



Daffodil
International
University

Project Title: “Ultimate Platform for Efficient Service Management & Scheduling”

Submitted by

A.K.M Nazmul Huda

ID: 163-35-1728

Department of Software Engineering

Daffodil International University

Supervised by

Md. Rajib Mia

Lecturer

Department of Software Engineering

Daffodil International University

This project report has been submitted in fulfilment of the requirement for the
Degree of

Bachelor of Science in Software Engineering

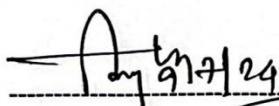
Date of submission: 10th June 2024

APPROVAL

APPROVAL

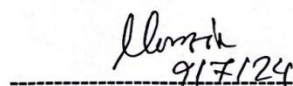
This project titled on “Service Station”, submitted by A.K.M. Nazmul Huda (ID: 163-35-1728) to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



Dr. Engr. AKM Masum
Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Chairman



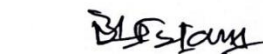
Dr. Marzia Ahmed
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1



Md. Rajib Mia
Lecturer
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2



Dr. Md. Manowarul Islam
Associate Professor
Dept. of Computer Science and Engineering
Jagannath University

External Examiner

DECLARATION

DECLARATION

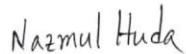
I announce that I am rendering this study document under Md. Rajib Mia, Lecturer, Department of Software Engineering, Daffodil International University. I therefore, state that this work or any portion of it was not proposed here therefore for Bachelors' degree or any graduation.

Supervised By



.....
Md. Rajib Mia
Lecturer
Department of Software Engineering
Daffodil International University

Submitted By



.....
A.K.M Nazmul Huda
ID: 163-35-1728
Department of Software Engineering
Daffodil International University

ACKNOWLEDGEMENT

I am A.K.M Nazmul Huda, I would like to express my sincere appreciation to my supervisor, Rajib Mia, a lecturer in the Department of Software Engineering (SWE). His invaluable guidance and support during the planning and development stages of this system were instrumental. Without his keen insights and strategic advice, reaching the final phase of my work would not have been possible. His role was significant, from providing the initial concept to offering continuous support throughout the project.

I am also deeply grateful to my family and friends for their constant support and encouragement during this journey.

ABSTRACT

This documentation details the design and implementation of a software system intended to simplify finding desired services at affordable costs. The project was developed under the expert guidance of Rajib Mia, a lecturer in the Department of Software Engineering (SWE), whose strategic insights and ongoing support were crucial throughout the development process.

The system is designed to help users easily locate services in any category within nearby areas. Key features include an intuitive search interface, location-based service discovery, and cost comparison tools. The development process encompassed thorough research, careful planning, and iterative testing to ensure the system meets all specified requirements and standards.

This report describes the methodologies used, challenges faced, and solutions adopted during the project. The successful completion of this project was made possible by the steadfast support of my supervisor, family, and friends, whose contributions were invaluable.

TABLE OF CONTENT

APPROVAL	ii
DECLARATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	iv
 Chapter 1	
Introduction	1
 1.1 Project Overview 1	
1.2 Project Purpose 1	
1.2.1 Proposed System 2	
 Chapter 2 System Analysis 3	
 2.1 Feasibility Analysis 3	
2.2 Technical Feasibility..... 3	
2.3 Operational Feasibility..... 3	
2.4 Schedule Feasibility 3	
2.5 Functional Requirements..... 4	
2.6 Non-Functional Requirements..... 5	
2.7 System Requirements..... 6	

Chapter 3 System Design.....	7
3.1 Development Model.....	7
3.1.1 Use Case Diagram.....	8
3.1.2 Use Case Descriptions.....	9-18
3.2 Activity Diagram.....	19-23
3.3 Sequence Diagram.....	24-27
3.4 ERD Diagram.....	28
3.5 Block Diagram	29
 Chapter 4 Development Tool and Technology.....	 30
4.1 Integrated Development Environment (IDE).....	30
4.2 Programming Language.....	30
4.3 User Interface Design.....	30
4.4 Database.....	30
4.5 Deployment and Hosting.....	30
 Chapter 5 System Testing.....	 31-36

5.1 Testing Features.....	31
5.2 System Testing Plan.....	31
5.2.1 Test Approach.....	36
Chapter 6 User Manual.....	37-45
6.1 Registration Page.....	37
6.2 Login Page.....	38
6.3 Dashboard Page.....	39
6.4 Service List Page.....	40
6.5 Create Service Page.....	41
6.6 Service Page.....	42
6.7 Single Service and purchase Services page.....	43
6.8 Order page for client.....	44
6.9 Order page for service provider.....	44
6.10 Confirm order by service provider.....	45
Chapter 7	
Conclusion.....	46

7.1 Project Link.....	46
7.2 Limitations.....	46
7.3 Future Scope.....	46
Reference.....	46

CHAPTER 1: INTRODUCTION

1.1 Project Overview

The Service Finder system makes it easy to find, schedule, and purchase services. It allows users to communicate more effectively with providers, facilitating communication. Users can purchase services, make appointments, and ask specific questions. Providers build inventories, process requests, and respond quickly. This platform aims to simplify the process of finding and using services. It improves the user experience with simple features and clear terminology. It's a helpful tool for finding services and support providers offering great services.

1.2 Project Purpose

- **Expand Service Reach:** The directory service will help to expand the reach of any services. Users can search multiple of nearby services from a single platform only through this system. This effort aims to remove the facilitators of stumbling blocks around service discovery so that users can more readily find and use different services
- **Service Exploration Simplification:** The project is related to simplifying the task of identification services by providing easy-going search options and listing in a manner that defines/reset the whole notion. In addition, it enables users to access their overall options easily due to the user-friendly interface.
- **Seamless management of appointments:** the system is programmed to allow easy scheduling and confirmation of service appointments. With strong scheduling functionalities in place, both the users can easily deal with their appointments allowing service providers to compile a book of business

1.2.1 Proposed System

The goal is to provide a platform such as Service Finder in which it will ease the work related to finding, scheduling, and purchasing services with added functionality for a better user experience the proposed system seeks to meet these needs using a variety of components:

Allow the users to limit and filter their search results according to location, price range, and service category;

It is user-friendly as far as searching and browsing the database to ensure that you can find the services that are intended for your needs/preferences.

User-friendly design — The UI/UX will have advanced design features for easy-to-use and pleasant functionality.

Chapter 2: SYSTEM ANALYSIS

2.1 Feasibility Analysis

As a result, further research should be conducted to determine the viability of the Research Point app through a comprehensive feasibility analysis. This evaluation should take several forms, from whether the application is technically feasible, to how one will likely manage running a service at full scale

2.2 Technical Feasibility

We will invest in robust servers, software, and user-friendly technology so that we are always prepared to scale up or down securely (using Secure Sockets Layer infrastructure) to be compatible with EC2CHANGE's technology.

2.3 Operational Feasibility

Developed user interfaces and well-defined options Case should have proper resource management solutions.

2.4 Schedule Feasibility

To implement these functionalities without suffering the burdens of incredibly high demand in terms of API usage, realistic timelines, and diligent resource management will be our priority.

2.5 Functional Requirements

1. User and Provider will be requested for pass with fill-up email ID.
2. User can register with their valid email and password.
3. The system will not permit users to log in using a password or email that is invalid.
4. Users and providers both will be able to add user details and update them.
5. User can search for their desired service.
6. If a user cannot find their desired service, they can send a service request.
7. User and Provider can view and check the service details
8. Users can select the package if the provider is available on this date and confirm the order.
9. If a user finds a provider on his schedule, he can confirm the service package.
10. The provider can also reject an order if it is not possible.
11. Provider can create new services and besides this, he can update and delete services.
12. After confirming package payment should be done.

