**HIMALAYA DARSHAN COLLEGE**

**Tribhuvan University**

**Institute of Science and Technology**

**A Project Report On**

**" SAAJHA: *EVERY SERVICE TO YOUR DOORSTEP* "**

**A MOBILE APPLICATION TO SEARCH FOR SERVICES LOCALLY**

In Partial Fulfilment of Requirements for the Bachelor Degree in Computer Science and Information Technology (BSc. CSIT)

**Submitted to**

Department of Computer Science and Information Technology

Himalaya Darshan College

Biratnagar, Nepal

**Submitted by**

Anish Bhakta Joshi (21608/075)

Simran Sah (21622/075)

Date: 22nd Baisakh, 2080

**Himalaya Darshan College**

**Affiliated to Tribhuvan University**

# **Supervisor’s Certificate/Recommendation**

I hereby recommend that this project prepared under my supervision by Mr. Anish Bhakta Joshi and Ms. Simran Shah entitled **" SAAJHA: *EVERY SERVICE TO YOUR DOORSTEP* "** in partial fulfilment of the requirement for the degree of Bachelor of Science in Computer Science and Information Technology awarded by Tribhuvan University be processed for the evaluation.

**………………………..**

**Mr. Kushal Niraula**

Project Supervisor

Department of IT

Himalaya Darshan College

Main Road, Biratnagar-12

# **Letter of Approval**

This is to certify that this project is prepared by Mr. Anish Bhakta Joshi and Ms. Simran Shah entitled **" SAAJHA: *EVERY SERVICE TO YOUR DOORSTEP* "** in partial fulfilment of the requirement for the degree of Bachelor of Science in Computer Science and Information Technology awarded by Tribhuvan University has been well studied. In our opinion, it is satisfactory in the scope and quality of a project for the required degree.

|  |  |
| --- | --- |
| **Signature of the Supervisor** | **Signature of the HOD/Coordinator** |
| **Signature of the External Examiner** | **Signature of the Internal Examiner** |

# **Acknowledgement**

We would like to express our sincere gratitude to all those who have contributed to the completion of this final year project report. Without their support and assistance, this project would not have been possible.

Firstly, we would like to thank our project supervisor, **Mr. Kushal Niraula**, for his guidance, valuable suggestions, and constant encouragement throughout the project. His expertise and experience in the field have been instrumental in shaping our ideas and bringing them to fruition.

We are also grateful to the faculty members of Himalaya Darshan College, who have provided us with the necessary resources and facilities to carry out this project successfully. We would like to sincerely thank the Head of the Department of IT, **Er. Sumit Babu Shah**, for his support and cooperation.

We would also like to extend our appreciation to our colleagues and friends who have provided us with moral support and motivation to complete this project. Their assistance and feedback have been invaluable in refining our work.

Lastly, we would like to thank our families for their unconditional love and encouragement throughout our academic journey. Their support and belief in our abilities have been the driving force behind our success.

In conclusion, we sincerely thank everyone who contributed to the completion of this final year project report.

**Project Members**

Anish Bhakta Joshi (21608/075)

Simran Sah (21622/075)

# **Abstract**

The objective of this project is to develop a mobile application that gives service providers a facility to showcase their services by registering themselves at no cost and recommending the nearest service provider to the customer by the implementation of the Haversine formula.

The project involved several discussions, meeting with the supervisors and so many days of coding and debugging. We first decided to analyse similar existing projects and found out ways to develop the project which would be better. In addition, this project includes flutter technology for the development with Dart programming as front-end, Nodejs as back-end, and Mongo DB as the database. The project was completed in around 3 months.

The results of the project demonstrate how home services like cleaning, carpeting, plumbing, etc. are delivered to the doorstep with just one click. This discusses the main theme of online home services and registering yourself as a worker in your field and providing an authenticated and authorized login module for the users such as service seekers, service providers, and the admin, by providing appropriate credentials at the time of registration. This project provides the facilities to the customer such as registration, display profile of service provider, map navigation, etc. These results have significant implications for both service providers as well as service seekers.

The project has several limitations which provide directions for future research. The key limitations are IOS-incompatible, no payment integration, no feedback to the service providers, and only live location detection.

Overall, the project has provided valuable insights into the possibility of changing the traditional method of seeking services in a new, modern, and advanced way. The project has the potential to make a significant impact on the current trend of seeking services.

**Keywords:** Authenticated, Credentials, Dart, Detection, Flutter, Mongo DB, Nodejs.

**Table of Contents**

**Title Page No**

**[Supervisor’s Certificate/Recommendation](#_Toc134205603)** [ii](#_Toc134205603)

**[Letter of Approval](#_Toc134205604)** [iii](#_Toc134205604)

**[Acknowledgment](#_Toc134205605)** [iv](#_Toc134205605)

**[Abstract](#_Toc134205606)** [v](#_Toc134205606)

**[Table of Contents](#_Toc134205606)** [v](#_Toc134205606)i

**[List of Abbreviation](#_Toc134205607)** [viii](#_Toc134205607)

**[List of Tables](#_Toc134205608)** [ix](#_Toc134205608)

**[List of Figures](#_Toc134205609)** [x](#_Toc134205609)

**[CHAPTER 1 : INTRODUCTION](#_Toc134205610)** [1](#_Toc134205610)

[1.1 Introduction 1](#_Toc134205611)

[1.2 Problem Statement 1](#_Toc134205612)

[1.3 Objectives 2](#_Toc134205613)

[1.4 Scope and Limitations 2](#_Toc134205614)

[1.5 Development Methodology 2](#_Toc134205615)

[1.6 Report Organization 3](#_Toc134205616)

**[CHAPTER 2 : BACKGROUND STUDY AND LITERATURE REVIEW](#_Toc134205617)** [5](#_Toc134205617)

[2.1 Background Study 5](#_Toc134205618)

[2.2 Literature Review 5](#_Toc134205619)

**[CHAPTER 3 : SYSTEM ANALYSIS](#_Toc134205620)** [7](#_Toc134205620)

[3.1 System Analysis 7](#_Toc134205621)

**[3.1.1 Requirement Analysis](#_Toc134205622)** [7](#_Toc134205622)

**[3.1.2 Feasibility Analysis](#_Toc134205623)** [9](#_Toc134205623)

**[3.1.3 Analysis](#_Toc134205624)** [11](#_Toc134205624)

**[CHAPTER 4 : SYSTEM DESIGN](#_Toc134205625)** [14](#_Toc134205625)

[4.1 Design 14](#_Toc134205626)

[4.2 Algorithm Details 16](#_Toc134205627)

**[CHAPTER 5 : IMPLEMENTATION AND TESTING](#_Toc134205628)** [18](#_Toc134205628)

[5.1 Implementation 18](#_Toc134205629)

**[5.1.1 Tools Used](#_Toc134205630)** [18](#_Toc134205630)

**[5.1.2 Implementation Details of Modules](#_Toc134205631)** [19](#_Toc134205631)

[5.2 Testing 21](#_Toc134205632)

**[5.2.2 System Testing](#_Toc134205633)** [23](#_Toc134205633)

**[CHAPTER 6 : CONCLUSION AND RECOMMENDATION](#_Toc134205634)** [25](#_Toc134205634)

[6.1 Conclusion 25](#_Toc134205635)

[6.2 Future Recommendation 25](#_Toc134205636)

**[REFERENCE](#_Toc134205637)** [26](#_Toc134205637)

**[APPENDICES](#_Toc134205638)** [27](#_Toc134205638)

# **List of Abbreviations**

**Abbreviations Full Form**

GUI Graphical User Interface

SQL Structured Query Language

UML Unified Modelling Language

UI User Interface

VS Visual Studio

IDE Integrated Development Environment

SSD Solid State Drive

# **List of Tables**

**Table Page No**

[Table 1: Gantt Chart 10](#_Toc134208136)

[Table 2: Registration Testing 22](#_Toc134208137)

[Table 3: Login Testing 22](#_Toc134208138)

[Table 4: View and Edit User’s Details Testing 23](#_Toc134208139)

[Table 5: Search and Algorithm Testing 23](#_Toc134208140)

[Table 6: System Testing 24](#_Toc134208141)

# **List of Figures**

**Figure Page No**

[Figure 1: Agile Development 3](file:///C:\\Users\\TUF%20Gaming\\Desktop\\Report.docx" \l "_Toc134208813)

[Figure 2: Use Case Diagram of Saajha 8](file:///C:\\Users\\TUF%20Gaming\\Desktop\\Report.docx" \l "_Toc134208814)

[Figure 3: Class Diagram of Saajha 11](file:///C:\\Users\\TUF%20Gaming\\Desktop\\Report.docx" \l "_Toc134208815)

[Figure 4: Object Diagram of Saajha 12](#_Toc134208816)

[Figure 5: Sequence Diagram of Saajha 13](#_Toc134208817)

[Figure 6: MVC Architecture 14](#_Toc134208818)

[Figure 7: Component Diagram of Saajha 15](file:///C:\\Users\\TUF%20Gaming\\Desktop\\Report.docx" \l "_Toc134208819)

[Figure 8: Deployment Diagram for Saajha 16](#_Toc134208820)

[Figure 9: Flowchart of Service Seeker 19](#_Toc134208821)

[Figure 10:: Flowchart of Service Provider 20](#_Toc134208822)

[Figure 11: Flowchart of Admin 21](#_Toc134208823)

[Figure 13: DIO 28](file:///C:\\Users\\TUF%20Gaming\\Desktop\\Report.docx" \l "_Toc134208824)

[Figure 12: API Routes 28](#_Toc134208825)

[Figure 14: Validation 29](#_Toc134208826)

[Figure 15: Algorithm Implantation 29](#_Toc134208827)

# **CHAPTER 1** **: INTRODUCTION**

## Introduction

“**SAAJHA:** *every service to your Doorstep*” is a very dynamic and very easy-to-understand Android application. It is known globally that, in today’s market, it is extremely difficult to start a new small-scale business and live through the competition from well-established and settled owners. In a fast-paced time of today, when everyone is squeezed for time, the majority of people are finicky when it comes to receiving and giving service.

