
# An Undergraduate Internship/Project on A customer and technology management software

By

**Hossain Sahriar Kabir**

Student ID: **1722015**

**Summer, 2021**

Supervisor:

**Sheikh Abujar**

**Lecturer**

Department of Computer Science & Engineering

Independent University, Bangladesh

**August 27, 2021**

**Dissertation submitted in partial fulfillment for the degree of Bachelor of Science in Computer Science**

**Department of Computer Science & Engineering**

**Independent University, Bangladesh**

# Attestation

I hereby declare that this project and report on 'A customer and technology management software' is all my work and has not been copied from any other source, research, article. The whole project is done by me and the whole report is prepared by me. It has been completed under the guidance of Sheikh Abujar. I also declare that my whole project is genuine which I learned in my internship period. And all other information sources used in this report have been duly acknowledged in it.

Signature

Date 27/08/2021

Hossain Sahriar Kabir

Name

# Acknowledgement

This internship project could not have been complete without the participation and assistance of so many people whose names may not all be enumerated. Their contributions are sincerely appreciated and gratefully acknowledged. I want to say that it is very delightful to have this kind of special internship program where we are directly involved in building something. Working for a real-life company in a real-time project is difficult. But it became easy when other people helped us. And It brings joy when the whole project is finished. And a real product is ready for use. I would like to express my gratitude and respect to my internship supervisor Sheikh Abujar, Lecturer, Independent University, Bangladesh for his constant guidance, advice, encouragement in the overall preparation of this report and project. I also want to thank Mechanic koi. They give me a great opportunity to work on a real-time project. They teach me how to deal with corporate problems, manage customers and work on a wonderful project. I also want to thank my internship colleagues to help and support me in the harsh moments. I would also like to thank Independent University, Bangladesh (IUB) for arranging and giving us a chance to participate in such an excellent and beautiful internship program. Above all, to the Greatest Almighty ALLAH, the author of knowledge and wisdom.

# Letter of Transmittal

August 27, 2021
Sheikh Abujar
Lecturer
Department of Computer Science and Engineering
Independent University, Bangladesh.

Subject: Internship Report submission Summer, 2021.

Dear Sir,
It is of immense pleasure and honor to submit my Internship report on 'A customer and technology management software for Mechanic Koi' under your guidance. In this report, I present my project work, analysis, and my achievements.

I have completed my Internship from Mechanic koi as a Software Intern which was conducted from June 01, 2021, to date. In my internship period, I have gathered knowledge in various aspects and real-life working experience. In this report, I include all the project works, experiences, and knowledge that I have achieved during this internship.

I would like to thank you for your constant support, guidance, and kindness. I have tried to complete this with the utmost honesty and sincerity. I hope and pray that this report fulfills all the requirements and is up to your expectations.

Sincerely,
Hossain Sahriar Kabir
ID : 1722015

# Evaluation Committee

.....

Signature

.....

Name

.....

Supervisor

.....

Signature

.....

Name

.....

Internal Examiner

.....

Signature

.....

Name

.....

External Examiner

.....

Signature

.....

Name

.....

Convener

# Abstract

The automotive industry is growing so fast in Bangladesh. Nowadays people are more passionate about car-bikes etc. This market is growing rapidly with new business opportunities. And here Mechanic koi is trying to give a 360-degree vehicle solution and gives emergency support to any vehicle facing on-road turbulence. Mechanic koi always try to provide all kinds of support to the customer. But this is not easy. There are huge customers who want a variety of services and choosing the right vendor for the right service is extremely difficult. And that's why Mechanic koi needs customer relationship management (CRM) software to communicate with customers and provide the best support. With this CRM software, Mechanic koi can easily track customer behavior and activity, demanded service, and the best vendor for different services. To compete with competitors and to advance in the market, there is no other alternative to conduct market research. And to do that Mechanic koi must need customer Relationship Management (CRM) software.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Overview/Background of the Work

Mechanic koi is an online platform that provides a 360-degree vehicle solution and gives emergency support to any vehicle facing on-road turbulence. Technically 'Mechanic Koi' gives support in every portion of a vehicle owner's life cycle and gives an innovative resolution for every vehicle-related problem. [1]

Mechanic koi already has a website and mobile application. By that, customers can get any type of vehicle problem solution in real-time. Customers can also get an appointment. There are so many services and packages available from Mechanic koi. Customers can easily choose them and get help immediately. Mechanic koi has a team of experts to help customers whenever needed.

However, maintaining a large number of customers is not an easy task. To maintain customers, Mechanic koi also needs to track the best vendor. This scale is huge. And that's why Mechanic koi feels they need a Customer Relationship Management (CRM) system of their own.

This project will be a total customer and backend management software for the internal management of the company. With this software, Mechanic koi can manage their customer, service, and vendor smoothly.

## 1.2 Objectives

The objective of this software is described here

- The main objective is, handle customers, vendors, and services smoothly. In one place all information will be gathered, stored, and managed by the admin panel.

Mechanic koi management can easily maintain this system.

- This system can increase customer satisfaction. The customer only chooses the service he needs. Mechanic koi management will choose the best vendor for customer service.
- Mechanic koi can easily track and monitor customer's activity and behavior. It will help Mechanic koi to understand customer demand and current trends.
- Based on customer data, Mechanic koi will create their marketing campaign, promotional activities. Based on customer demand, Mechanic koi will add service and modify old service. Set discounts on different services.
- And most importantly Mechanic koi management or Admin panel can use this system from anywhere anytime. This system will be portable and comfortable to use.

## 1.3 Scopes

- Customers will get all kinds of car solutions anytime.
- Customers can take appointments anytime they want.
- Customers will get the best vendor based on their service.
- All information will be stored automatically.
- Management can handle customers and vendors easily.
- Management can easily find the best vendor and recommend him to the customer.
- Best service can be tracked and newer service will be added as customer demand.

# Chapter 2

# Literature Review

## 2.1 Relationship with Undergraduate Studies

In my undergraduate studies, I took different courses and learned lots of things. I learn how to build something, think about something, how to develop, research, and analysis, something. I also introduced myself to the world of logic and decision-making during my undergraduate study time. As a computer science student, I learn programming, data structure, algorithms, database management systems, web applications & the internet, and system analysis.

To build this project, I mostly work on database management systems and web applications & the internet. This two-course is directly related to my project. In this project, I build relational databases and connect each other. I also work on the front-end. I connect the back-end with the front-end via Rest API.

I also use my system analysis course lesson here to design my system user interaction and design architecture.

## 2.2 Related works

Nowadays each company has one of its own Customer Relationship Management (CRM) systems. Some successful CRM systems are described below.

**Apple CRM:** Apple has its own CRM system which they use for its growth strategy. Every apple user has a unique id. When users access Apple services like App Store, Apple Music, iCloud, etc, Apple tracks their behavior and recommends similar things. It's more likely to be used for marketing purposes.

**Coca-Cola CRM:** Coca-Cola mostly uses it for promotional activity and marketing purposes. Coca-Cola also uses it to deal with customer issues. Coca-Cola also monitors customer activity and social media interaction to promote their product or create a campaign. The Coca-Cola team uses it on their mobile app anytime anywhere in real-time.

**Amazon CRM:** Currently Amazon has one of the best efficient and complete CRM systems in the world. Amazon online shopping is the reason for their CRM system. When a user creates their personal account in Amazon, they add it to the CRM system. Amazon tracks every purchase, browsing history, search history to understand customer interest. And based on that, Amazon recommends them, email them and give them a discount offer. Amazon also creates marketing campaigns and email campaigns based on user activity. [2]

**E-khata:** E-khata is a paid CRM system, where users can track and manage their local customers. Here users can store accounting information, sale information, dynamic dashboard, and report generation. It works as a daily sales book.

**SManager:** SManager is also another paid CRM system specially built for local businessmen. Here local businessmen can store every day's sales information. Users can have the report of the highest selling product, lowest selling product. Users also get everyday sales reports. Smanager also provides digital loans for businessmen. This loan process is easy and simple. They also give users the option to create their own online shop. Every user's data is safely stored in their database.

# Chapter 3

# Project Management & Financing

## 3.1 Work Breakdown Structure

We break down our whole project into smaller tasks to speed up the process. Each team member was assigned one task at a time. This helps us to complete the whole project easily. And this work breakdown structure helps us to understand our workflow.

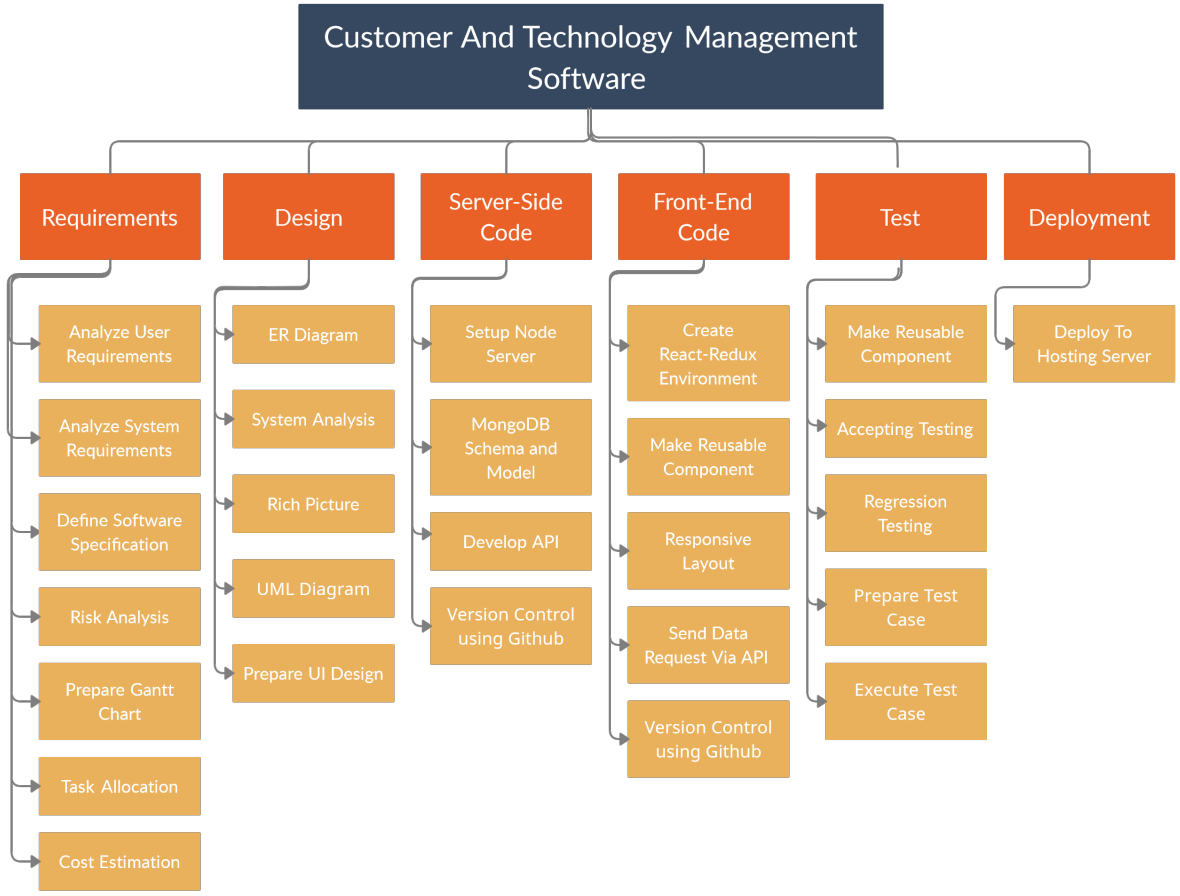


Figure 3.1: Work Breakdown Structure of CTMS

## 3.2 Process/Activity wise Time Distribution

As I mentioned some sections in the work breakdown structure for our project, we also made a time allocation for them. Details time allocation is described below