2.6 Non-Functional Requirements

1. Using token-based authentication, session, Validation, 2FA it will be secure from unauthorized access.
2. The system should work 24/7 a user can get access and service.
3. Data or process requirement concerned with defining the precision which the solution will record or produce data.
4. It's a way how easy to support, change and enhance the system.

2.7 System Requirements

Hardware Requirements:

- **Servers:**
 - Multi-core processors with high processing power.
 - Sufficient RAM for efficient data processing and storage.
 - Ample storage capacity for research papers and system data.
- **End-User Devices:**
 - Smartphone's running iOS 7 / Android 8 or later.
 - Windows 7 or later (Browsers - Latest versions of Chrome, Firefox, Edge, or Safari).
 - MacOS 10.12 (Sierra) or later (Browsers - Latest versions of Safari, Chrome, or Firefox).
 - Most modern Linux distributions released in the past 5 years (Browsers - Latest versions of Chrome or Firefox).

Database Requirements:

- **Research Paper Metadata:**
 - Fields: Author names, titles, keywords, abstracts.
 - Data: Document URLs, download statistics.
- **User Profiles:**
 - Fields: Usernames, email addresses, profile details.
- **Security Measures:**
 - Data Protection: Encrypted storage of sensitive user data.
 - Access Control: Role-based permissions.

Security Requirements:

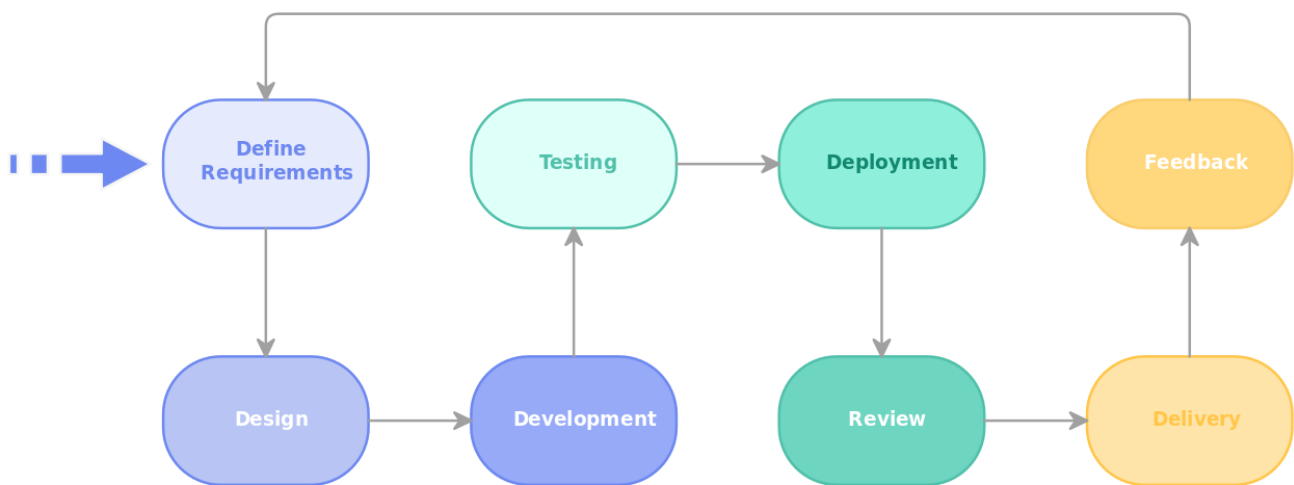
- **User Authentication:**
 - Method: Secure login with password encryption.
 - Technique: Hash algorithm for security.

Chapter 3: System Design

3.1 Development Model:

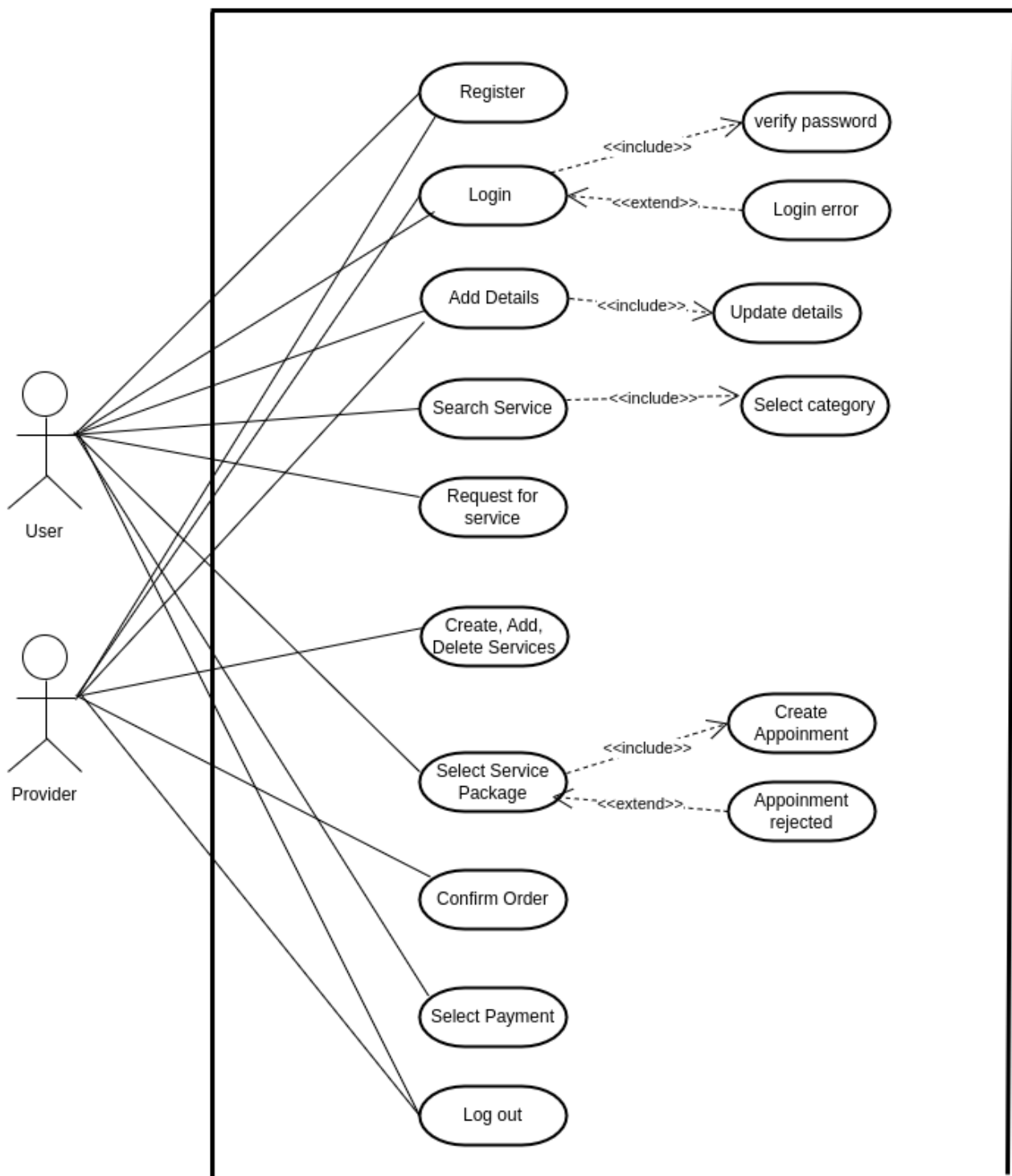
The Agile SDLC model is being followed in the development of the "Service Station" application. Since the project's needs are expected to evolve and adapt over time. The Agile model has been chosen because of its:

1. Flexibility and iterative development
2. Enhanced collaboration and communication
3. Rapid delivery and continuous improvement
4. Fit for dynamic, high-risk initiatives



- Agile Software Development Process -

3.1.1 Use Case Diagram



3.1.2 Use Case Descriptions

1. Register

Use Case Name	Register
Actor	User, Provider
Pre-Condition	None
Flow and Counts	1. The user or provider selects the "Register" option. 2. The user or provider enters the required information. 3. The system creates the account.
Alternative Flow	If the password verification fails, the user or provider is prompted to re-enter the password.