This application provides the user to search the home services job and get the facilities of every kind of service like plumbing, carpenter, laundry, and so on to their doorstep. The interface of the Android application is very easy and anybody can easily work on it. Customers nowadays want the most simple, effective, and smooth work-flow. This application provides them to meet their need.

The Android application is also very useful because the customer doesn’t have to visit the service provider’s office, he/she can easily book his/her order via this application and he/she can also pay the payment online through this Android application. So, he/she can book an order without any kind of disturbance. It will provide the choice of choosing the workers whom they wanted to hire. From the viewpoint of registering as a worker, the workers who want to register themselves in the application can fill up the simple details about themselves and can accept the work through the admin.

To make this application work successfully we have used some latest technology such as Android for the platform, and Flutter, Dart, Nodejs, and Mongo DB for the development.

## Problem Statement

For finding any sort of household services regarding electricity, construction, home decors, etc., we have to first find specific personnel relating to that particular field which can be achieved either by calling them in case they are familiar or by seeking help from our neighbours and relatives. After finding the right one, we again need to bargain about the service charges. In this process, we face many problems as it is time-consuming and manual. On the other hand, many service providers are also deprived of opportunities as many households often seek service from their familiar and regular service providers. There can be miscommunication which may lead to unpleasant service. Hence, to solve this issue, “**SAAJHA:** *every service to your doorstep*”, is proposed and originally designed for every household and service provider.

This system greatly simplifies the household service-seeking process for both the service provider and service seekers. This system will also provide a fixed service charge to the service provider based on their experience and the feedback they receive.

The challenges encountered in the present system are a major drawback to the realization of efficiency and satisfaction. There isn’t proper feedback and service charge evaluation which results in difficulty in identifying the proper service provider. Service providers also don’t have the registration option and services are assigned manually after the order is placed.

## Objectives

The objectives of this project “**SAAJHA:** *every service to your doorstep*” is as follows:

* To let the service providers and seekers register and log in themselves.
* To recommend the nearest service provider to the service seekers.

## Scope and Limitations

**Scope**

* The application provides a convenient way to book any sort of service from a mobile phone.
* It creates an opportunity for service providers to showcase their skills.
* It provides fast and reliable service.

**Limitation**

* The application is only an Android-based application, not an IOS-based application. So, IOS users might not have the privilege to use the application.
* Users need the internet and suitable devices to make this application useful.

## Development Methodology

Our project had team collaboration and uncertain and changing project scope, so we followed the Agile Development Methodology for the development of the project. The agile development methodology is an iterative and incremental approach that focuses on flexibility, collaboration, and rapid delivery. It involves planning, short sprints, daily check-ins, continuous delivery, sprint review and retrospective meetings, and iteration. Agile development is an iterative process, with the team refining and improving the project backlog and delivery process over time, based on feedback from the supervisor.

Figure 1: Agile Development

## 1.6 Report Organization

On completion of our project development, we have documented the milestone and the final document report and they have been organized under the following chapters:

Chapter 1: Introduction

Chapter 2: Background Study and Literature Review

Chapter 3: Requirement Analysis and Feasibility Analysis

Chapter 4: System Design

Chapter 5: Implementation and Testing

Chapter 6: Conclusion and Future Recommendations

**The First Chapter** consists of a brief introduction about our project title giving a broad idea about the behind our project and a problem of definition. It consists of the objectives, scope, and limitations of the project.

**The Second Chapter** consists of a background study and literature review giving a short description of the papers and the articles that we have gone through.

**The Third Chapter** consists of the use case diagram defining the functional requirements of the system, a feasibility analysis describing the economical, technical, and operational feasibility of the system, and a Gantt chart describing the time that was taken to achieve the different tasks and other diagrams.

**The Fourth Chapter** describes the detail of the database design, interface design, input-output design, and process design.

**The Fifth Chapter** contains information about the implementation tools used, different types of testing done, and test cases used to test the developed system.

**The Sixth Chapter** consists of the summary of the entire project and the things that we learned on the completion of the project.

# **CHAPTER 2** **: BACKGROUND STUDY AND LITERATURE REVIEW**

## 2.1 Background Study

To obtain household services such as carpenter, plumber, construction, home décor, etc., one typically needs to locate a specialist in the relevant field. This can be achieved either by contacting someone known to provide the service or by asking friends and family for recommendations. Once a suitable service provider is found negotiations may be required to settle on a price for the service. This process can be time-consuming and labour-intensive and may result in problems.

But this traditional paradigm of searching for services and service providers has been affected by new media technologies which make it easier. Press announcements and bulletin boards or store walls have been left behind over the internet and unlimited social network options connect service providers with people that need them. [1]

## 2.2 Literature Review

There are now several digital platforms available that provide an online marketplace for household services. These platforms connect service providers with potential customers, making it easier for people to find the services they need and for service providers to reach a wider audience.

Some of the popular platforms include Handy, HomeAdvisor, Sajilo Sewa, Bhetayo, GharSewa, etc. This allows users to search for specific services and choose the best service providers.

**Handy:** Handy is the leading platform for connecting individuals looking for household services with independent service professionals. From home cleaning to handyman services, Handy instantly matches thousands of customers every week with top-rated professionals in cities all around the world. With a seamless 60-second booking process, secure payment, and backed by the Handy Happiness Guarantee, Handy is the easiest, most convenient way to book home services. [2]

**Home Advisor:** HomeAdvisor is an online marketplace that connects homeowners with local service professionals for home-related tasks such as home repair, remodeling, and maintenance. The platform allows users to search for services in their area. Homeowners can also receive quotes from multiple service professionals and compare prices before choosing a service provider. HomeAdvisor aims to simplify the process of finding home service professionals and provide a reliable and trustworthy platform for home-owners and service providers alike. [3]

**Sajilo Sewa:** Sajilo Sewa is an online service marketplace based in Nepal that connects customers with a wide range of service providers for tasks such as cleaning, plumbing, electrical work, and home repairs. The platform was launched in 2016 and aims to simplify the process of finding reliable and affordable service providers in Nepal. Sajilo Sewa offers a user-friendly website and mobile app that enables customers to search for services in their area and compare prices and reviews from different service providers. Service providers can also create profiles on the platform and offer their services to potential customers. [4]

**Bhetayo:** Bhetayo.com is an online marketplace based in Nepal that connects customers with service providers for a variety of services, including home maintenance, cleaning, plumbing, and electrical work. The platform aims to provide a simple and convenient way for Nepalese customers to find reliable service providers. Bhetayo.com enables customers to search for services in their area, compare prices and reviews from different service providers, and book appointments online. [5]

# **CHAPTER 3 : SYSTEM ANALYSIS**

## 3.1 System Analysis

### **3.1.1 Requirement Analysis**

The project's requirement analysis impacts the outcome. Thus, one of the most crucial stages of application development is this one. The specifications for this application must serve the needs of the general public and society.

**Software Requirement**

Operating System: Windows 10 (or above), Android

Frontend: DART, Flutter

Backend: NodeJS, Express JS

Database: Non-Relational Database (MongoDB)

Development Toolkit: Visual Studio Code

**Hardware Requirement:**

Minimum: 4GB RAM

Processor: Intel Core i5 10th Generation (or above),

512 GB SSD (or above)

#### **i. Functional Requirement**

This home service-providing application has many modules, which make the software more efficient and user-friendly. The modules make the maintenance of the database easier.

The different modules in this project are described below:

* + - Admin
    - Service Provider
    - Service Seeker

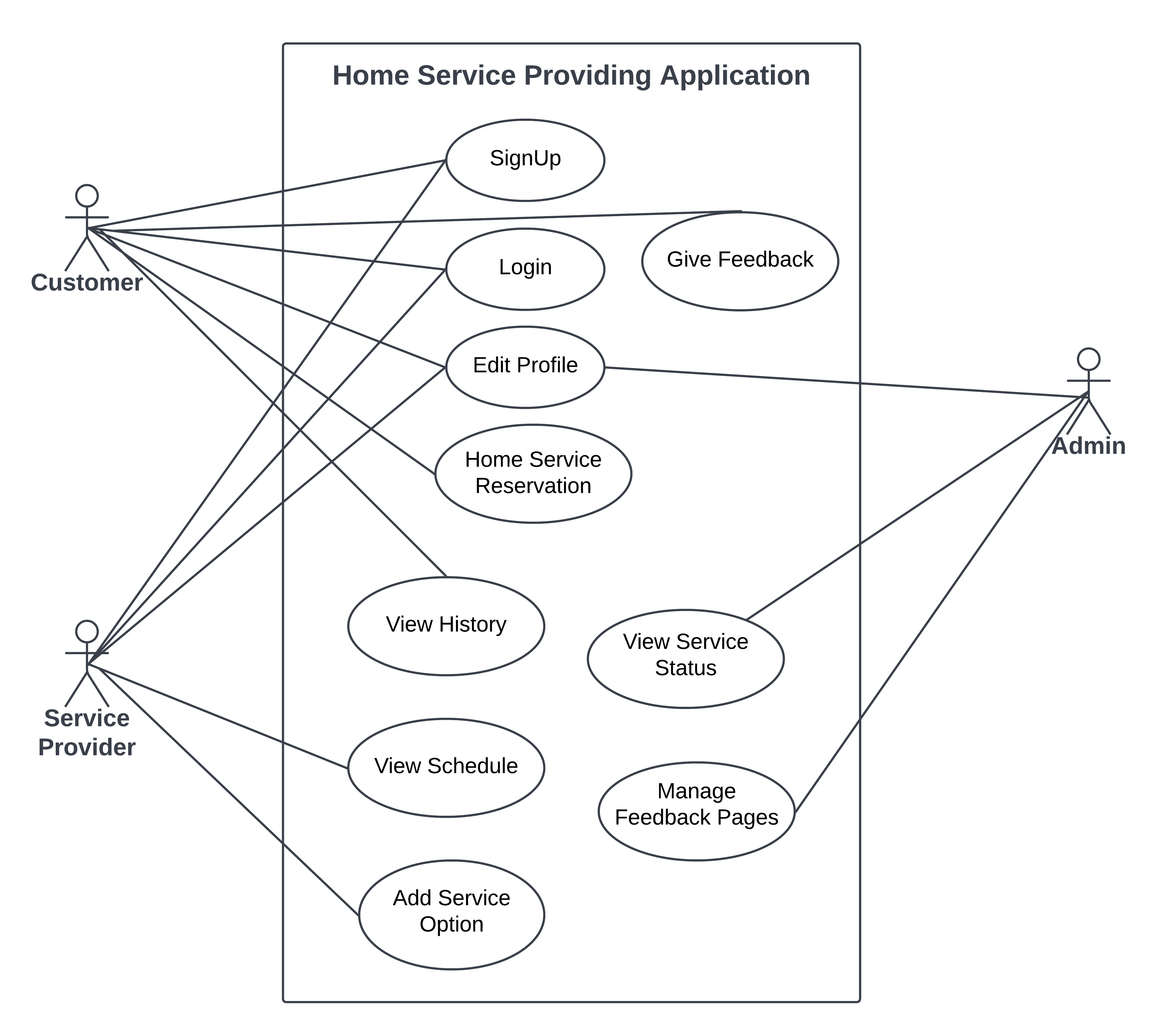


Figure 2: Use Case Diagram of Saajha

#### **ii. Non-Functional Requirement**

Non-Functional requirements are the requirements that define the standards and criteria that should be achieved through the application. Some non-functional requirements of the application are:

* **Speed:** The application is speed and requests/responses to/from the server are quick.
* **Usability:** The application is easy to use for new users.
* **Standards:** Strict coding standards are followed during the development.
* **Scalability:** The application can be scaled with the increase in number of users. For this database tuning, query optimization and other processes might be required.
* **Efficiency:** The application is highly efficient in its design and implementation for providing a good user experience.

### **3.1.2 Feasibility Analysis**

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of an existing or proposed venture, opportunities and threats present in the environment, the resources required to carry through, and ultimately the prospects for success. A feasibility study is a process of determining whether or not a project is worth doing. The feasibility study helps us to decide how to proceed with the project. Some of the feasibility studies include:

#### **i. Technical Feasibility**

In Technical Feasibility, current hardware and software resources and required technology are analyzed/assessed to develop the project. This technical feasibility study reports on whether there exists correct required resources and technologies which will be used for project development.

It is the measure of the availability of a specific technical solution and the availability of technical resources and expertise.

* Node.js is used for the backend.
* The proposed system uses MongoDB, a NoSQL database.
* Dart is used for the front end.

#### **ii. Economic Feasibility**

In Economic Feasibility, the cost and benefit of the project are analyzed. A detailed analysis is carried out what will be the cost of the project for development which includes all required costs for final development such as hardware and software resources required, design and development costs and operational costs, and so on. After that, it is analyzed whether the project will be beneficial in terms of finance for us or not.

This system is economically feasible since the cost involved for both hardware and software are approachable. The majority of people are already in possession of smartphones, commonly Android phones. So, there is no additional cost on the general user’s side. Hence, this project is economically feasible.

#### **iii. Operational Feasibility**

In Operational Feasibility, the degree of providing service to requirements is analyzed along with how easy the product will be to operate and maintain after deployment. Along with this, other operational scopes are determining the usability of the project, determining suggested solution by the software development team is acceptable or not, etc.

* Since the application is very easy to understand, one session of training is enough to use and understand it.
* Simple and easy-to-use GUI.

#### **iv. Schedule Feasibility**

The schedule is the estimation of the duration of the project and the work division for carrying out the project work. A project is said to have failed if it takes too long to complete or is completed after its deadline.

In a Schedule Feasibility Study, mainly timelines/deadlines are analysed for the proposed project which includes how much time teams will take to complete the final project which has a great impact on the organization as the purpose of the project may fail if it can’t be completed on time.

Table 1: Gantt Chart

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.N.** | **Task Name** | **4 Months (120 Days)** | | | | | |
| **0 - 20** | **20 - 40** | **40 - 60** | **60-80** | **80-100** | **100-120** |
| 1. | Planning & Analysis |  | | | | | |
| 2. | Design |  | | | | | |
| 3. | Implementation |  | | | | | |
| 4. | Testing |  | | | | | |
| 5. | Documentation |  | | | | | |

### **3.1.3 Analysis**

**Object Modeling**

Object modeling is a modeling approach for software modeling and designing. It was developed basically as a method to develop object-oriented systems and to support object-oriented programming. It describes the static structure of the system.

**Class Diagram**

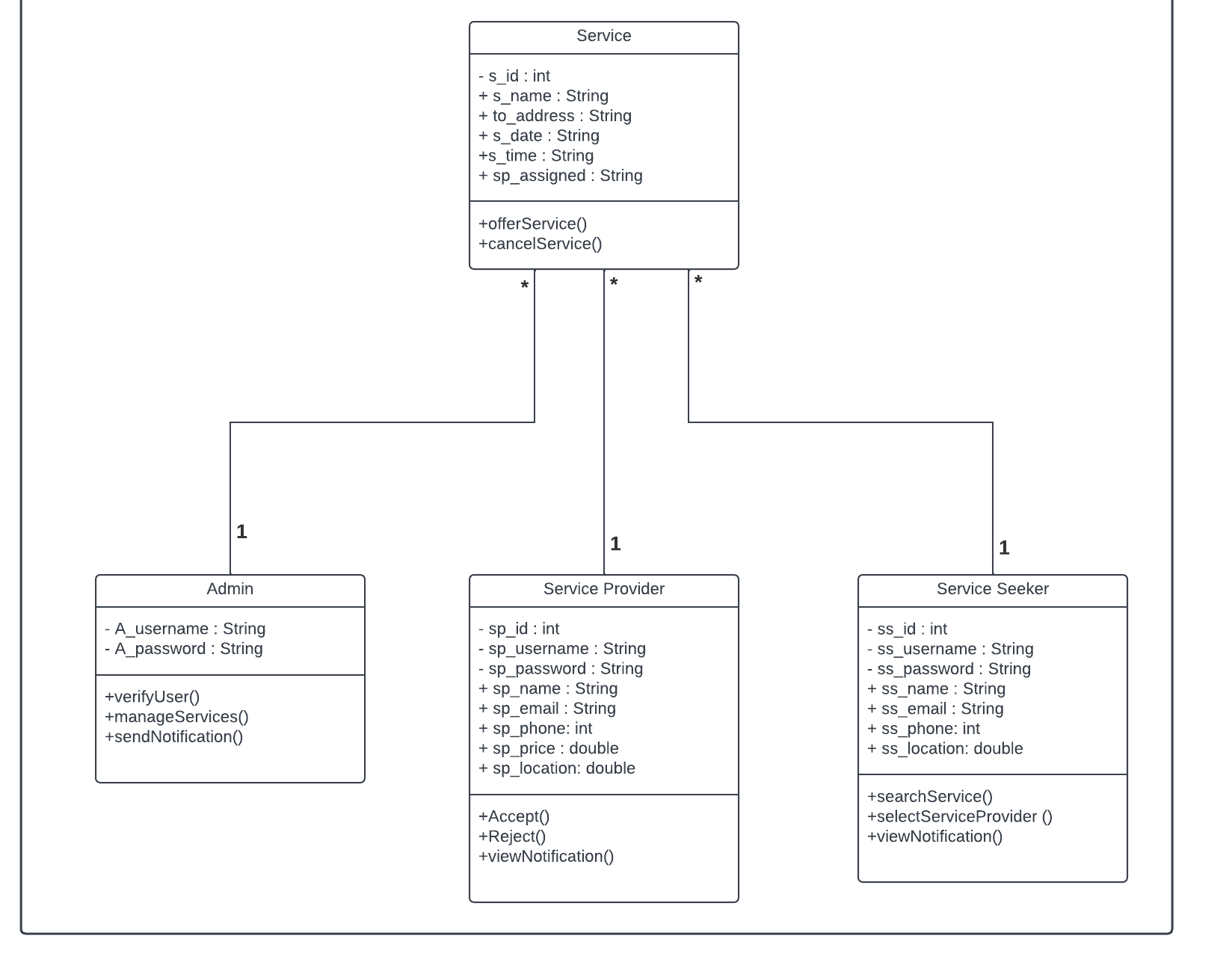
 A class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Figure 3: Class Diagram of Saajha

**Object Diagram**

An object diagram shows this relation between the instantiated classes and the defined class, and the relation between these objects in the system. They are useful to explain smaller portions of your system when the system class diagram is very complex, and also sometimes model recursive relationships in the diagram