| Section | Days | Work Percentage (%) |
|---|---|---|
| Requirements | 15 | 15% |
| Design | 15 | 15% |
| Server-Side-Code | 20 | 25% |
| Front-End-Code | 25 | 25% |
| Test | 15 | 15% |
| Deployment | 2 | 5% |
| Total | 92 | 100% |

Table 3.1: Activity wise time distribution

## 3.3 Gantt Chart

We also made a Gantt Chart schedule for our task.

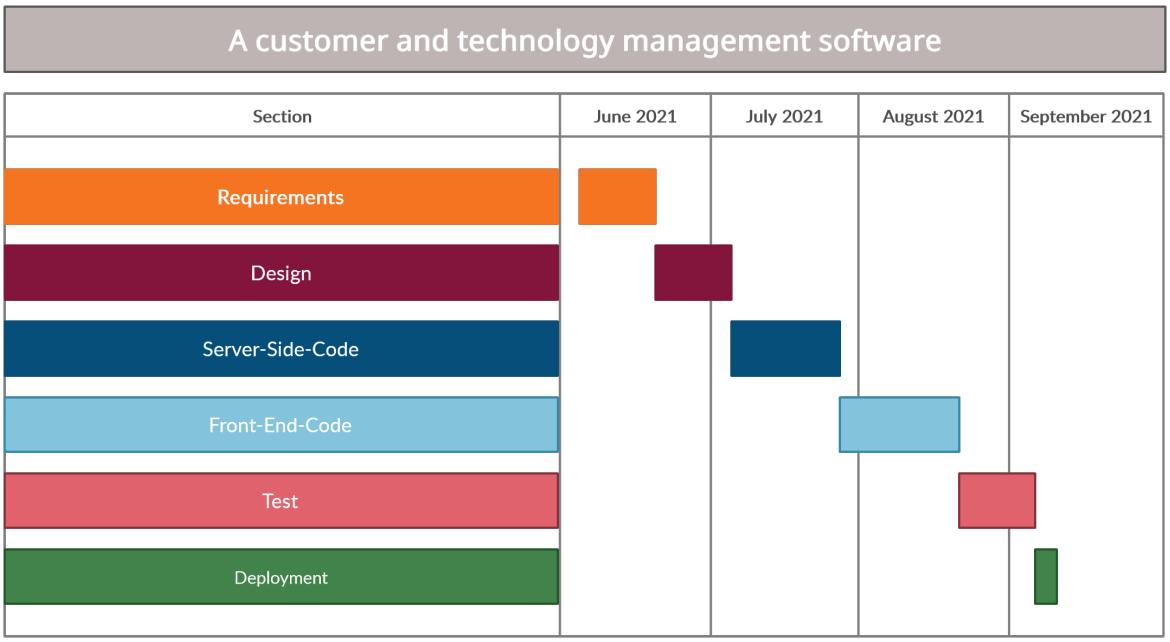


Figure 3.2: Gantt Chart Project Planner

## 3.4 Process/Activity wise Resource Allocation

In Work Breakdown Structure, we already figured out the process or activity we need to complete our project. Now, we are distributing this process or activity to the team members. So that we can complete our project within the estimated time.
newline

| **Section** | **Resource Allocation** |
|---|---|
| Requirements | Project Manager |
| Design | System Analyst |
| Server-Side-Code | Developer |
| Front-End-Code | Developer |
| Test | Q/A Tester |
| Deployment | Project Manager |

Table 3.2: Activity wise Resource Allocation

## 3.5 Estimated Costing

We estimate the cost for the entire project to build and run smoothly. It includes each and everything in our project.

| **Work Distribution** | **Costing** | **Payment** |
|---|---|---|
| Development | $ 1500 | One time |
| User Account | $ 500 | Monthly |
| Domain | $ 20 | Yearly |
| Hosting (Heroku) | $ 100 | Monthly |
| MongoDB (Atlas) | $ 200 | Monthly |
| Other Service | $ 200 | Monthly |

Table 3.3: Estimated Costing

# Chapter 4

# Methodology

For this software, we use the waterfall life cycle method. Because our requirement was fixed and clear from the very beginning. The waterfall life cycle method is best for that software, where requirements are fixed and do not change much in the future. The waterfall life cycle method is characterized by well-defined sequential activities where each step comes one after another in sequential order. When one step is complete, the next step comes. And there is no way to go to the next step when the previous step is incomplete. To build this software, we also go through some steps which are described below.

- **Planning:** Project managers and team leaders are identified the sprint priorities. They create a working procedure.
- **Requirements:** Then we work on requirements. We made a list of our functional requirements and non-functional requirements.
- **Design:** Then we start work on our web page design. At this step, we just made our UI design. We use HTML CSS and CSS framework to build our different components of the web page.
- **Server-Side Code:** Here we build our relational database. We use MongoDB as our database. Then we build our back-end server using Node js and Express js. And finally, we built our Rest API. We also conduct some tests before going to the next step.
- **Front-End Code:** Here we connect our front-end with our back-end via API. In this step, we use the javascript framework React js and a state management library called Redux. In this step, we ensure our request for data is received by the server and passes requested data from the database. We can also store data in the database from the front-end input field.

- **Test:** We conduct different tests on different modules. We try to find issues and solve the issue here.
- **Deployment:** When everything is finished, we deploy it on the hosting server.

These were our all steps from beginning to end.

# Chapter 5

# Body of the Project

## 5.1 Work Description

Mechanic koi is an online platform that provides a 360-degree vehicle solution and gives emergency support to any vehicle facing on-road turbulence. Mechanic koi will build a Customer Relationship Management (CRM) software to communicate and manage customers properly.

In this software, everything will be automated. Only Mechanic koi Management people will use this. There will be Clients. Who buys service. This software will track every service a client takes. When a new client comes, the system will save his information and store it. When this client will come again in the future, this CRM software will recognize the client and update his data.

There will be Vendors in this software, who will give service to customers. Every vendor's performance and behavior will be monitored by this software. A good vendor will get more and more work. A vendor should be smart, well-behaved, and knowledgeable about his work. Here clients can give ratings on vendors. This software will judge a vendor by his cumulative ratings.

And lastly, there will be a Service option. Where all services will be added and stored. When Mechanic koi launches a service, it will be added here. When Mechanic koi launches a package, it will also be stored here. Mechanic koi admin panel can create, update and delete a service. When a particular service is provided to a customer by a vendor, its data will be stored in the Service Appointment area. Here service, customer, and vendor relationships will store.

To build this software, Mechanic koi will use Node js and Express js. They will use

MongoDB as their database. And also, they will use React-Redux to build their front end. This system will work in real-time. This system will reduce the company's previous work and make them more comfortable to use. And stored data will help the company launch new on-demand services and choose the best vendor for customers.

## 5.2 System Analysis

### 5.2.1 Six Element Analysis

| ~~Process~~ | System Role | | | | | |
|---|---|---|---|---|---|---|
| | Human | Non Computer Hardware | Computer Hardware | Software | Database | Communication and Network |
| Login | System Admin | N/A | Computer | Browser | MongoDB | WAN |
| Create Customer | System Admin | N/A | Computer | Browser | MongoDB | WAN |
| Customer Profile | System Admin | N/A | Computer | Browser | MongoDB | WAN |
| Customer List | System Admin | N/A | Computer | Browser | MongoDB | WAN |
| Create Service/Package | System Admin | N/A | Computer | Browser | MongoDB | WAN |
| Update Service/Package | System Admin | N/A | Computer | Browser | MongoDB | WAN |
| Create Vendor | System Admin | N/A | Computer | Browser | MongoDB | WAN |
| Vendor Profile | System Admin | N/A | Computer | Browser | MongoDB | WAN |
| Vendor List | System Admin | N/A | Computer | Browser | MongoDB | WAN |
| Create Service Appointment | System Admin | N/A | Computer | Browser | MongoDB | WAN |
| Update Service Appointment | System Admin | N/A | Computer | Browser | MongoDB | WAN |

Table 5.1: Six Element Analysis

### 5.2.2 Feasibility Analysis

To build and implement our project successfully we need to do a feasibility analysis. It will tell us the benefits of our project. It will also tell us some essential factors that we all need to consider before we start our project.

- **Technical Assessment:** This whole project will build in the MERN stack. To store its data, we will use the MongoDB database. And we will store our data at MongoDB Atlas. MongoDB Atlas is one of the best cloud database stores for MongoDB. We will choose an Indian server to get maximum performance. Also, MongoDB has different package options as customer demand. Also, we will use Node Express for our backend. Node gives us a good environment and express gives us a good optimization speed and performance. In the front end, we will use React js and Redux Js. React js is the most used javascript framework today. It manages front-end code brilliantly. And at last, we will use Redux js to store our data. Redux is a store management framework that stores temporary data globally. Data can move easily in Redex js. Using the MERN stack for this project is a wise decision. To maintain this software, the company needs some employees who have basic knowledge about computers and the internet. This software is easy to use and handle.

- **Operational Feasibility:** This software is easy to use. Employees can easily understand what's going on there and how to do everything. To operate this, the company doesn't need an expert. Anybody can use this software. Maintenance is not an issue for this software. But management should choose a good hosting site to deploy the project. A good hosting site can provide the best performance.