2. Login

Use Case Name	Login
Actor	User, Provider
Pre-Condition	1. The user or provider has an existing account.
Flow and Counts	1. The user or provider selects the "Login" option. 2. The user or provider enters their login credentials. 3. The system authenticates the credentials. 4. The user or provider is logged in.
Alternative Flow	If the login fails, the system displays a "Login Error" message.

3. Add Details

Use Case Name	Add Details
Actor	User, Provider
Pre-Condition	1. The user or provider is logged into the system.
Flow and Counts	1. The user or provider selects the "Add Details" option. 2. The user or provider enters additional information. 3. The system updates the profile details.
Alternative Flow	If the update fails, the system displays an error message.

4. Search Service

Use Case Name	Search Service
Actor	User
Pre-Condition	None
Flow and Counts	1. The user selects the "Search Service" option. 2. The user selects a category. 3. The system displays a list of services in the selected category.
Alternative Flow	If no services are found, the system displays a "No services found" message.

5. Request for Service

Use Case Name	Request for Service
Actor	User
Pre-Condition	1. The user is logged into the system.
Flow and Counts	1. The user selects a service. 2. The user sends a request for the selected service. 3. The system sends the request to the provider.
Alternative Flow	If the request fails, the system displays an error message.

6. Create, Add, Delete Services

Use Case Name	Create, Add, Delete Services
Actor	Provider
Pre-Condition	1. The provider is logged into the system.
Flow and Counts	1. The provider selects the "Create, Add, Delete Services" option. 2. The provider enters or deletes service details. 3. The system updates the list of services.
Alternative Flow	If the update fails, the system displays an error message.

7. Select Service Package

Use Case Name	Select Service Package
Actor	User
Pre-Condition	1. The user is logged into the system. 2. The user has searched for services.
Flow and Counts	1. The user selects a service package. 2. The system creates an appointment. 3. The system confirms the appointment.
Alternative Flow	If the appointment is rejected, the system displays an "Appointment Rejected" message.

8. Confirm Order

Use Case Name	Confirm Order
Actor	User
Pre-Condition	1. The user has selected a service package.
Flow and Counts	1. The user selects the "Confirm Order" option. 2. The system confirms the order.
Alternative Flow	If the order confirmation fails, the system displays an error message.

9. Select Payment

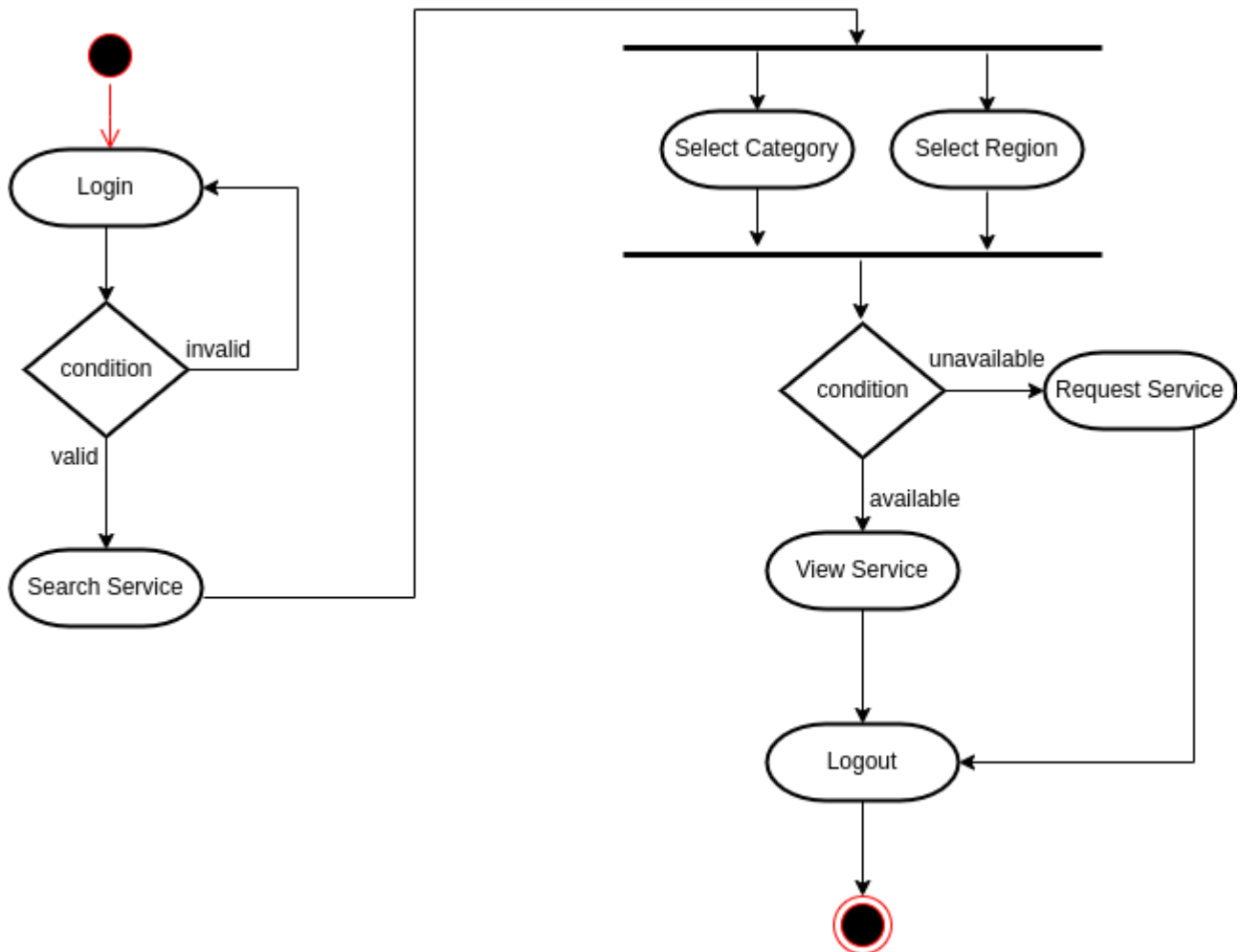
Use Case Name	Select Payment
Actor	User
Pre-Condition	1. The user has confirmed their order.
Flow and Counts	1. The user selects a payment method. 2. The system processes the payment.
Alternative Flow	If the payment fails, the system displays an error message.

10. Log out

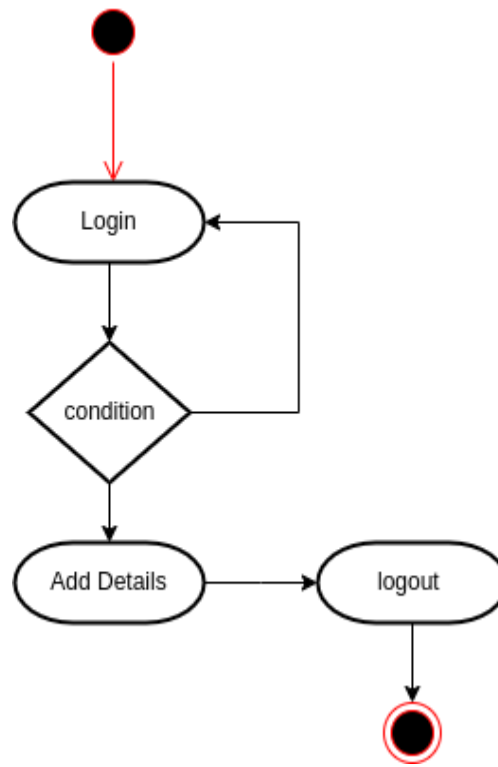
Use Case Name	Log out
Actor	User, Provider
Pre-Condition	1. The user or provider is logged into the system.
Flow and Counts	1. The user or provider selects the "Log out" option. 2. The system logs the user or provider out.
Alternative Flow	If the log out fails, the system displays an error message.