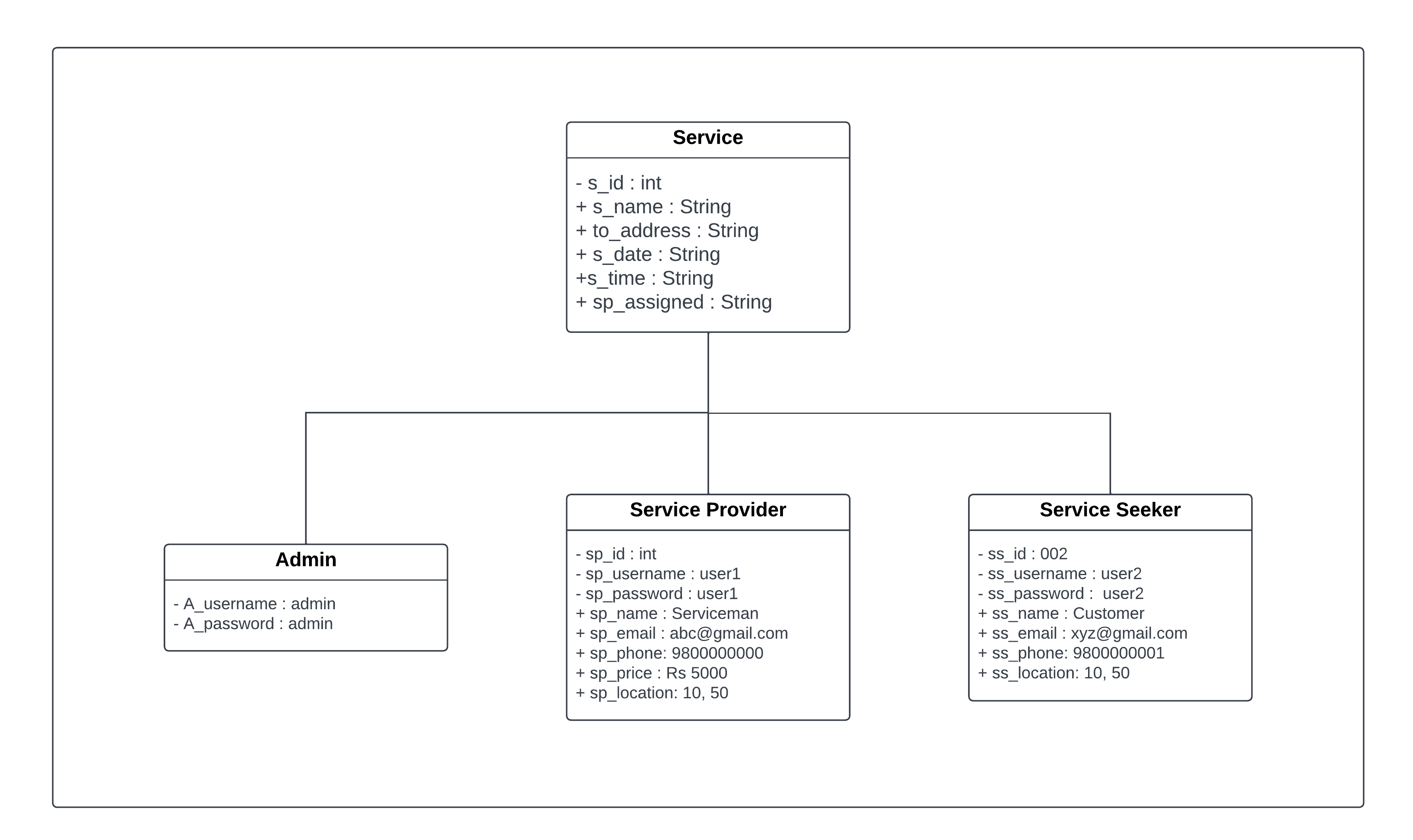


Figure 4: Object Diagram of Saajha

**Dynamic modelling**

Dynamic Modelling describes those aspect of the system that are concerned with time and sequencing of the operations. Dynamic model is represented graphically with the help of state diagrams. State diagram relates with events and states. Events represents external functional activity and states represents values objects.

**Sequence Diagram**

Sequence diagrams in UML specifically focus on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

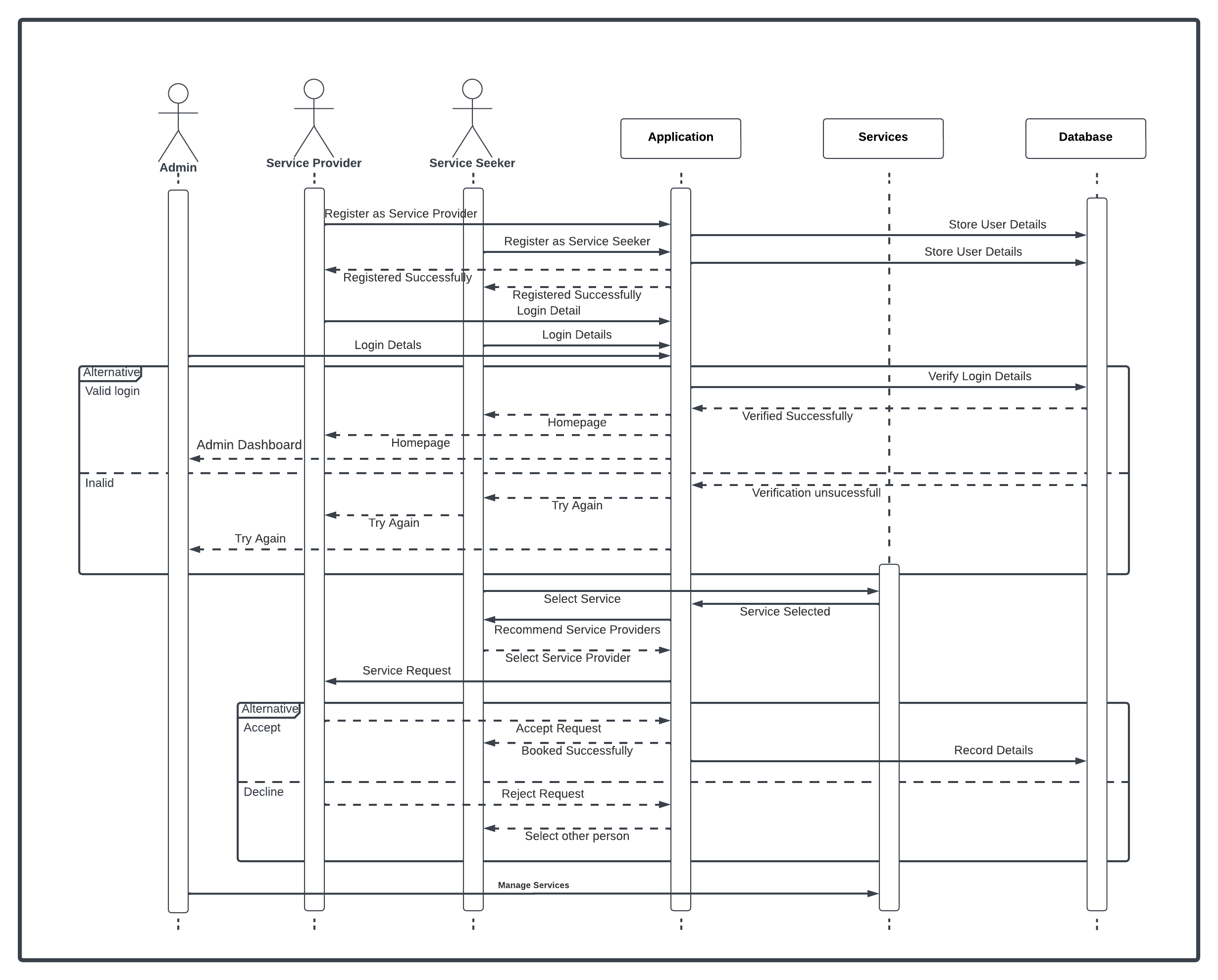


Figure 5: Sequence Diagram of Saajha

# **CHAPTER 4 : SYSTEM DESIGN**

## 4.1 Design

System Design captures the overall design of the system. This system is modeled on MVC architecture (Model, View, Controller). We have chosen this architecture because it will help us in the maintenance of the application and facilitates us with other benefits like easy modification, multiple views, fast development, etc.