- **Economic Feasibility:** To build this project, the company has to spend $1500. After development, almost $500 will cost per month for different services. Hosting service, Database service will take most of the cost. And employees need to operate this software. But this software has great economic benefits. With this software, the company can easily handle and track customer, vendor, and service data. Most importantly, companies can understand, present trends, and in-demand services. So the company can easily launch service. This software will help to do market research for the company. So the company can benefit economically from this software.

### 5.2.3 Problem Solution Analysis

To build this project, we face some problems. We also try to solve this problem as soon as possible. Some problems were,

- **NPM Error:** To build our project, we use the NPM package manager. We use lots of NPM packages. Most of them work perfectly, but some give us an error, where we have nothing to do. This type of bug directly comes from the package itself. At this time, we downgrade the package version to remove errors.

- **Github Branch Misunderstanding:** Not everybody has enough knowledge about Github. Some developers in my team are not familiar with git branching. There was some time when a person created a branch and committed to it. And other people create a new branch from other branches instead of the main branch. This was problematic. But we manage it later.

- **Different Coding Patterns:** There are lots of different ways to write code, create and implement functions. Also, plenty of coding structures are available. Everybody writes code in their way, which they feel comfortable with. But sometimes other coders don't understand this way of code. This creates a problem in debugging time. We also face this and overcome it.

### 5.2.4 Effect and Constraints Analysis

We build this system to maintain client, vendor, and service smoothly. Also, the company can get up-to-date information about present in-demand services and trends. The company can easily analyze data and launch new services or stop old, fewer demand services. However, this system also has some limitations. The client or vendor can not use this software. This software can only use company management people or admin panels. This has some problems. Sometimes data can be corrupt. So people who use this software should be aware of the data they input. In the future, we will work on updating this software with new features. We will try to use AI to verify our data and analyze it quickly.