3.2 Activity Diagram

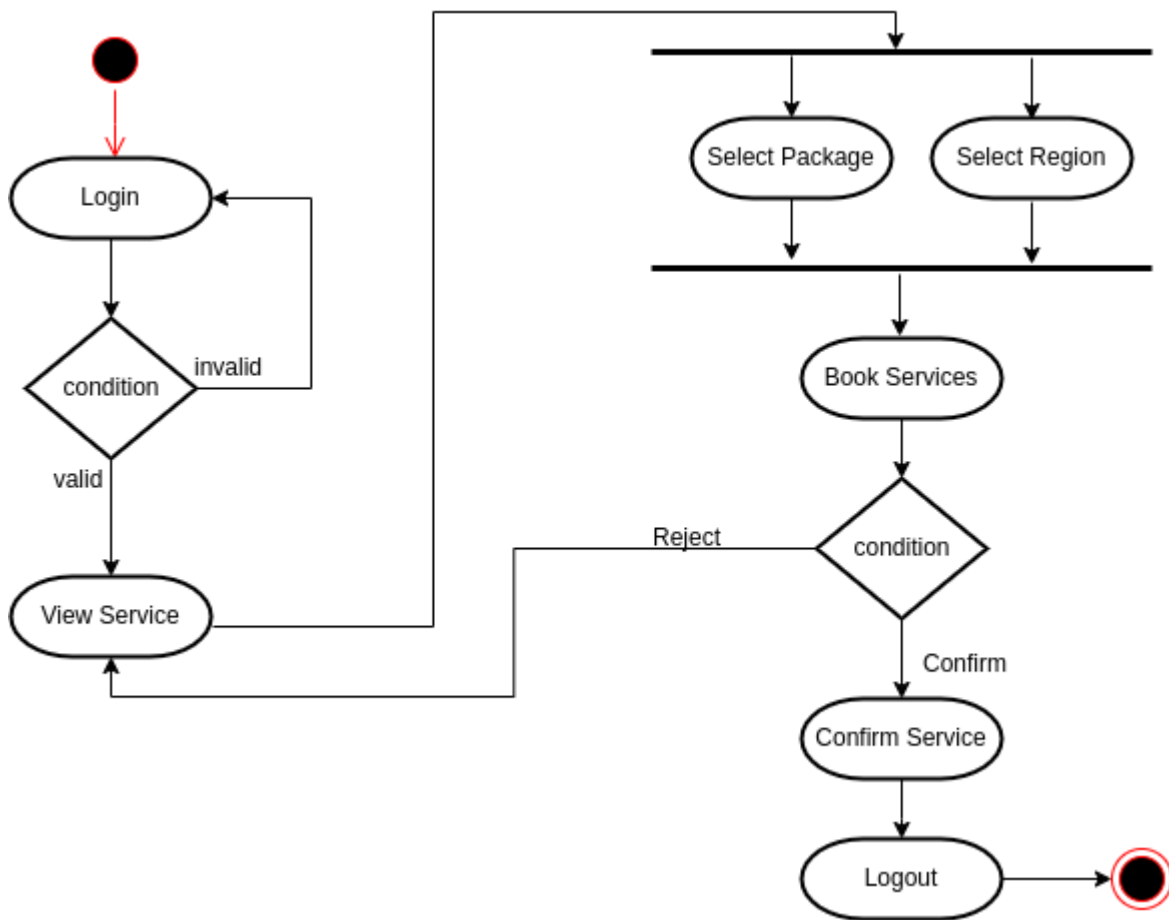
View Service



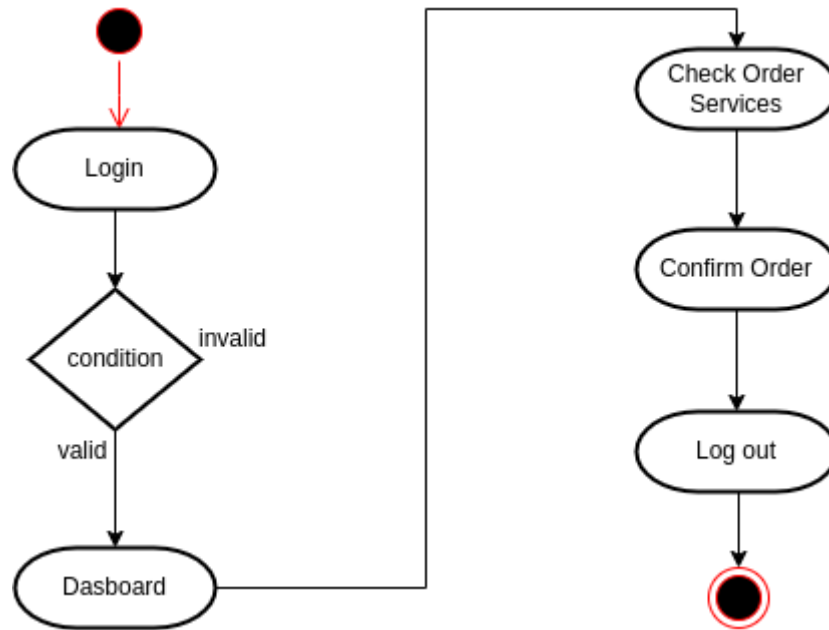
Add details (user/provider)