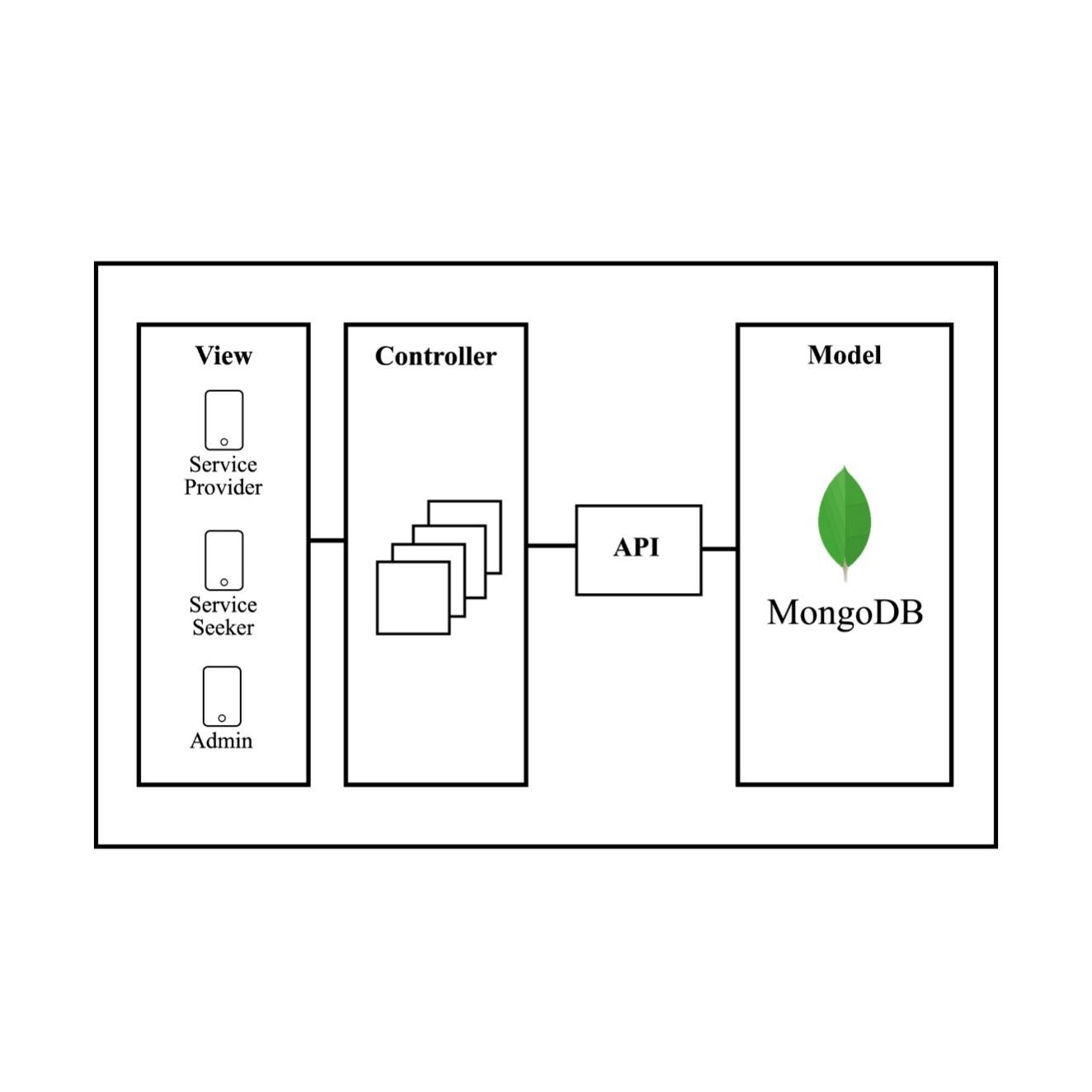


Figure 6: MVC Architecture

In the View, we have two modules, one for the service provider and the other for the service seeker, followed by their controllers, for each view, which will act as a bridge between the view and the model. And in the model, there is MongoDB as a database and Google Apps to obtain the location and trace routes.

The application's login design considers two options, the first option allows entering as a service provider and the second option as a service seeker. In both cases, the user will be able to validate their access using their Gmail. After successful login, both get the privilege of the homepage. The service seeker searches and selects the desired service, and a list of service providers will appear that will show the location of the worker and other attributes like name, contact, price, experience, etc., from which the one is booked. In the same way, the service provider gets notified about the booking made by the service seeker and can accept or reject it.

**Object Oriented Approach**

**Component Diagram**

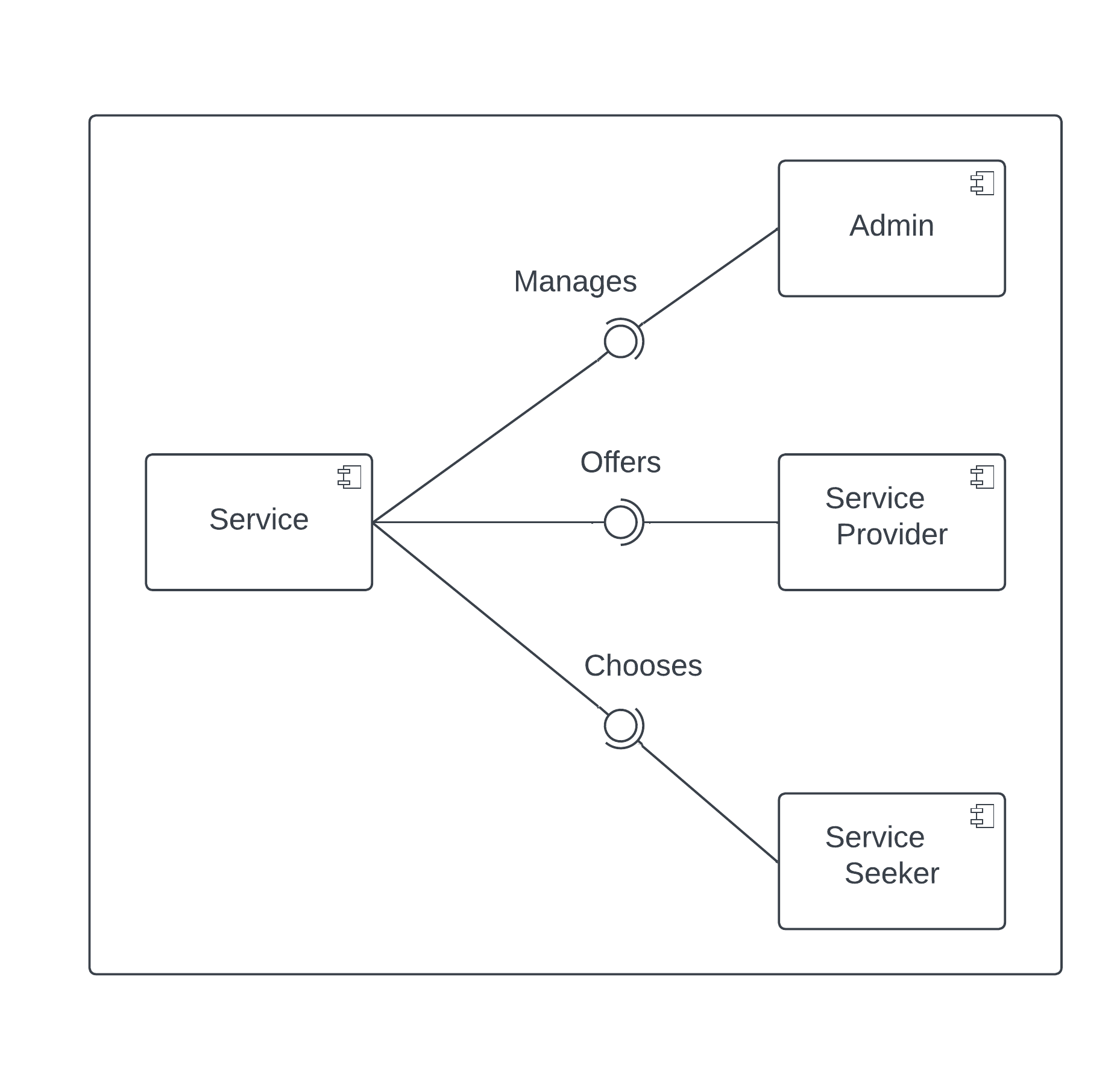
A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

Figure 7: Component Diagram of Saajha

**Deployment Diagram**

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

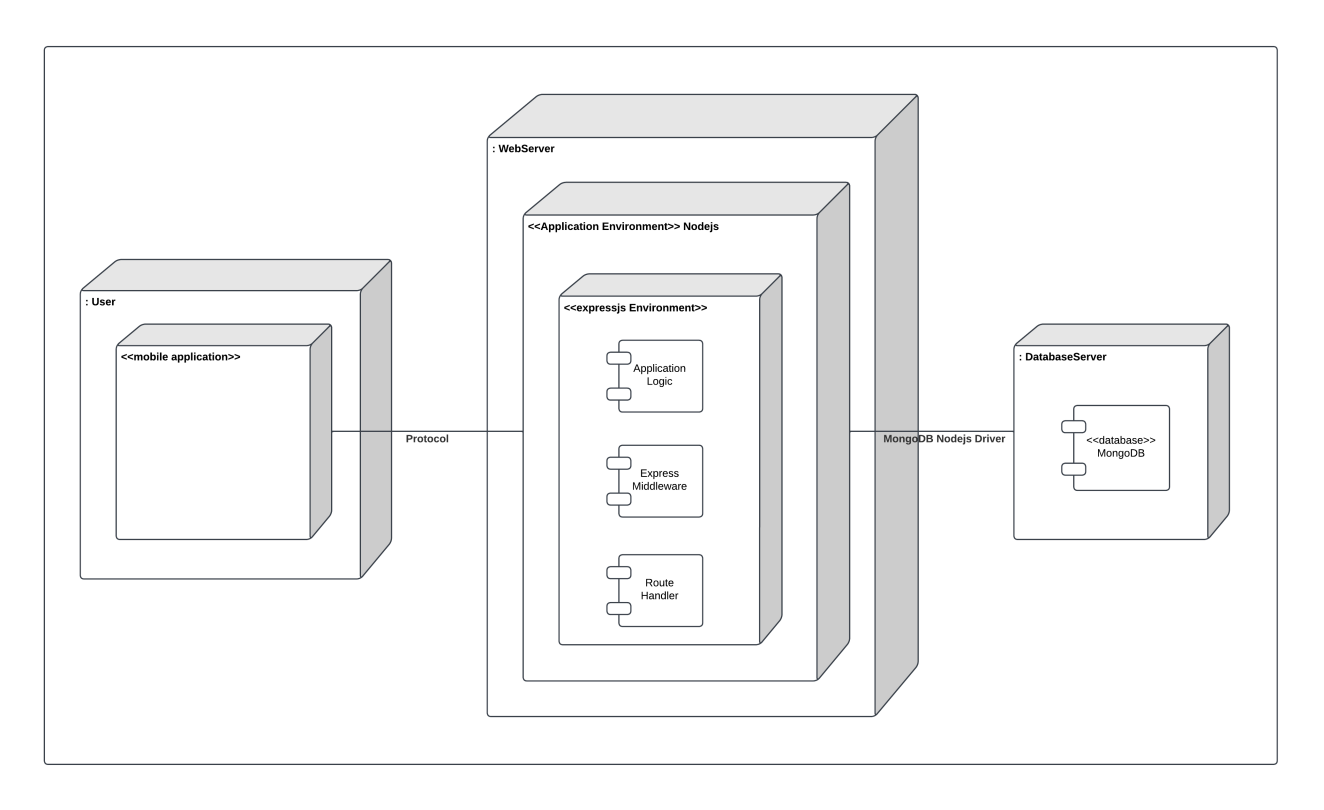
It is typically used to visualize the physical hardware and software of a system. It helps to understand how the system will be physically deployed on the hardware.

Figure 8: Deployment Diagram for Saajha

## 4.2 Algorithm Details

An algorithm is a procedure used for solving a problem or performing a computation. Algorithms act as an exact list of instructions that conduct specified actions step by step in either hardware- or software-based routines.