## 5.3 System Design

### 5.3.1 Rich Picture

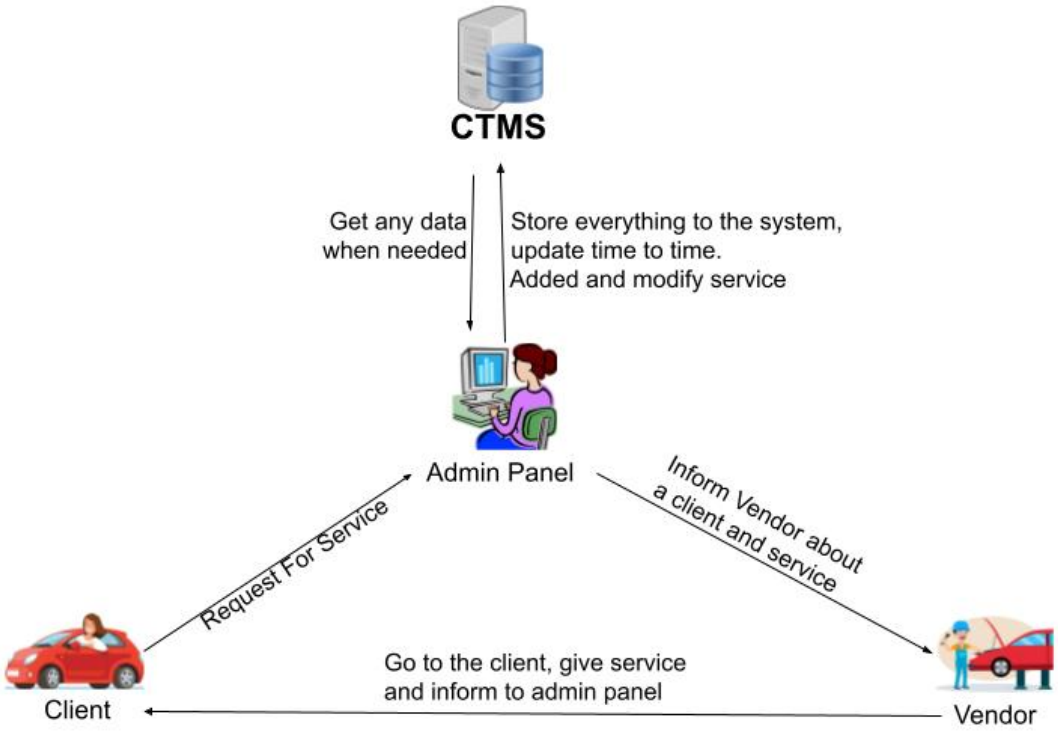


Figure 5.1: Rich Picture

### 5.3.2 UML Diagrams

#### Use Case Diagram

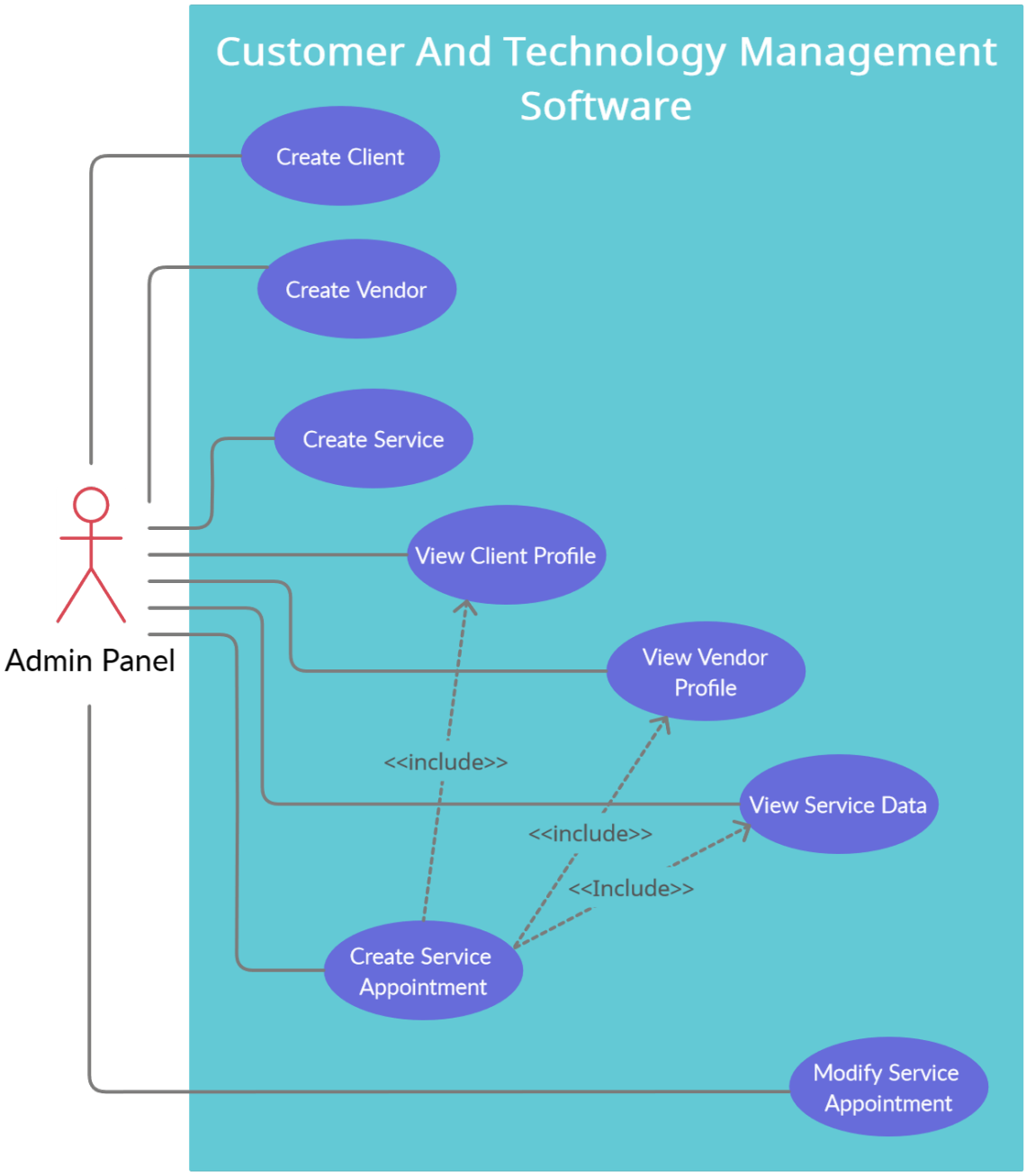


Figure 5.2: Use Case Diagram

### Entity Relationship Diagram

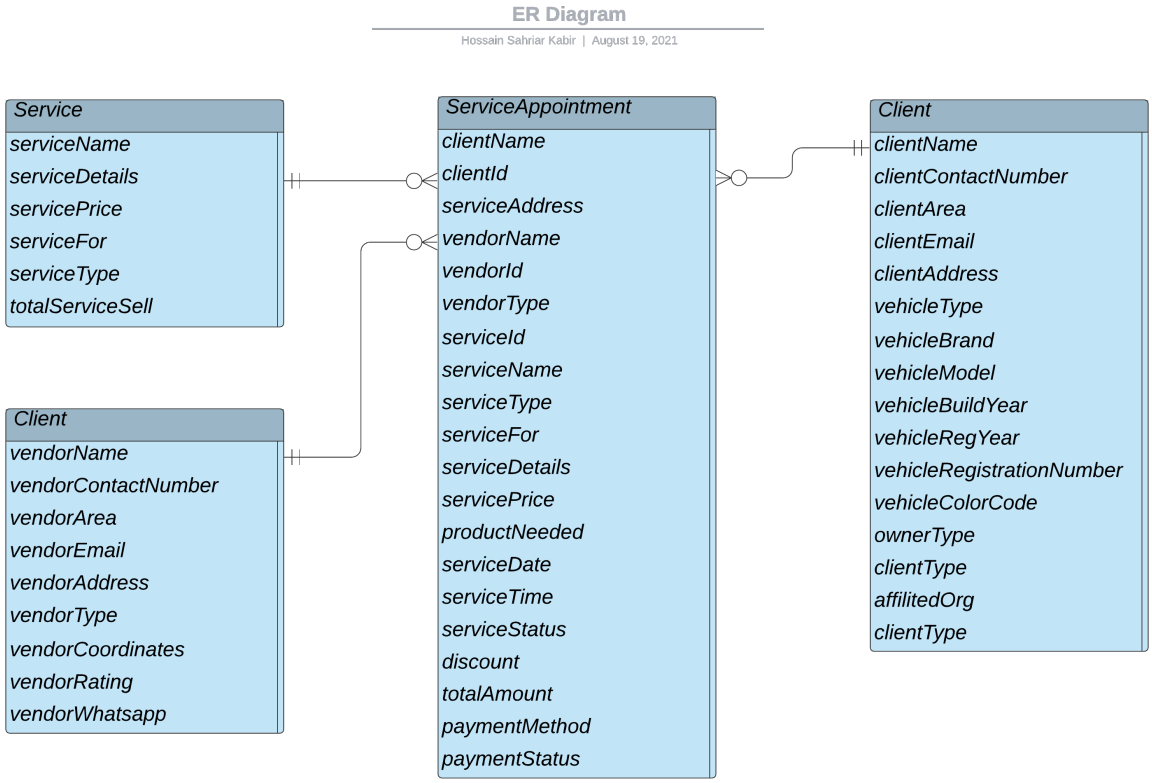


Figure 5.3: Entity Relationship Diagram

## Activity Diagram

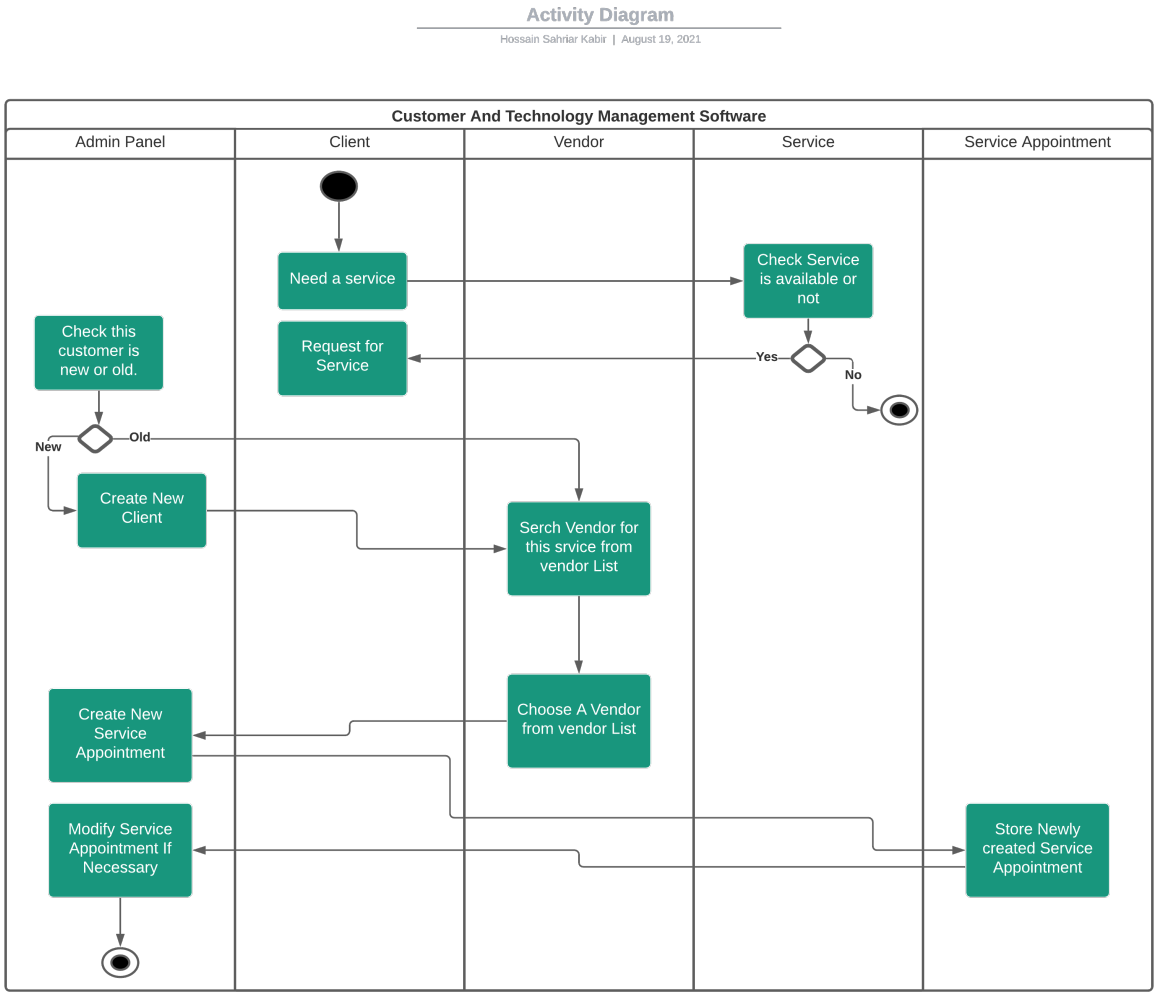


Figure 5.4: Activity Diagram

### 5.3.3 Functional and Non-Functional Requirements

**Functional Requirements**

| Functional Requirement Number | Functional Requirement Description |
|---|---|
| FR 1 | The system shall generate a unique id for every service, client, vendor, and service appointment. |
| FR 2 | The system shall check new user's emails are unique or not. And the system only accepts unique emails from new users. |
| FR 3 | The system shall allow users to create updates and delete services. |
| FR 4 | The system shall store all service appointments for future research and analysis. |
| FR 5 | The system shall display search results as the user search queries. |
| FR 6 | The system shall display all services a particular client uses. |
| FR 7 | The system shall store all browser cookies and search queries to give users a better user experience. |
| FR 8 | The system shall allow displaying clients and vendor profiles. |
| FR 9 | The system shall find most selling service and less selling service. |
| FR 10 | The system shall prioritize top-rated vendors. |
| FR 11 | The system shall allow users to update service appointments to track their progress. |
| FR 12 | The system shall adjust service prices with discounts. |
| FR 13 | The system shall keep track of all services selling money. |

Table 5.2: Functional Requirements

**Non-Functional Requirements**

**Usability**

- The UI of this system is clean and simple. Users can get used to this software so fast.
- Any user can easily use this software.

**Performance**

- The system will be responsive with mobile and any type of device.
- The system will respond to user requests in less than 0.1 seconds.

**Information**

- Every service information will be stored in the database for future use.

**Economy**

- This system will deploy to a cloud hosting provider, So we will not need a data center or server.
- We will activate the pay-as-you-use package from the cloud. provider. So we will only pay as we use.
- This system will work as a market analysis tool. We can easily find in-demand services with this software.

**Control**

- This system will be secure.
- User data will safely backup for disaster recovery.

**Service**

- The system will deploy on the cloud and be regularly maintained by the system admin and other employees.
- Users can use this software from anywhere in the world.
- The system will be portable, changing OS to OS or device to device does not create any problem.
- Necessary system updates will come from time to time.

**Reliability**

- The system will be reliable and available for any kind of OS, device, computer, mobile.
- Because of our safe backup, no data can be lost.
- System updates or any kind of testing will not affect the running system.