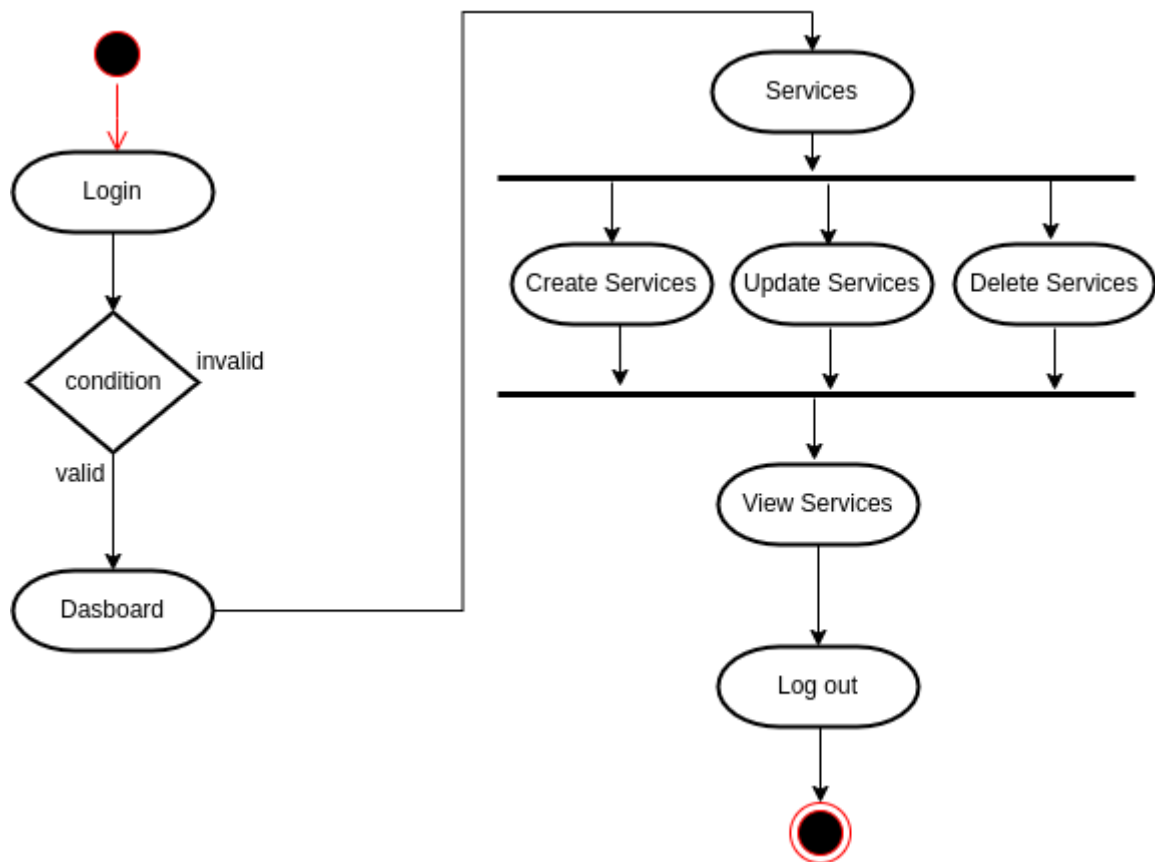
Confirm Service



Confirm Order

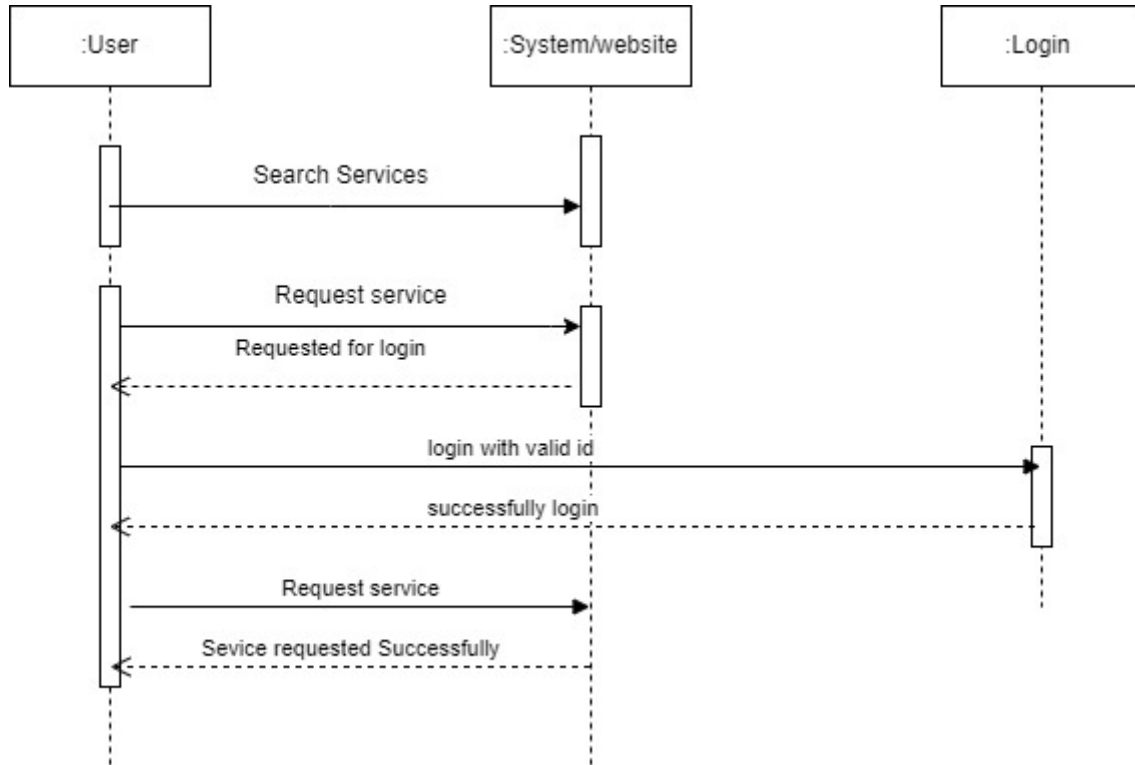


Create, update and delete Service

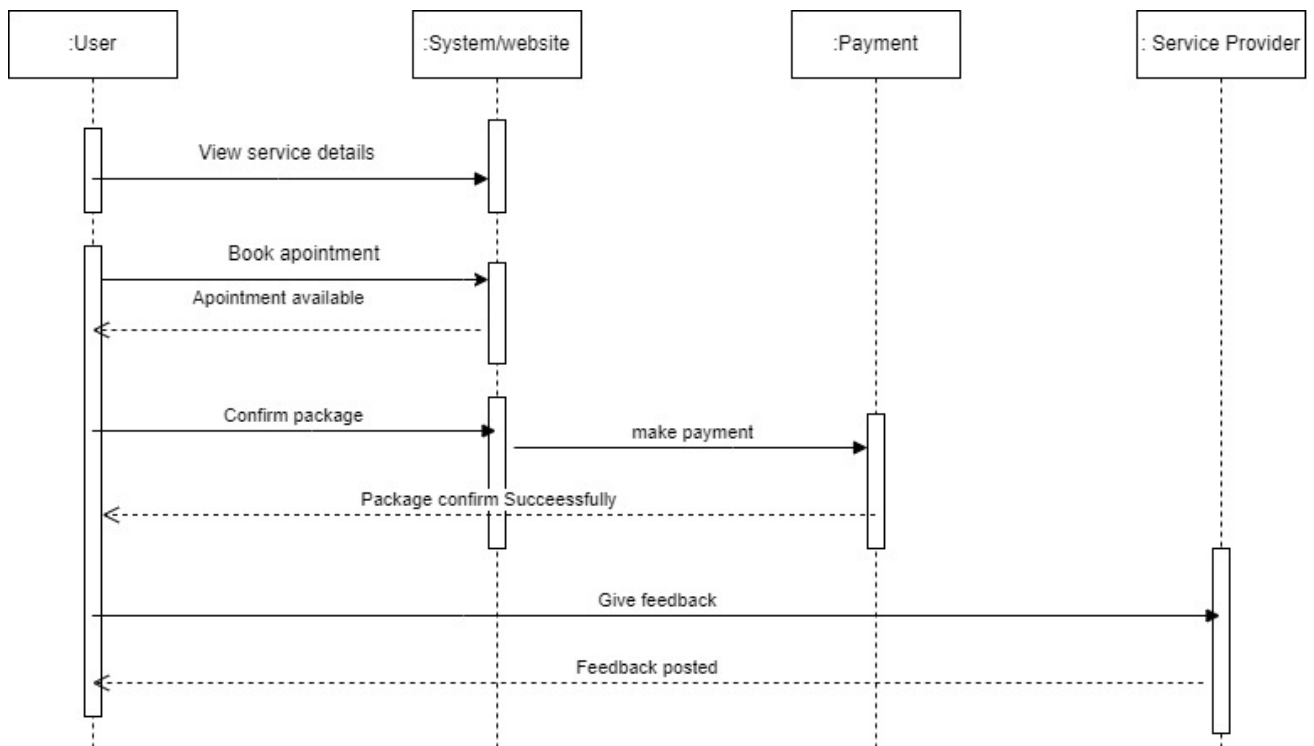


3.3 Sequence Diagram

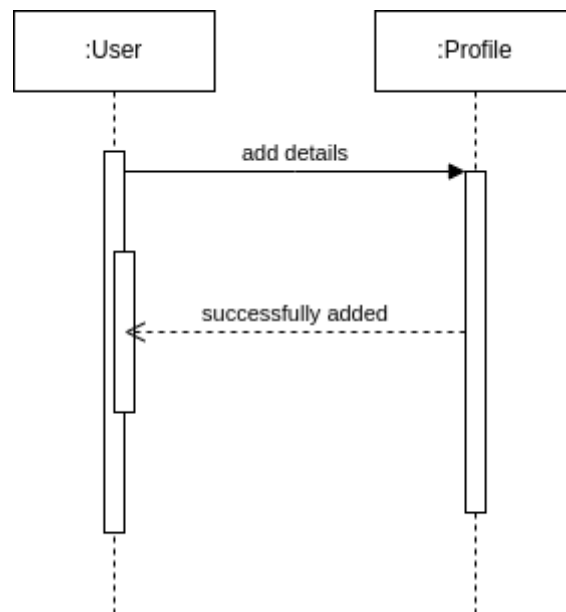
Sequence Diagram (User Request Service)