**Haversine Algorithm**

The Haversine algorithm is the implementation of the Haversine formula. The Haversine formula is used to calculate the distance between two points, with inputs of latitude and longitude as the initial point and end point. To calculate the nearest distance between the service provider and the service seeker, the Haversine formula is used.

The haversine formula is given by:

(1)

Here the values of and in (1) are obtained by using (2) and (3) respectively.

(2)

And,

(3)

Where,

* R = earth’s radius:6371 km
* *∆ lat*  = (*lat2 − lat1*)
* *∆ long* = (*long2 − long1*)
* d = distance (km)

The basic steps to be followed are:

**Step 1:** Start and initialize the radius of the earth as r=6.371

**Step 2:** Detect the coordinates and create an object

**Step 3:** Apply Haversine Formula to calculate the distance.

**Step 4:** Repeat **Step 3** for all other coordinates

**Step 5:** Display in according to the nearest distance

# **CHAPTER 5 : IMPLEMENTATION AND TESTING**

## 5.1 Implementation

The implementation phase of the development life cycle is the phase to materialize the conceptual model into a real physical product. In this phase, the application is actually developed, tested, and integrated into the existing system. It involves the actual coding of the software and the development of any necessary documentation, as well as the testing and debugging of the software to ensure that it meets the requirements and specifications.

### **5.1.1 Tools Used**

#### **i. Visual Studio Code**

VS code is a powerful text editor with built-in support for Dart, Nodejs, and other programming languages. The feature set and rich extensions made it a perfect choice.

#### **ii. Postman**

Postman was used to testing our REST APIs during development. Postman is a powerful tool to test the API with built-in cookies and bearer token support. It is a lightweight easy-to-use tool to test API as compared to the browsers.

#### **iii. Git**

Git is used as a version control system during development.

#### **iv. Github**

GitHub is used to set up a remote repository for remote collaboration among team members.

#### **iv. Lucid Chart**

A lucid chart is used to draw different UML diagrams and flowcharts.

#### **v. Programming languages**

* **Frontend:** Dart
* **Backend:** Nodejs
* **Framework:** Flutter

#### **vi. Database**

MongoDB is used as a database platform which is a NoSQL database.

### **5.1.2 Implementation Details of Modules**

#### **i. Service Seeker**

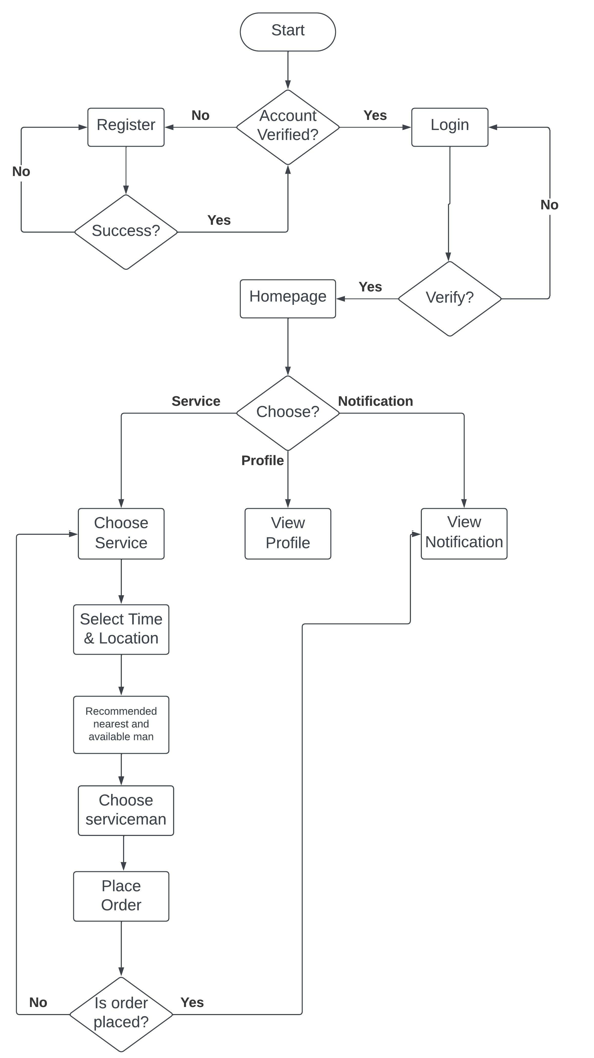


Figure 9: Flowchart of Service Seeker

#### **ii. Service Provider**

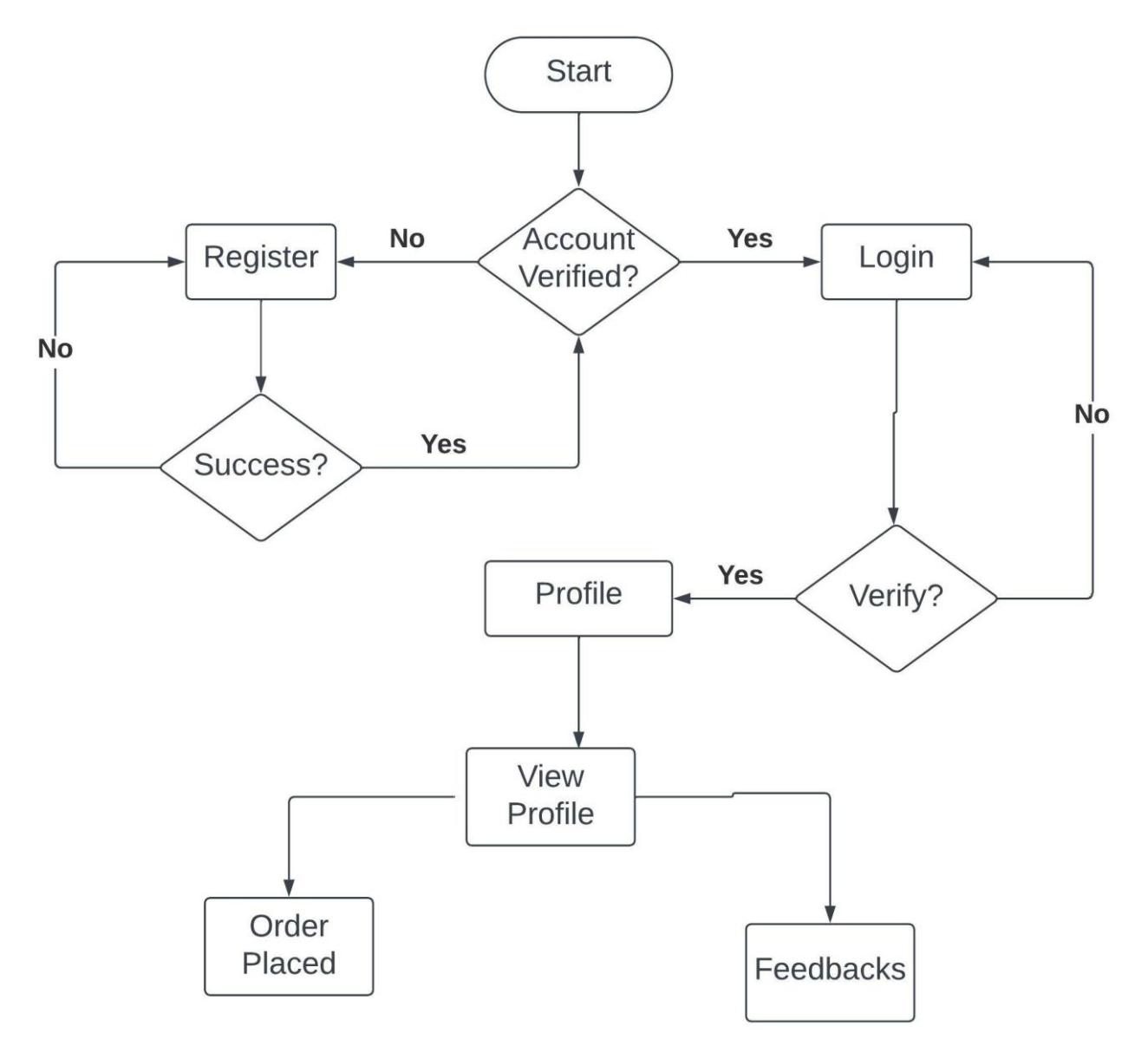


Figure 10:: Flowchart of Service Provider

#### **iii. Admin**

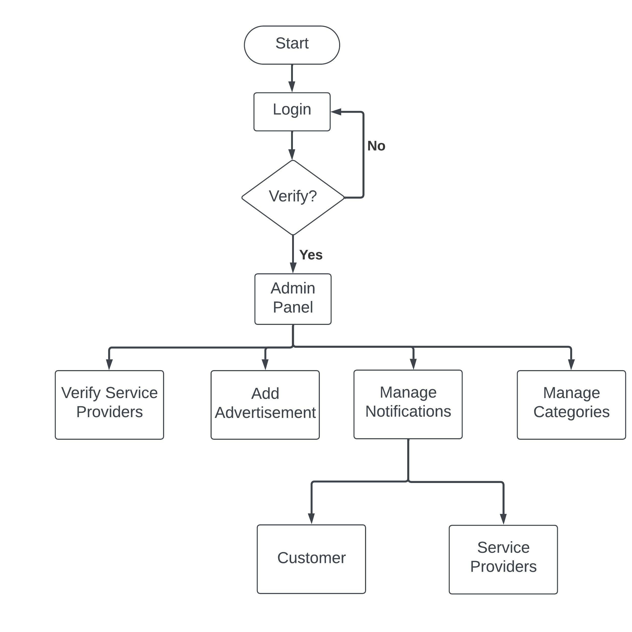


Figure 11: Flowchart of Admin

## 5.2 Testing

Testing strategies are the approach for testing the system or application. It is the process to get some information about the key issues of the developed system and verify the application functions correctly. By running tests against our application consistently, we can verify our application’s correctness, functional behavior, and usability before we release it publicly. Testing is the process of running a system with the intention of finding errors. Testing can be done using varieties of levels. Some of the levels of testing we carried out are:

**5.2.1 Unit Testing**

A unit is the smallest testable part of the software. It usually has one or a few inputs and usually a single output. Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently checked for proper operation. Unit testing increases confidence in changing/maintaining code. We have tested the module in an attempt to discover any error in the code.