## 5.4 Product Features

### 5.4.1 Input

#### Sign Up Page

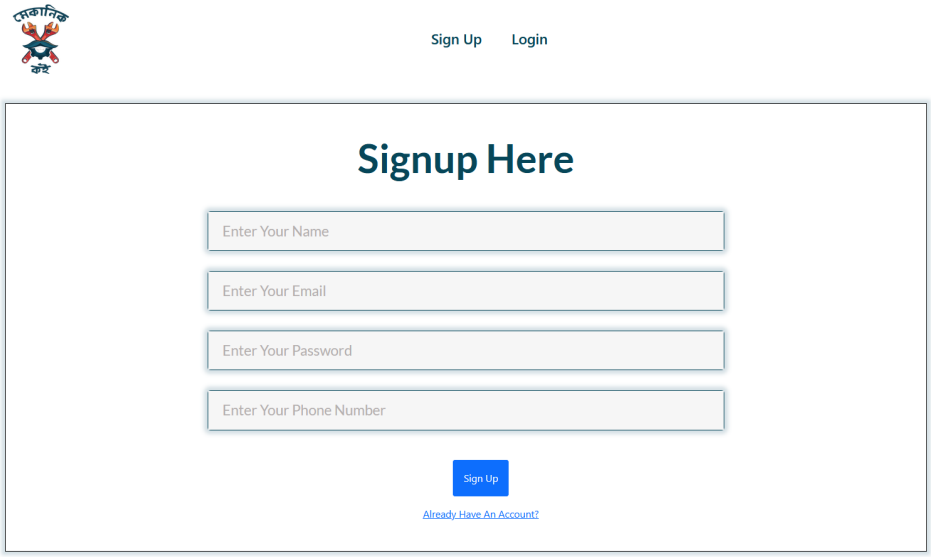


Figure 5.5: Sign Up Page

#### Login Page

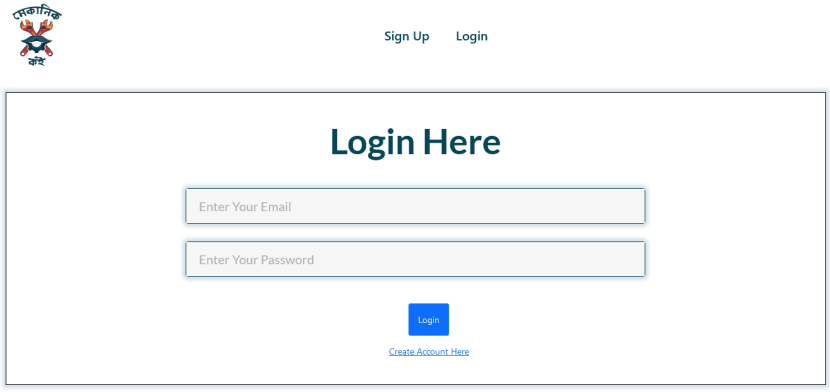


Figure 5.6: Login Page

### Add Client Page

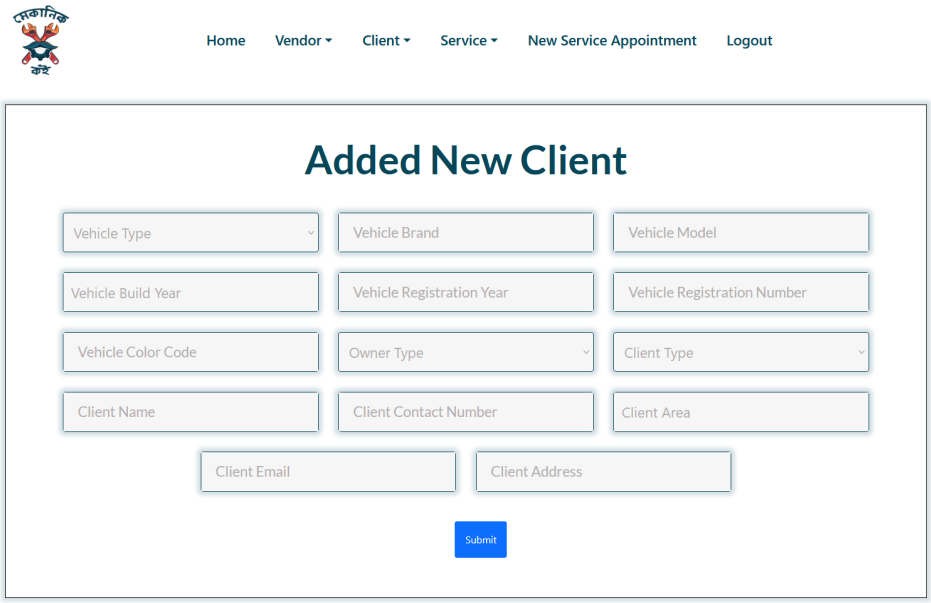


Figure 5.7: Add Client Page

### Add Service Page

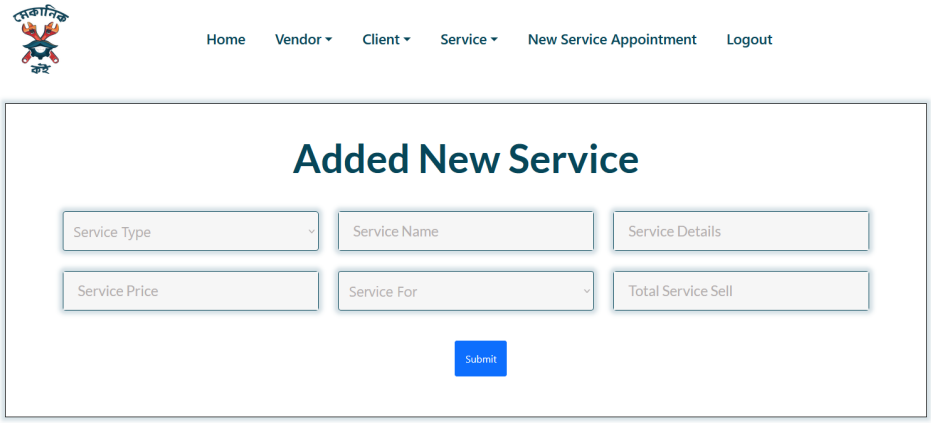


Figure 5.8: Add Service Page

### Add Service Appointment Page

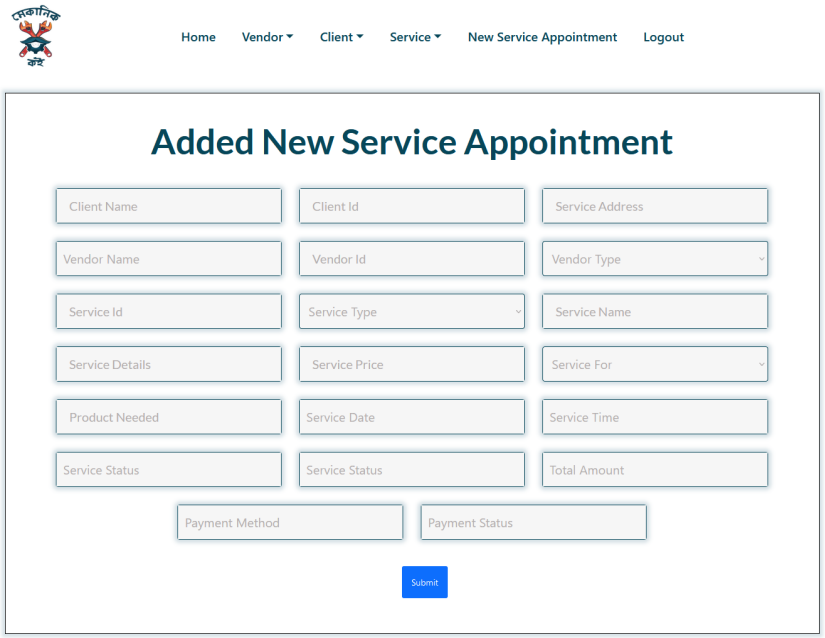


Figure 5.9: Add Service Appointment Page

## 5.4.2 Output

### Client List Output Page

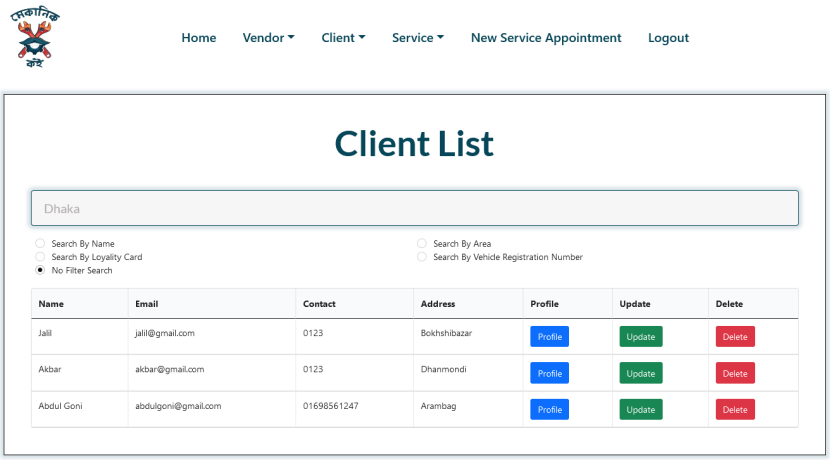


Figure 5.10: Client List Output Page

## Service List Output Page

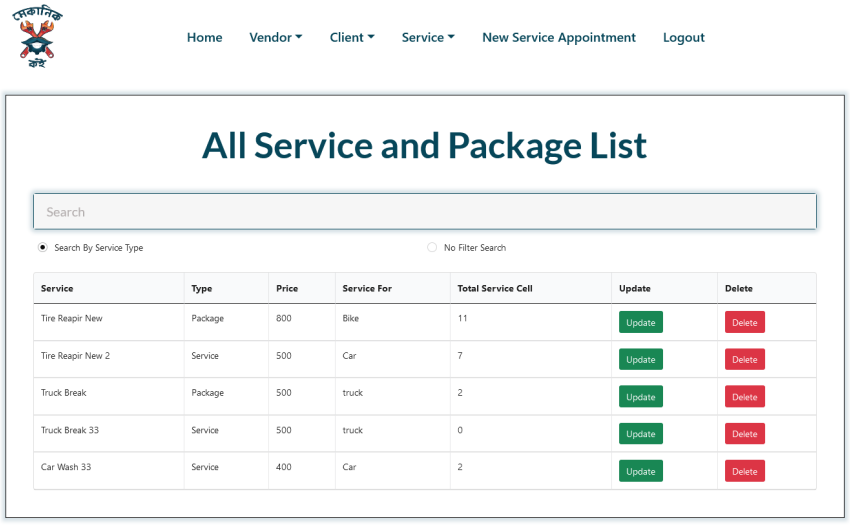


Figure 5.11: Service List Output Page

## Service Appointment Output Database

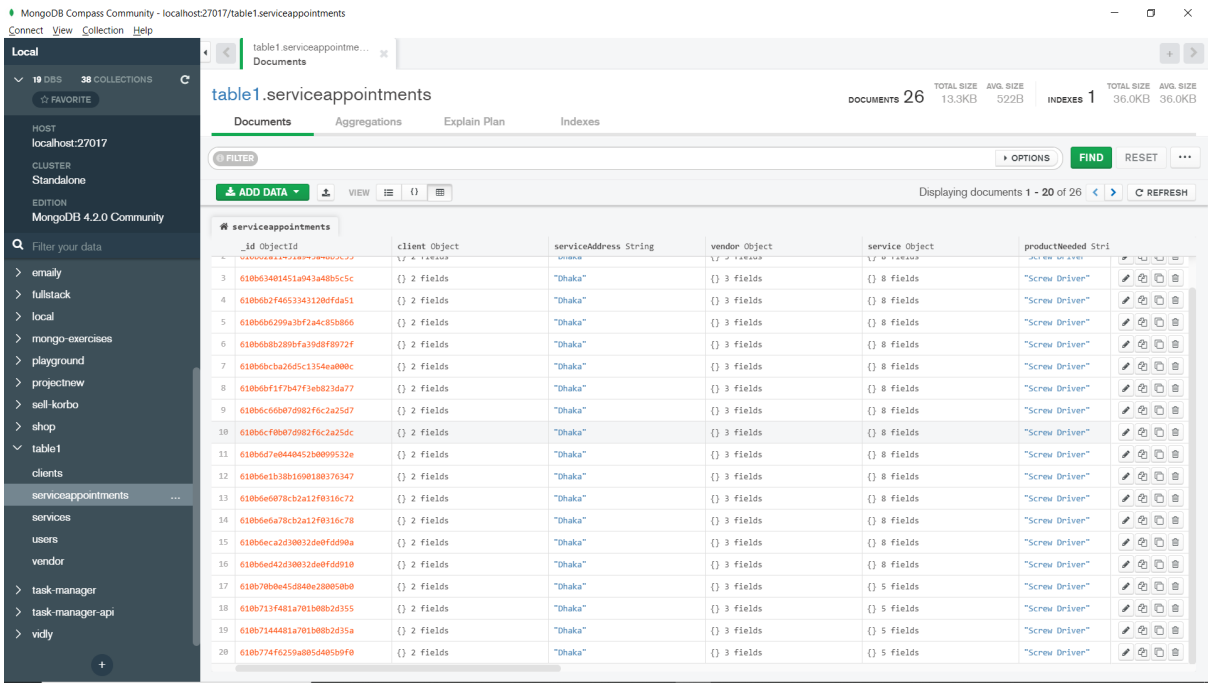


Figure 5.12: Service Appointment Output Database

### 5.4.3 Architecture

There are lots of architecture available for software development. To build this project we use the MERN stack. We use MongoDB for the database, Express js for the server-side framework, React js for the front-end part, and Node js as a javascript environment to build a back-end server. MERN is a full-stack. It actually follows the traditional 3-tier architectural pattern. Here React js is the front-end display tier, Express.js and Node.js is the backend application tier, and MongoDB is the database tier.