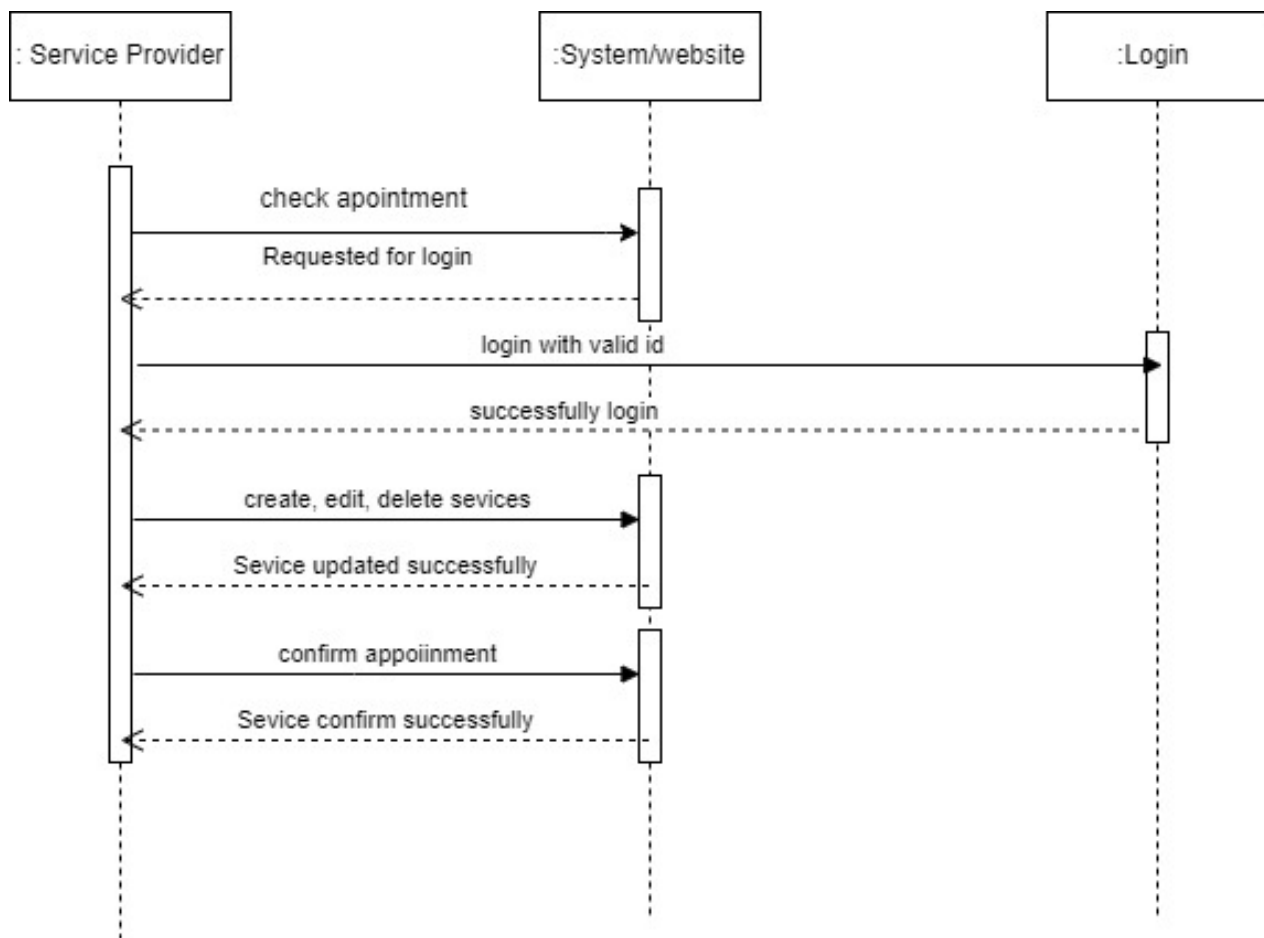
Sequence Diagram (User Book Appointment)



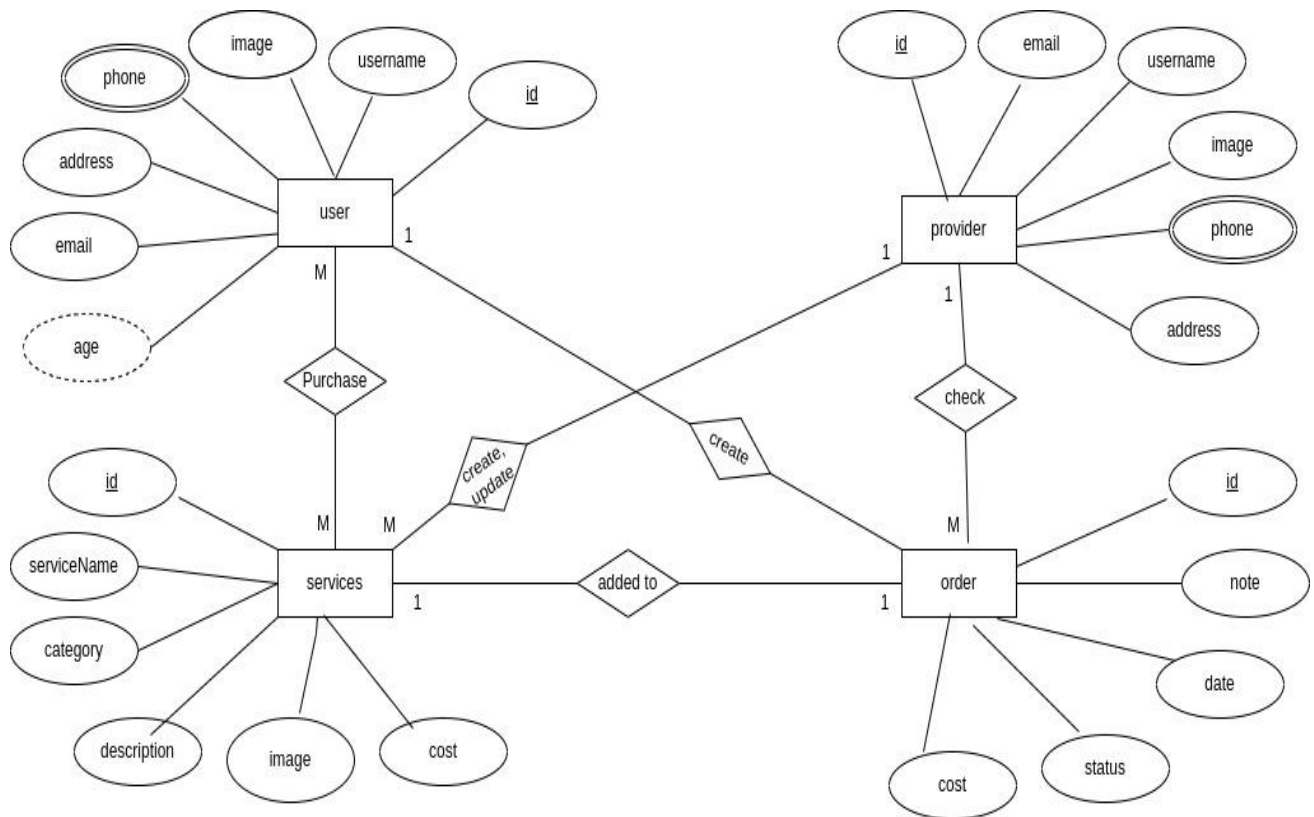
Sequence Diagram (Add details)