Table 2: Registration Testing

|  |  |  |
| --- | --- | --- |
| **Case Name** | Registration | |
| **Purpose of Testing** | Register users as service providers or seekers | |
| **Test Attribute** | Filling out the required information asked during registration. | |
| **Test Unit** | **Test Case** | **Test Result** |
| Register Button | Invalid format or missing values | Display appropriate message |
| Register Button | Valid Format and no missing values | Redirect to the Login Page and store the details in the database |

Table 3: Login Testing

|  |  |  |
| --- | --- | --- |
| **Case Name** | Login | |
| **Purpose of Testing** | Login users | |
| **Test Attribute** | Filling out username and password. | |
| **Test Unit** | **Test Case** | **Test Result** |
| Login Button | Incorrect username and password | Display login unsuccessful message |
| Login Button | Correct username and password | Redirect to the Homepage |

Table 4: View and Edit User’s Details Testing

|  |  |  |
| --- | --- | --- |
| **Case Name** | View and Edit User’s Details | |
| **Purpose of Testing** | Retrieval and Update of the database | |
| **Test Attribute** | Click on the profile and edit button | |
| **Test Unit** | **Test Case** | **Test Result** |
| Profile Display | Click on the profile | Display the detail of the user |
| Edit Button | Change the existing details and Save | Display the save changes message and update the database. |

Table 5: Search and Algorithm Testing

|  |  |  |
| --- | --- | --- |
| **Case Name** | Search and Algorithm | |
| **Purpose of Testing** | Retrieval and Update of the database | |
| **Test Attribute** | Click on the profile and edit button | |
| **Test Unit** | **Test Case** | **Test Result** |
| Search | Type your required service | Display the list of services and allow them to choose one |
| Algorithm | Select the service required | Display the service provider on the basis of the nearest location. |

### 

### **2.2 System Testing**

System Testing is testing on a complete, integrated system to evaluate the system’s compliance with its specific requirements. This testing is done to ensure that the system meets the requirements. System testing involves testing the software code for the following:

* Testing the fully integrated applications in order to check how components interact with one another and with the system as a whole.
* Verifying through testing every input in the application to check for desired outputs.
* Testing of the user’s experience with the application.

Table 6: System Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Strategy** | System Test | | | |
| **Test Module** | Complete application test includes registration, login, profile view, service search, nearest service provider recommendation | | | |
| **S.N.** | **Conditional Test** | **Expected Test** | **Actual Test** | **Remarks** |
|  | Register as a service provider and seeker. | The details provided are stored in the database. | As per the expected result. | Pass |
|  | Log in as a service seeker | Redirect to the homepage of the application | As per the expected result | Pass |
|  | Search for a service or select the featured service | Display a list of service providers on the basis of the nearest distance from the seekers. | As per the expected result | Pass |
|  | Select the service provider and proceed | Send a notification to the service provider | As per the expected result | Pass |
|  | Log in as a service provider | Show the option to accept or reject the request from the service seeker | As per the expected result | Pass |

# 

# **CHAPTER 6** **: CONCLUSION AND RECOMMENDATION**

## 6.1 Conclusion

It has been a matter of immense pleasure, honor, and challenge to have this opportunity to take up this project and complete it successfully. The project is designed considering the problem and traditional method of finding service providers. The project aims to solve the manual method of seeking services by transforming it into a digital system where everything can be done just from your mobile.

The entire project development process teaches us the following skills:

* Understanding database handling.
* Group working/ Team working skills.
* We learned to analyze a problem from the user's point of view and how to make it user-friendly by hiding its complications.
* It was a great help for us to understand and analyze our self-skills and interests that will be helpful for us to choose our field in the future.
* We studied carefully and understood the criteria for making the application more demanding.
* We realized the importance of maintaining a minimal margin for error.

## 6.2 Future Recommendation

Currently, the application is designed for only Android-powered mobile phones. This project can be further developed by enhancing additional features to it.

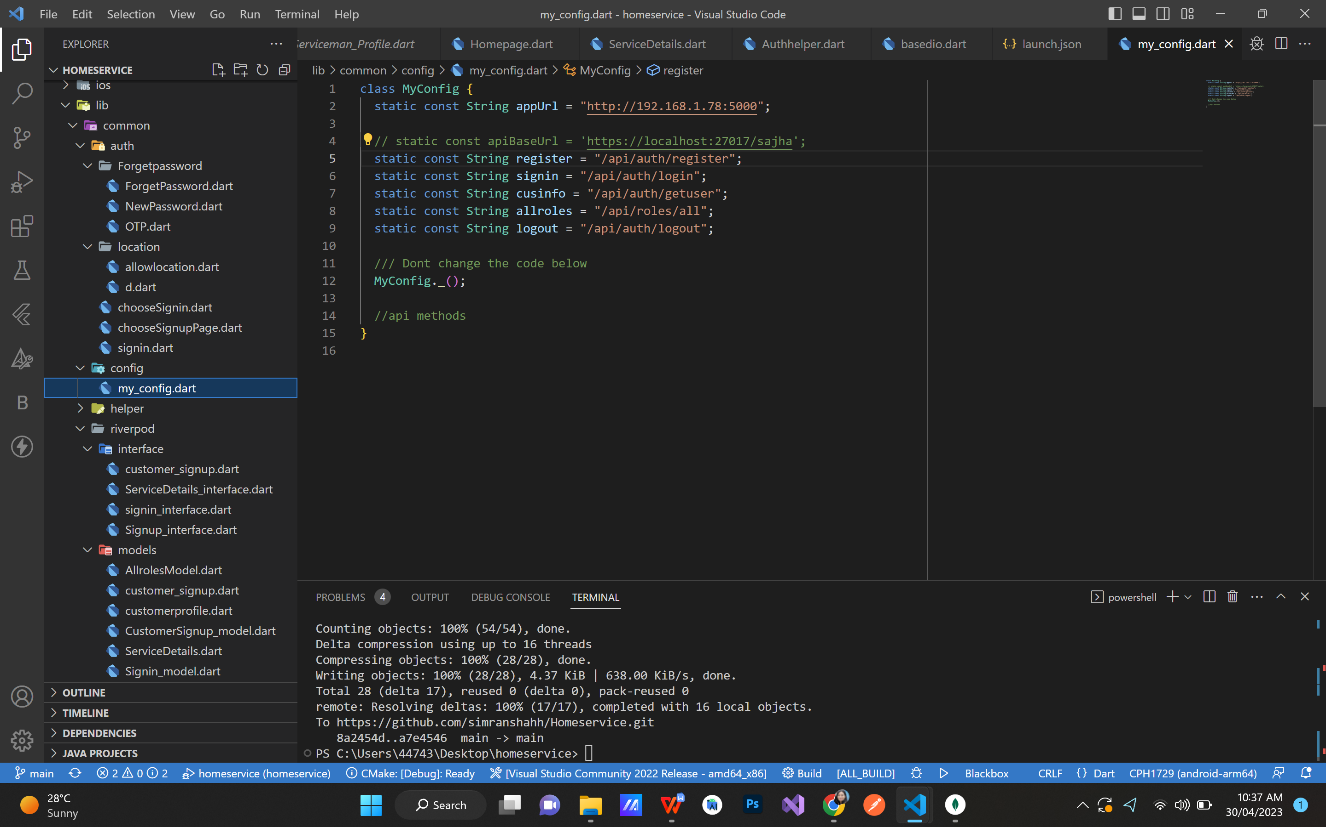
If the following considerations are done, then the project can have a highly commercial scope:

* Its IOS-compatible version can be developed.
* Payment integration.
* Feedback to the service providers can be added.

# **REFERENCE**

|  |  |
| --- | --- |
| [1] | D. L. T. a. A. P. Mancheno, "Incidencia de la Comunicación Visual en clasificados," Universidad de Guayaquil, Equador, 2019. |
| [2] | "About Handy," Angi, 2023. [Online]. Available: https://www.handy.com/about. |
| [3] | HomeAdvisor, "HomeAdvisor," [Online]. Available: https://www.homeadvisor.com/. |
| [4] | S. Sewa, "Sajilo Sewa," [Online]. Available: https://www.sajilosewa.com/. |
| [5] | Bhetayo, "Bhetayo," Imagio Creation, 2023. [Online]. Available: https://www.bhetayo.com/. |
| [6] | D. I. N. Q. J. M. N. S. J. L. H. &. C. J. P.  . Quispe, "Mobile App for the Promotion of Home Services.," IEEE Engineering International Research Conference (EIRCON), 2020. |