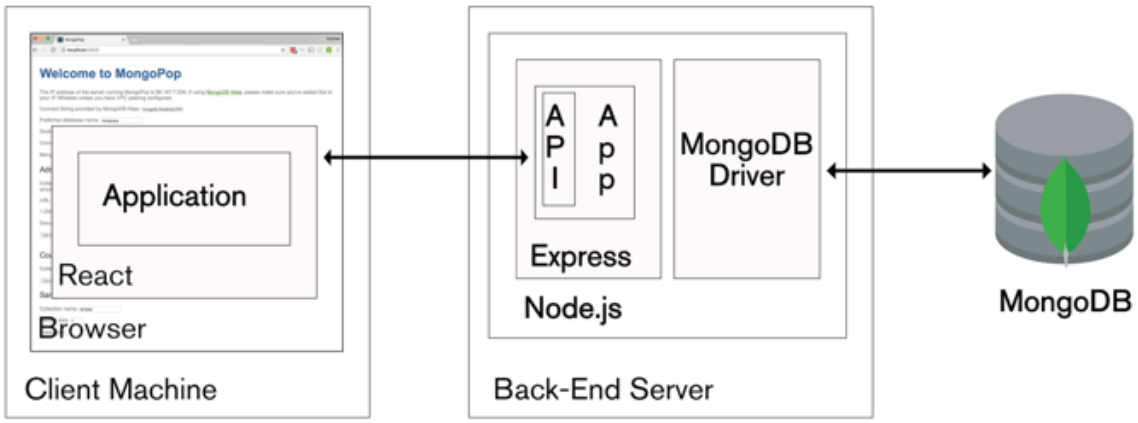


Figure 5.13: Traditional 3-tier Architecture

# Chapter 6

# Results & Analysis

Mechanic koi takes service orders from the internet, phone calls, email, and various sources. They are building CRM software to keep records of every client, vendor, and service. This software will only use Mechanic Koi management people or this software will be maintained by the admin panel. That's why other people cannot use this software. When someone hits their domain, they will land on the homepage.

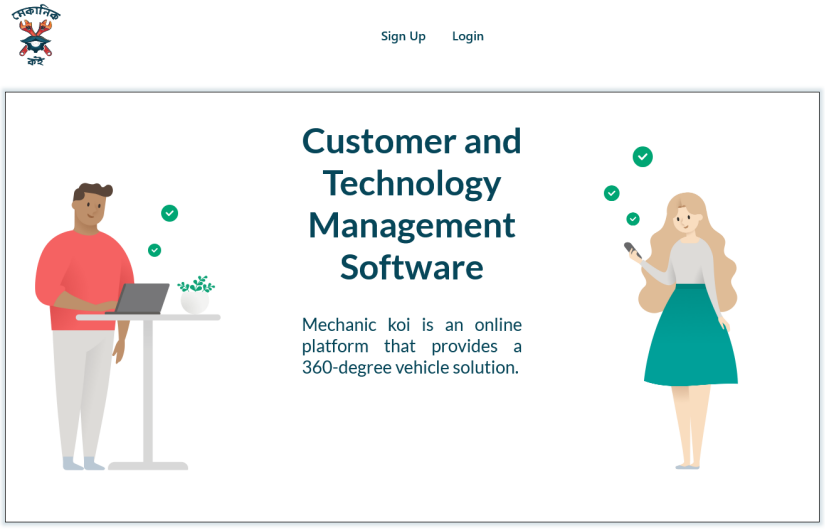


Figure 6.1: Home Page Before Login

Then the user will choose either to go to the Sign Up page or Login page. Here the Sign Up page will only be used by management people.

They will create an account for their employee, who is going to maintain this software. Management will create accounts and give employees their email and password. On the Login page, if the user's email and password are incorrect then the system will send them

a error message.

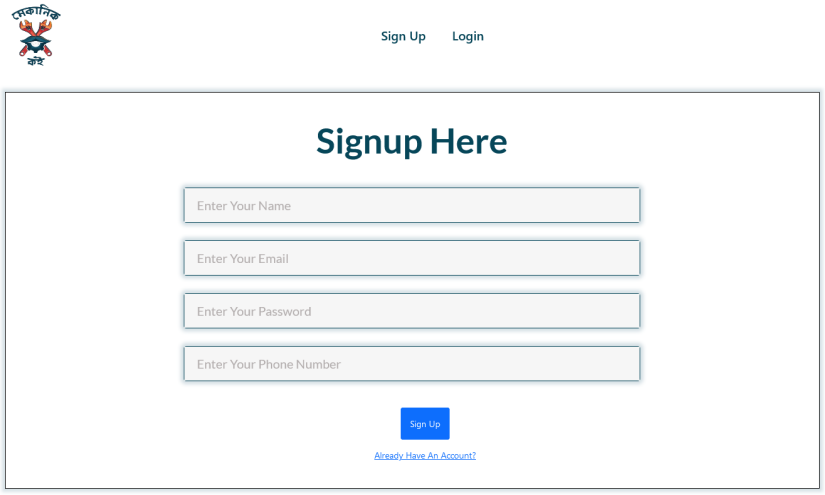


Figure 6.2: Sign Up Page

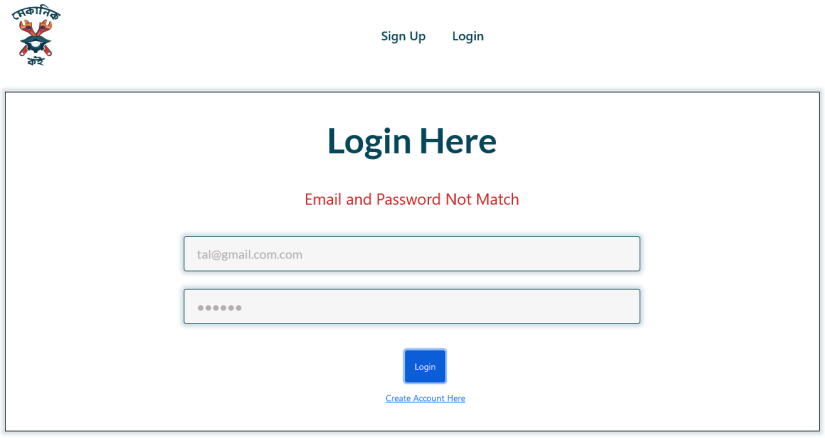


Figure 6.3: Login Page With Error Message

If the user completes their login, they will redirect to the Homepage.

Now users can create client, vendor, service, and service appointments. Also can see the client list, vendor list, service list, and service appointment list. Users can see client or vendor profiles and can update them.

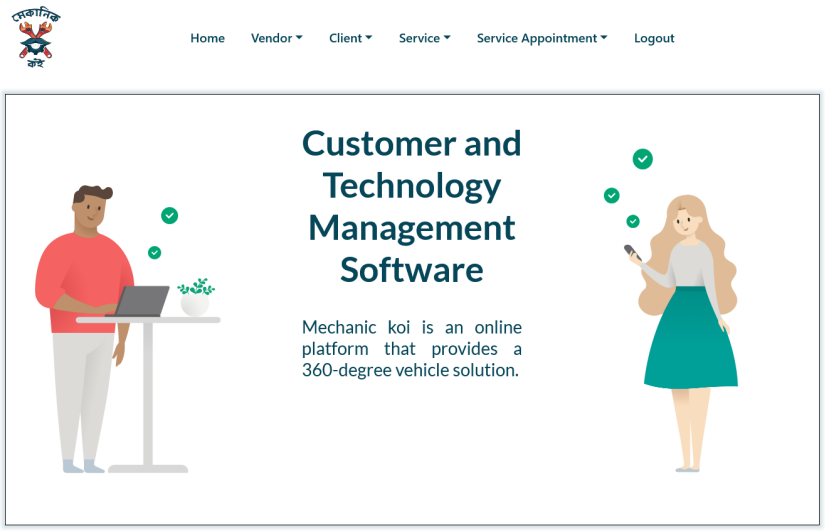


Figure 6.4: Home Page After Login

To create a client, users have to click on the Client option from the Navbar. A drop-down will appear. Then users have to choose the Added New Client option. A form will appear.

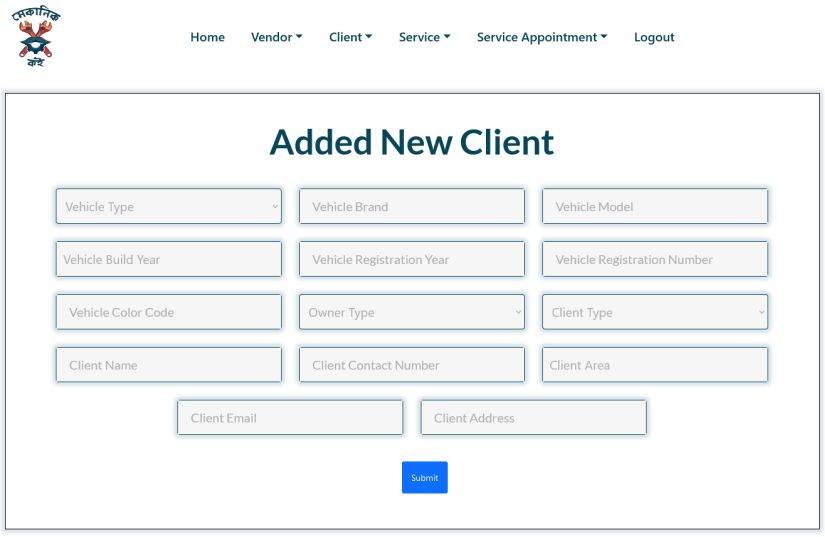


Figure 6.5: Added New client

Here users have to fill all the input fields. If any one of them is missing, the system will send an error message. When all input fields will complete, then a new client will be created. After completing this, the system will redirect users to the Homepage again.

Now users can see all client lists here. To see this, users have to click on the Client

from the Navbar, then choose the All Client List option. Users will see a table of clients with a search option.

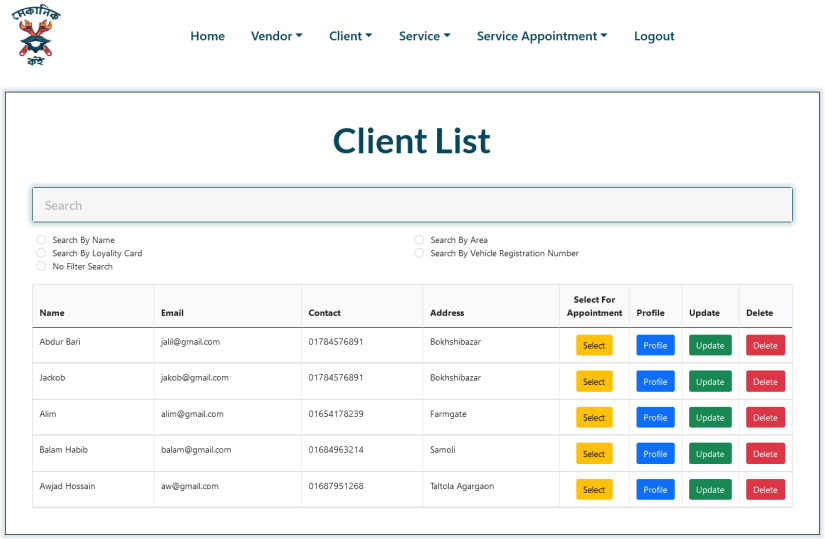


Figure 6.6: Client List

Users can search for a client here, by name, area, loyalty card, and vehicle registration number. In the Client List table, each row has a Select, Profile, Update, and Delete button. The Delete button can delete a client profile from the server. The Update button will redirect the user to the Client Profile Update page. Here any client data can be modified. The select button will select this client for creating a new service appointment. We will see this later.