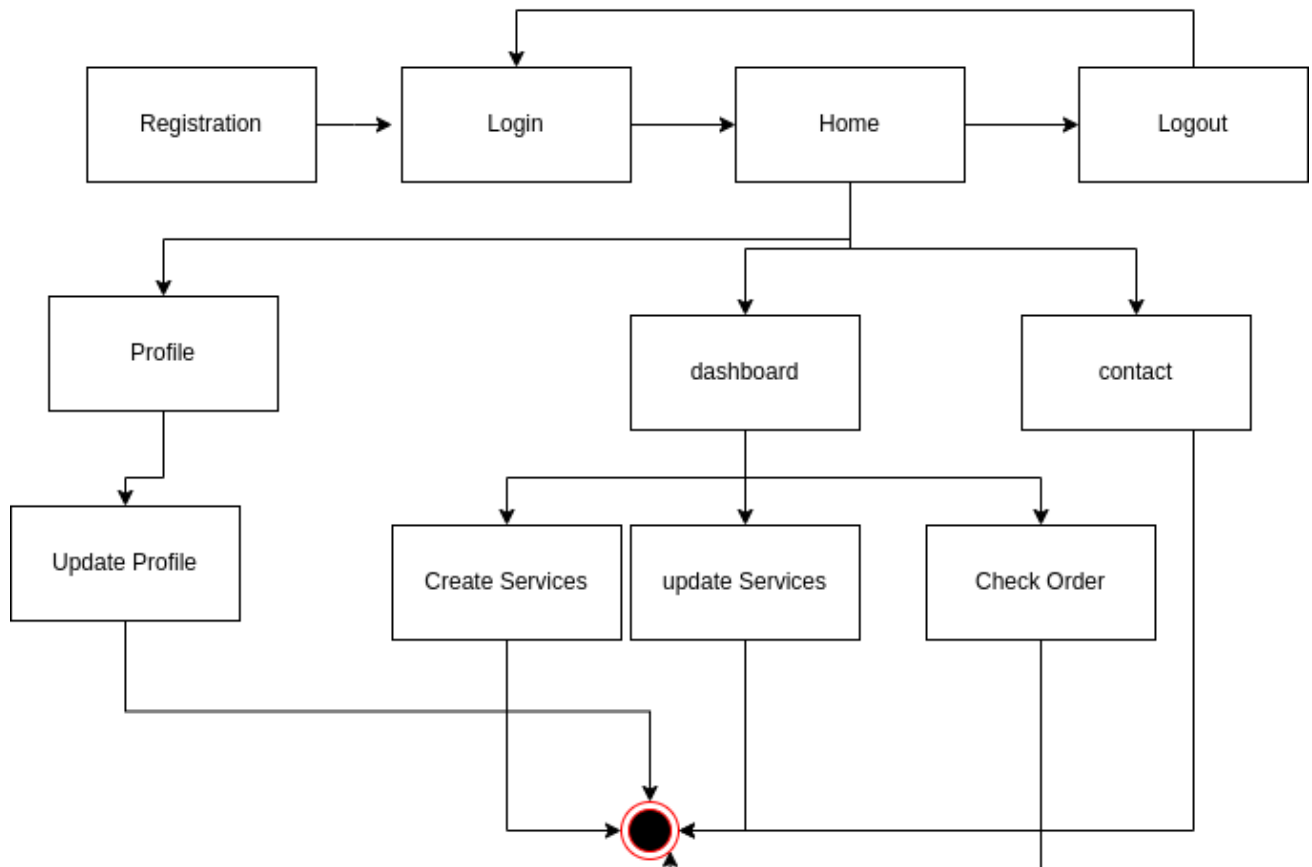
Sequence Diagram (Service provider)



3.4 ER Diagram



3.5 Block Diagram



Chapter 4: Development Tool and Technology

4.1 Integrated Development Environment (IDE)	Visual Studio Code
4.2 Programming Language	JavaScript
4.3 User interface Design	Figma
4.4 Database	MongoDB
4.5 Deployment and hosting	Versel

Chapter 5: System Testing

5.1 Testing Features

System testing is all about making sure that the whole software system, once integrated, meets the specified requirements. Here's a breakdown of how structure this system testing based on the use cases mentioned above:

5.2 System Testing Plan

1. Register

Objective: To verify that users and providers can register successfully.

Test Steps:

- Go to the registration page.
- Enter valid user or provider details.
- Submit the registration form.
- Make sure the system verifies the password.
- Confirm that the account is created successfully.

Expected Results:

- The registration form should accept valid details.
- The password verification should work.
- The account should be created, and a confirmation message should be displayed.

2. Login

Objective: To make sure that users and providers can log in with their valid credentials.

Test Steps:

- Go to the login page.

- Enter your valid credentials.
- Submit the login form.
- Check that the system logs you in.
- Test with invalid credentials to see if the "Login Error" message appears.

Expected Results:

- You should be able to log in successfully with valid credentials.
- The system should display a "Login Error" message for invalid credentials.

3. Add Details

Objective: To check if users and providers can add or update their details.

Test Steps:

- Log in to the system.
- Go to the "Add Details" page.
- Enter any additional details.
- Submit the form.
- Verify that the details are updated.

Expected Results:

- The details should be added or updated successfully.
- The system should show a confirmation message.

4. Search Service

Objective: To ensure that users can search for services.

Test Steps:

- Log in to the system.
- Go to the "Search Service" page.
- Select a category.
- Verify that the system displays relevant services.

Expected Results:

- The system should show services related to the selected category.
- If no services are available, the system should display a "No services found" message.

5. Request for Service

Objective: To confirm that users can request services.

Test Steps:

- Log in to the system.
- Search for a service.
- Select a service and send a request.
- Verify that the request is sent to the provider.

Expected Results:

- The service request should be sent successfully.
- The system should display a confirmation message.

6. Create, Add, Delete Services

Objective: To make sure that providers can manage their services.

Test Steps:

- Log in as a provider.
- Go to the "Create, Add, Delete Services" page.

- Add a new service.
- Verify that the service is added.
- Delete an existing service.
- Verify that the service is deleted.

Expected Results:

- The services should be added and deleted successfully.
- The system should display appropriate confirmation messages.

7. Select Service Package

Objective: To verify that users can select a service package.

Test Steps:

- Log in to the system.
- Search for and select a service package.
- Verify that an appointment is created.
- Check how the system handles appointment rejection.

Expected Results:

- The service package selection should be successful.
- An appointment should be created.
- If applicable, the system should display an "Appointment Rejected" message.

8. Confirm Order

Objective: To confirm that users can finalize their orders.

Test Steps:

- Log in to the system.

- Select a service package.
- Confirm the order.
- Verify that the order is confirmed.

Expected Results:

- Your order is confirmed successfully.
- The system will display a confirmation message.

9. Choose Your Payment Method

Objective: Ensure users can select a payment method.

Test Steps:

- Log in to the system.
- Confirm your order.
- Select your preferred payment method.
- Verify that the payment is processed.

Expected Results:

- The payment is processed successfully.
- The system will display a confirmation message.

10. Log out

Objective: Verify that users and providers can log out.

Test Steps:

- Log in to the system.
- Select the "Log out" option.
- Verify that the system successfully logs out the user or provider.

Expected Results:

- The user or provider is logged out successfully.
- The system will redirect to the login page.

5.2.1 Test Approach:

The test approach ensures that the Service Management System meets all requirements and functions correctly for users and providers. The strategy includes multiple testing levels:

- **Unit Testing:** This phase focuses on verifying individual components' correctness. It involves creating test cases for each unit (e.g., functions, methods, or classes). The scope includes core functionalities such as registration, login, service search, and payment processing.
- **Performance Testing:** The objective here is to validate the system's performance under various conditions. Key performance metrics such as response time, throughput, and resource utilization are identified. Different levels of user load are simulated, and system performance is measured, analysed, and optimized.

