <li>1. Login using the user credentials.</li> <li>2. In home page, select Broadcast option and type the problem question in the text field.</li> <li>3. Click on Send button to broadcast it.</li> </ol>	The application will be able to broadcast problem questions which user inputs.
TC5	Logout test case	To test the functionality of logging out of the user	<ol style="list-style-type: none"> <li>1. Click on the Gear icon</li> <li>2. Scroll &amp; Click on the logout button.</li> </ol>	The application should be able to logout the current user, and display login screen.

*Table 4.1 Test Cases*