The Profile button can redirect the user to the Client Profile. Here we will see the client profile, client vehicle profile, and all service appointments this user takes. Here users can easily track a client.

Now users can create a service or package. To create a service or package, users have to go to the Added New Service page from the navbar Service option. Then the user will see a form. Users have to fill them all in and submit them. If any input is missing, the system will send an error message. And if everything is fine, then the service will be created and the user will be redirected to the Homepage.


Figure 6.7: Client Profile

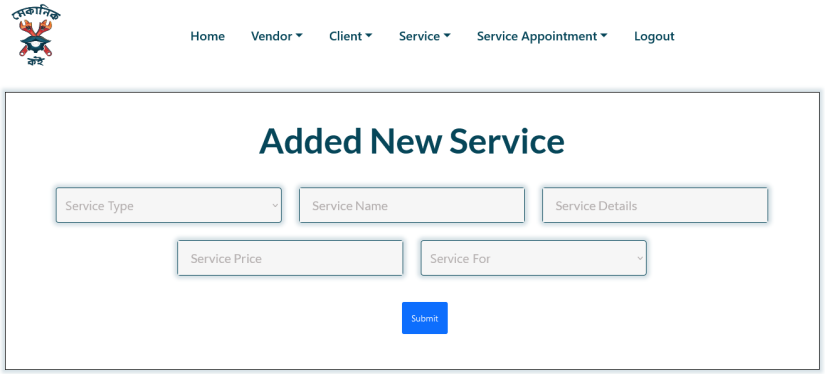


Figure 6.8: Create Service

Now the user can see the service list. Users should select Service from Navbar, then select All Service List and the user will be redirected to the service list page. Here users can see a service table and a search box. Users can search by service type here. In the table, the user can see the total sale of each service. By looking at it, users can find popular and less popular services easily.

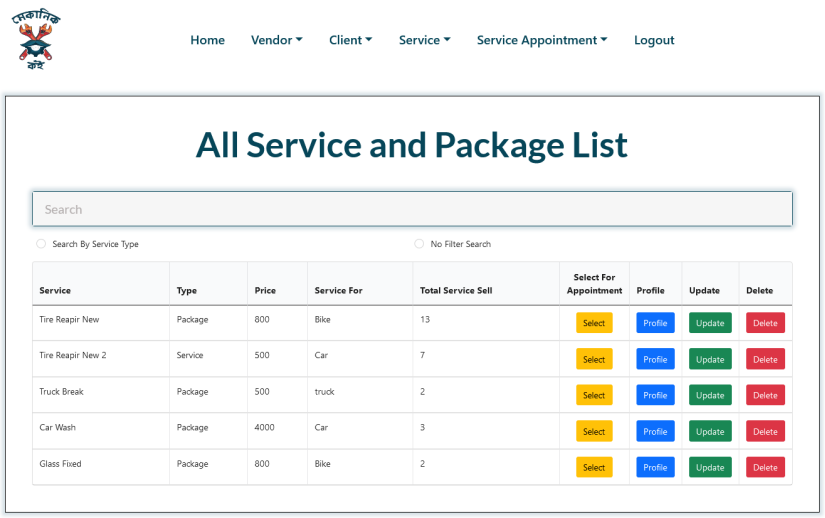


Figure 6.9: Service List

Here in the table, users also have four buttons. Select button, select this service for creating a new service appointment, Profile button will redirect to Service Profile page, Update button can redirect to Service Update page and lastly Delete button can remove service from the service list.

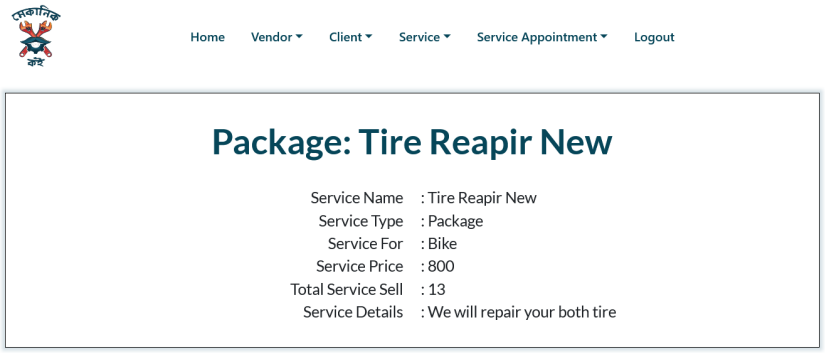


Figure 6.10: Service Profile

In the Vendor option, we can do the same thing as the Client and Service option. We can create a vendor, see the vendor list, can update it, delete it, can see the profile, and lastly can select it for the service appointment.

And lastly, users can create service appointments. This is the most tricky and most important part of this software. At first, users have to go to the Added New Service Appointment page.

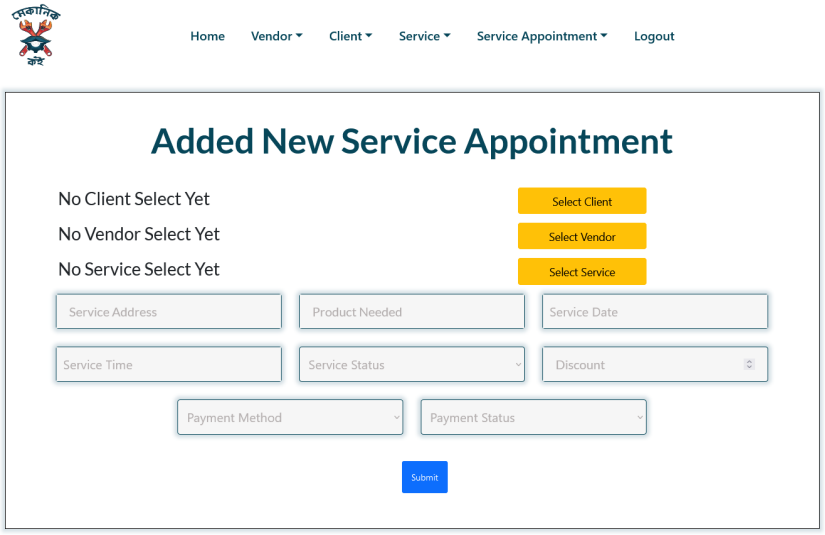


Figure 6.11: Service Appointment When Client Vendor And Service Not Select Yet

Now, no client, vendor, and service are selected. First, users have to click on "Select Client" button. Then the user will be redirected to the Client List page. Now from this table, the user has to select his targeted client. After selecting, the user will be redirected to the New service Appointment Page.

Now the user will see the client selected. Users can visit his profile from here. Also can change clients. Remember, only one client can be selected.

Now the user will click on the Select Service button. Then the user will be redirected to the Service List page. Again the user will select his service and redirect to the New Service Appointment page.

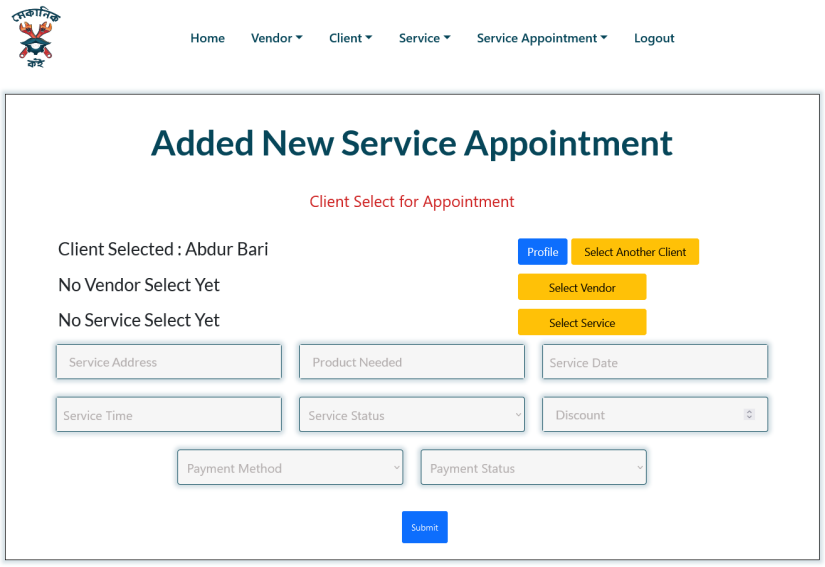


Figure 6.12: Service Appointment When Only Client Selected

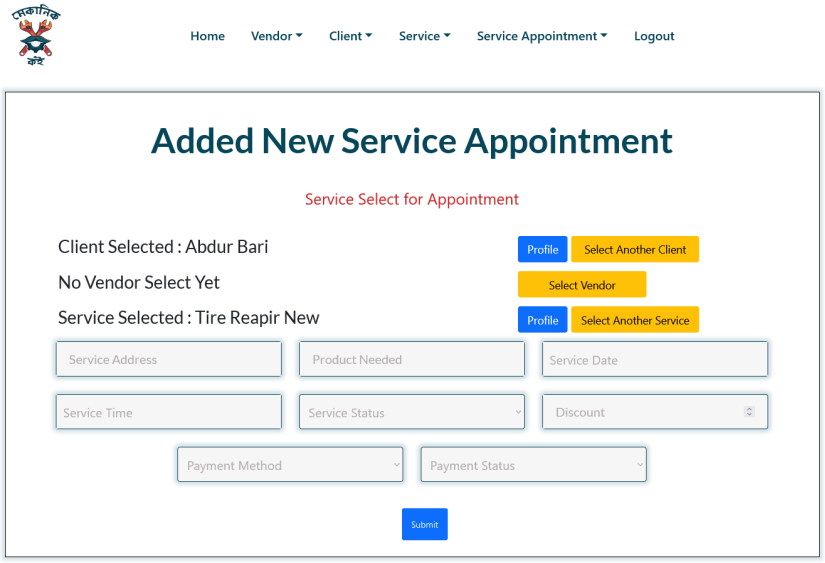


Figure 6.13: Service Appointment When Client and service Selected

Now client and service are selected. The vendor is still not selected. Now users have to click on the Select Vendor button and the same way select a vendor and redirect to the Added new service Appointment page.