Chapter 6: User Manual

6.1 Registration

Registration

Username

Email

Password

Register

6.2 Login Page

Login

Email

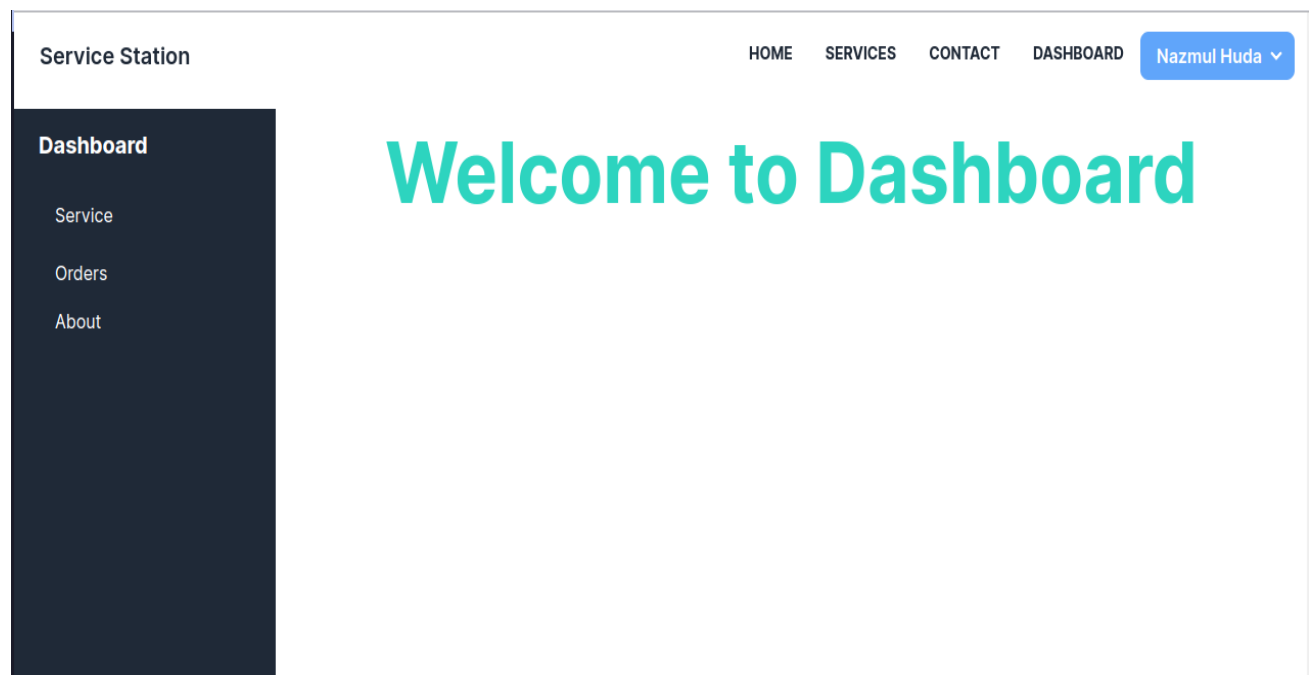
Password

Login

OR

Register

6.3 Dashboard



6.4 Service list page

Service Station

HOME SERVICES CONTACT DASHBOARD Nazmul Huda ▾

Dashboard

Service

Orders

About

Create Service

Service Name	Service Category	Service Cost	Service Area	Delete/Edit
Electricity	Electricity	2000	Dhaka	 
BroadBand	Internet	1000	Dhaka	 

6.5 Create Service page

Service Station

HOME SERVICES CONTACT DASHBOARD Nazmul Huda ▾

Dashboard

Service

Orders

About

Create Service

Service Name :

Service Category :

Discriptions :

Service Cost :

Select image

Choose File

No file chosen

Create Service

6.6 Service page

Service Station

Search Services

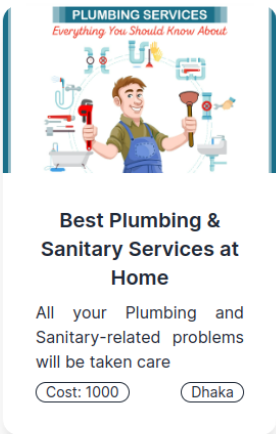
Filtered Services



ABC Technology

ABC IT has come a long way since its establishment

Cost: 500 Dhaka



Best Plumbing & Sanitary Services at Home

All your Plumbing and Sanitary-related problems will be taken care

Cost: 1000 Dhaka

SERVICES

Branding

Design

Marketing

Advertisement

COMPANY

About us

Contact

Jobs

Press kit

LEGAL

Terms of use

Privacy policy

Cookie policy

6.7 Single Service and purchase Services page

Service Station

HOME SERVICES CONTACT DASHBOARD Nazmul Huda ▾



ABC Technology

ABC IT has come a long way since its establishment in 1997. From small beginnings as a provider of dial-up & radio link Internet access to local businesses, we have grown consistently and organically as a communications provider serving a diverse portfolio of business class voice and data services.

Service Cost: 500 TK

Service Category: Internet

Service Area: Dhaka

PURCHASE SERVICE

SERVICES

COMPANY

LEGAL

Branding

About us

Terms of use

Design

Contact

Privacy policy

Marketing

Jobs

Cookie policy

Advertisement

Press kit

6.8 Order page for client

Service Station

HOME SERVICES CONTACT DASHBOARD Nazmul Huda ▾

Order Id	Service Name	Service Category	Service Cost	Appointment Date	Order Status	Delete
6665e561a64d67c40d0a40e3	Best Plumbing & Sanitary Services at Home	Plumbing	1000	2024-05-14T00:00:00.000Z	Cancelled	
6665e571a64d67c40d0a40f5	ABC Network	Internet	500	2024-06-14T00:00:00.000Z	Processing	
6665e8caa64d67c40d0a41a2	ABC Network	Internet	500	2024-05-14T00:00:00.000Z	Confirmed	

SERVICES

Branding
Design
Marketing

COMPANY

About us
Contact
Jobs

LEGAL

Terms of use
Privacy policy
Cookie policy

6.9 Order list for service provider

Service Station

HOME SERVICES CONTACT DASHBOARD Nazmul Huda ▾

Dashboard

Service
Order
About

Client Name	Service Name	Service Category	Service Cost	Appointment Date	Order Status	Delete
Nazmul Huda	Best Plumbing & Sanitary Services at Home	Plumbing	1000	2024-05-14T00:00:00.000Z	Cancelled	 
Nazmul Huda	ABC Network	Internet	500	2024-06-14T00:00:00.000Z	Processing	 
Nazmul Huda	ABC Network	Internet	500	2024-05-14T00:00:00.000Z	Confirmed	 

6.10 Confirm order by service provider

Service Station

HOME SERVICES CONTACT DASHBOARD Nazmul Huda

Dashboard

Service

Order

About

PLUMBING SERVICES

Everything You Should Expect From Us



Best Plumbing & Sanitary Services at Home

Client Name: Nazmul Huda

Address: jatrabari,dhaka

Division: Dhaka

Appointed Date: 2024-05-14

Order Status :

Cancelled

Update

Chapter 7: Conclusion

7.1 Project Link

Front-end: <https://github.com/nhsuprim/service-ui>

Back-end: <https://github.com/nhsuprim/service-api>

7.2 Limitations

- Two Factor Authentication facility isn't available.
- Email verification is not available.

7.3 Future Scope:

Many things can be simplified for better understanding. We can implement various coding strategies to enhance functionality. Currently, database backup is not possible, but in the next release, we will include this feature. This will allow customers to make decisions based on historical data. As new technologies emerge, we can adjust the system to better suit its intended environment. These advancements can also improve security, depending on potential risks.

Reference:

1. <https://www.elluminatiinc.com/e-services/>
2. <https://www.academia.edu>
- 3.