# **APPENDICES**



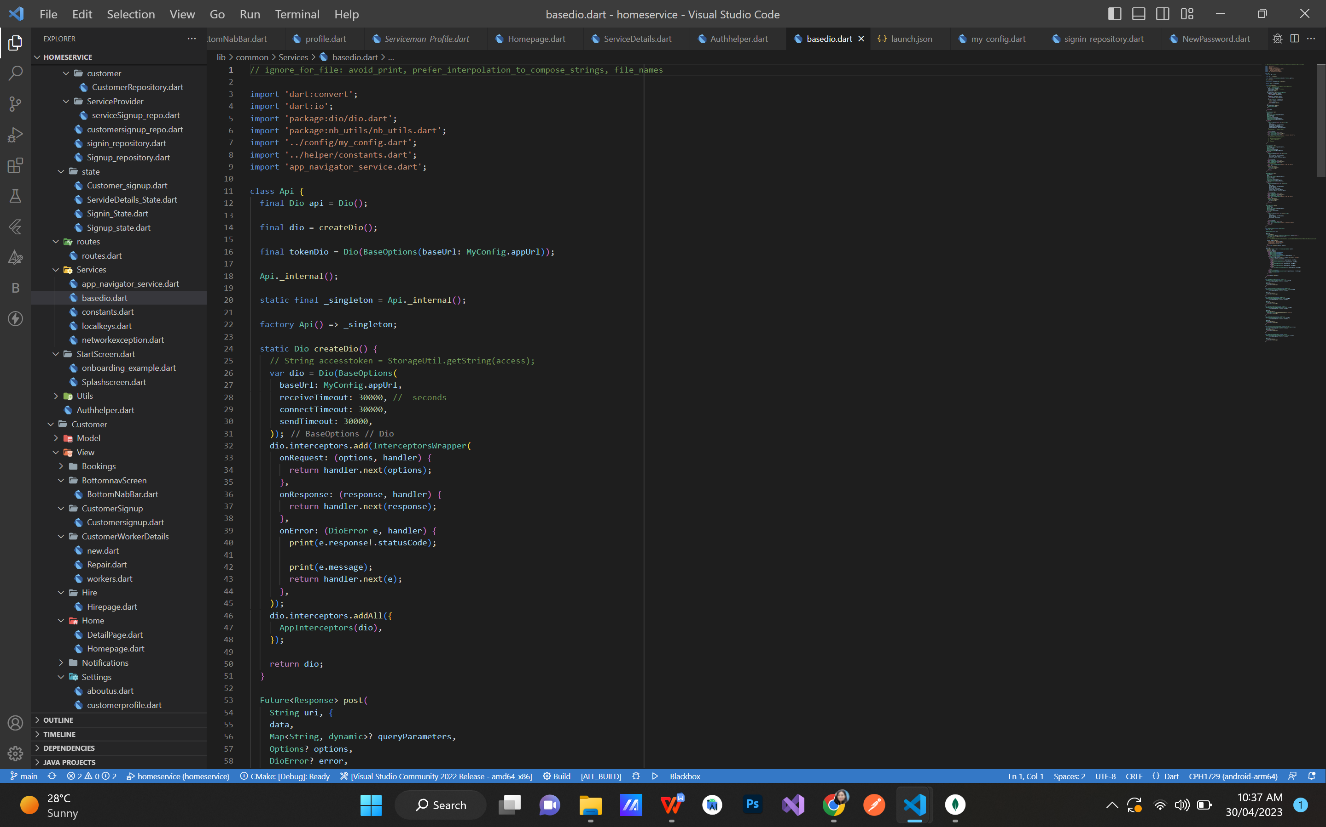
Figure 12: API Routes

Figure 13: DIO

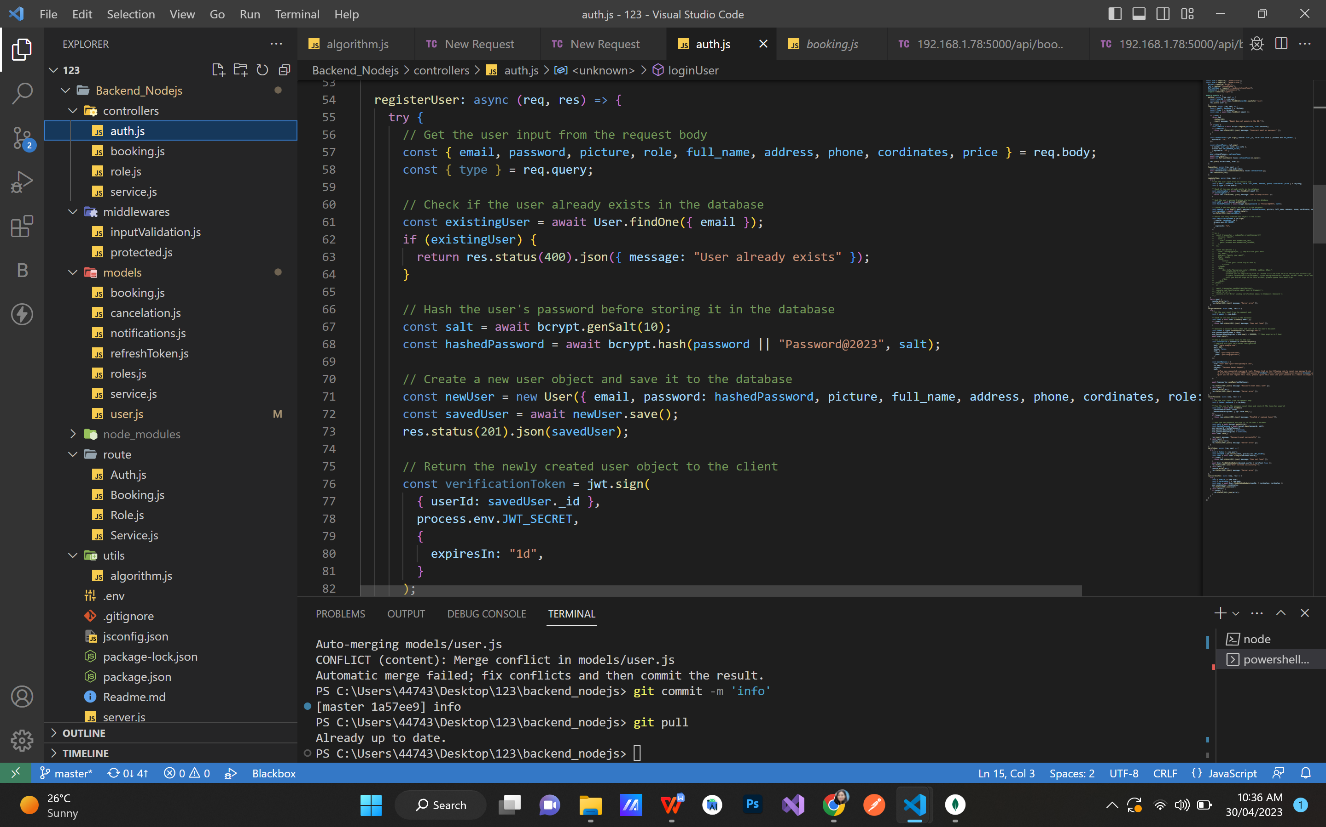


Figure 14: Validation

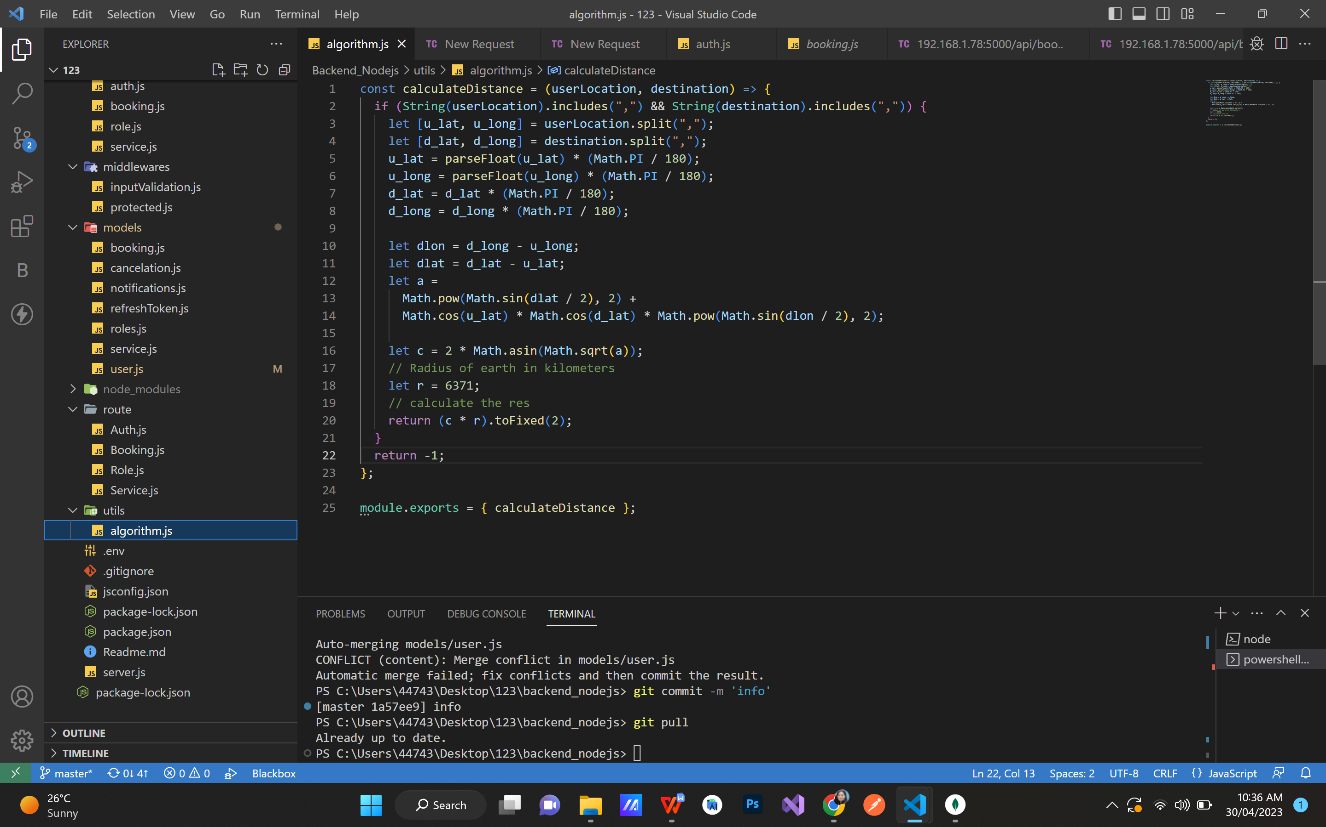


Figure 15: Algorithm Implantation