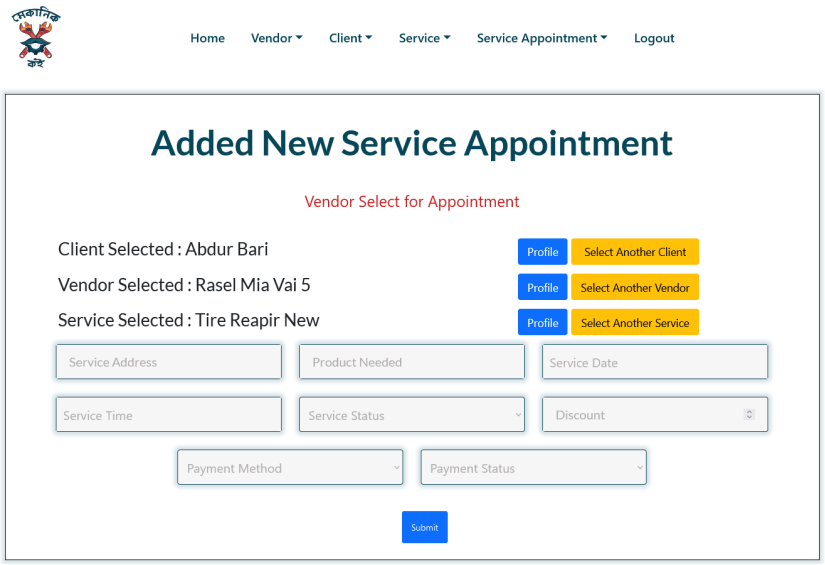


Figure 6.14: Service Appointment When Client Vendor and service Selected

Now all selected. Users just fill in the rest of the input field and submit it. If service appointments are created then the user will redirect to the Homepage.

Now users also can see Service Appointment List. Users first click on the Service Appointment option from Navbar, then select All Service Appointment List. Here users also see a table. All service appointments are listed here. From here, users can see service status, payment status, total price, service appointment order time and date, service appointment time and date, etc.

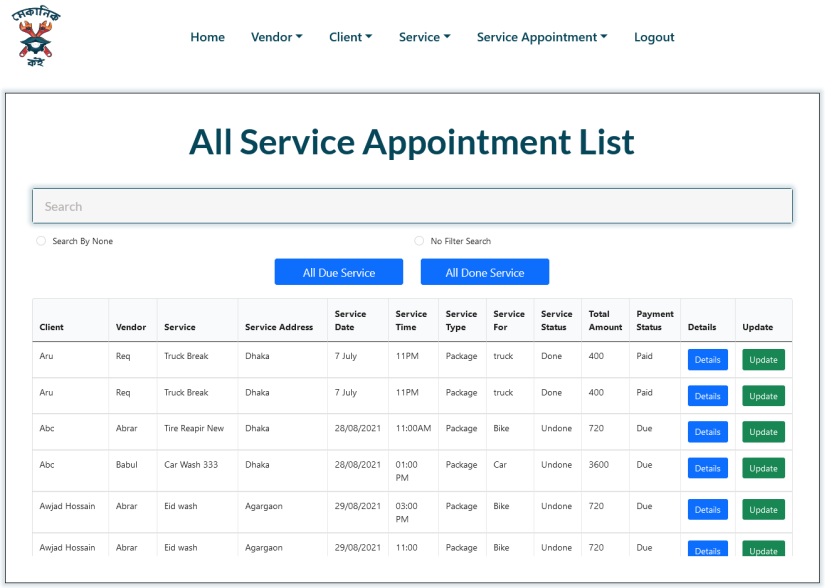


Figure 6.15: Service Appointment List

In this table, users will see two buttons. Update button can redirect users to the Update Service Appointment Page and give access to update Service Appointment. And lastly, the Details button allows the user to see each Service Appointment Details.

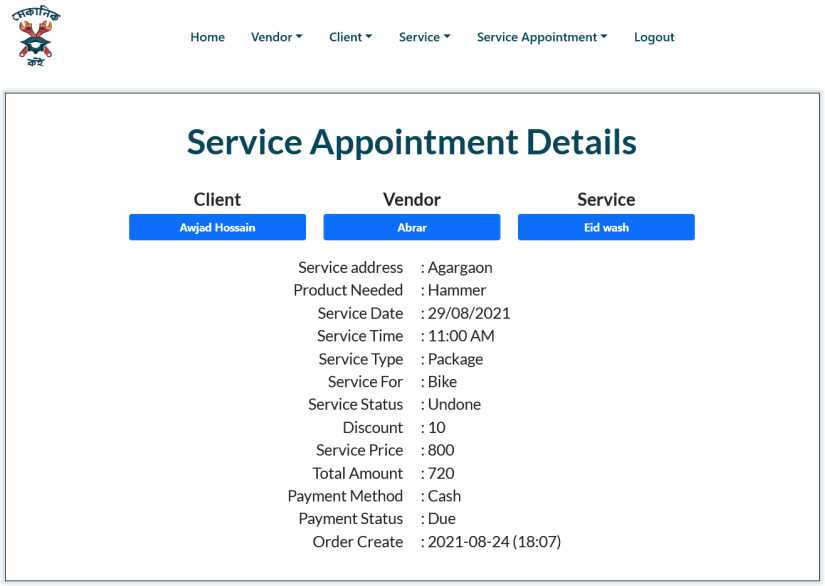


Figure 6.16: Service Appointment Details

This is all Mechanic Koi CRM software can do.

# Chapter 7

# Project as Engineering Problem Analysis

## 7.1 Sustainability of the Project/Work

Sustainability is the ability to be maintained at a specific rate or level. Sustainability in software development means maintaining and updating software from time to time, making the new feature as per market demand. Maintaining software is very important. Every software has bugs, and some bugs will appear in the future. We are aware of that. Mechanic Koi has a software development team to maintain their software at any time. They can fix bugs. If some bug is hard to handle by the software development team, then Mechanic Koi will contact an expert to solve this bug. Mechanic Koi has some software Development partners to help them any time.

Also, Mechanic Koi is concerned about updating their software from time to time as per market demand. To do that, the Mechanic Koi software development team will work. They first analyze the market and user demand. If they found something new, they will do a feasibility analysis to check if this feature is fitted with this software. If they find it fits perfectly, then this team will start work on it.

## 7.2 Social and Environmental Effects and Analysis

### 7.2.1 Social Effects

This software will create a good impact on our society. Our actual goal to build this software is to track client, vendor, and service data. So that we can make a change in our service quickly. And these things will create a significant impact on society. Mechanic Koi is a company that provides 360-degree vehicle solutions, any time anywhere. Clients

do no need to go to the car repair center for repair or maintenance. The Mechanic Koi vendor team will go to client's locations and help them with fixed client's problems. These things will save clients time and reduce hassle. Most of the time, the car repair center has no expert about cars. They have no professional education and experience. But Mechanic Koi always provides the best vendor with an academic degree, professional training, and practical experience. And that's why we believe this software can create a good impact on our society.

### 7.2.2 Environmental Effects

The environment is an important thing for us. We can't do any good for our society and business by damaging our environment. Mechanic Koi is aware of that. Mechanic Koi always works in an eco-friendly way. Mechanic Koi never launches a service that can damage our environment. Mechanic Koi vendors are highly concerned about the environment. They never do anything which can lead to environmental disaster. Mechanic Koi arranges a special program to make environmental awareness to the vendor so that the vendor can work in an eco-friendly way.

## 7.3 Addressing Ethics and Ethical Issues

Nowadays, cybercrime, hacking, and data breaching increase. Data security is the main issue. In this software, the Mechanic Koi team will collect and store so much data. That's why Mechanic Koi is concerned about it. They have a plan and roadmap to secure their data and protect it from other hackers and attackers.

At first, this software will deploy to a cloud server. Security in the cloud is very high. Every hosting cloud has its own security rules and regulation to protect it. Host cloud will manage it properly.

Second, all data will be stored in MongoDB atlas. Store data in MongoDB Atlas is a very wise decision. MongoDB Atlas is another cloud especially built for data storing and data protection. MongoDB Atlas is a database as a service or (DBaaS). It will protect our data at any cost.

Third, people who manage back-end servers and MongoDB atlas data are highly professional and skilled. They know when and what to do. They will protect this whole system at any cost.

# Chapter 8

# Lesson Learned

## 8.1 Problems Faced During this Period

Due to the Covid 19 pandemic, I did a remote internship. I work from home. Doing an office job from home is quite challenging. Communication with other colleagues is not simple. Internet speed in Bangladesh is not good, and connection breaks happen lots of times. That was a real problem. Also, I realize not all of my colleague's skills, knowledge, experience, and working patterns are similar. They all have different coding structures to write code. That was really problematic. I also find out some of them have no idea about Git Branching.

## 8.2 Solution of those Problems

Doing an internship from home was challenging. But I tried hard to complete my project. The Internet is the most essential for home internships. To avoid internet connection breaks and speed issues, I use both broadband and mobile data. If my broadband connection gets lost, my mobile data connects automatically. This way, I overcame a huge issue. My colleague and I use the Discord server to maintain our communication. We also arrange and attend meetings on the Discord server. Our skills and knowledge weren't the same. That's why we divide our work among ourselves. So that everybody can work independently.

# Chapter 9

# Future Work & Conclusion

## 9.1 Future Works

This software is evolving. The company will try to add new features to this software. But before that, the company will do some market research to understand user behavior. Only then, the company will go for the new major update. But the company can release some minor updates. Whenever a new bug arrives, the company needs to fix it and update it. The company has full preparation for this.

## 9.2 Conclusion

It was a wonderful experience to work with the Mechanic Koi team. During this internship, I learned a lot of things. I worked on this project as a full-stack developer. So I learned lots of things about the MERN stack. I learned how to connect MongoDB Express js, React Js, and Node js together. When development finished, it was a wonderful experience. In this internship, I work with a team. Working with a team is difficult but there is lots of fun as well. I learn how to maintain a good relationship and how to act as a leader. I also want to thank my internship supervisor for guiding me. He helps me at all times. He was always at my side when needed. Overall it's a wonderful experience and I will always remember this moment.

# Bibliography

[1] "Overview of mechanic koi." https://www.linkedin.com/company/mechanic-koi/about/. Accessed: 2021-08-06.

[2] R. Singh, "Top 5 Customer Relationship Management Examples.." https://www.appvizer.com/magazine/customer/client-relationship-mgt/customer-relationship-management-examples, 2020. Accessed: 2021-08-06.