

---

# **Software Requirements and Design Document**

**for**

## **<QuickFix: A Location-Based Service Management Application>**

**Prepared by <Faizan Liaqat**

**Malik Ali Awan**

**Muhammad Hamza Nadeem>**

**<NeoTech>**

**<November 26, 2024>**

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 Product Scope	1
1.3 Title	1
1.4 Objectives	1
1.5 Problem Statement	1
<b>2. Overall Description</b>	<b>1</b>
2.1 Product Perspective	1
2.2 Product Functions	2
2.3 List of Use Cases	2
2.4 Extended Use Cases	2
2.5 Use Case Diagram	2
<b>3. Other Nonfunctional Requirements</b>	<b>2</b>
3.1 Performance Requirements	2
3.2 Safety Requirements	2
3.3 Security Requirements	2
3.4 Software Quality Attributes	2
3.5 Business Rules	3
3.6 Operating Environment	3
3.7 User Interfaces	3
<b>4. Domain Model</b>	<b>3</b>
<b>5. System Sequence Diagram</b>	<b>3</b>
<b>6. Sequence Diagram</b>	<b>3</b>
<b>7. Class Diagram</b>	<b>4</b>
<b>8. Package Diagram</b>	<b>4</b>
<b>9. Deployment Diagram</b>	<b>4</b>

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to specify the software requirements for the QuickFix desktop application, focusing on its first release. This document provides a comprehensive overview of the application's objectives, scope, and core functionalities. QuickFix aims to create a streamlined system where clients can discover and book local service providers with minimal effort. The requirements detailed here reflect the system's primary components, interfaces, and user interactions.

This SRS covers the entire scope of QuickFix, including its client-side and service-provider functionalities. The product integrates features like booking, location-based recommendations, feedback management, and dispute resolution.

## 1.2 Product Scope

QuickFix is a robust solution for clients and service providers aiming to simplify the process of locating and hiring skilled professionals. The software focuses on achieving the following:

- **Benefits:** Enhance the client experience by reducing the time and effort needed to find reliable service providers. Similarly, service providers gain a platform to market their skills effectively.
- **Objectives:** Provide a localized experience, where clients receive intelligent recommendations based on their geographical location and service preferences.
- **Goals:** Establish a scalable platform capable of supporting individual households, businesses, and eventually larger regions.

By aligning with the growing demand for localized service platforms, QuickFix supports corporate goals of increasing accessibility and efficiency in service delivery, while positioning itself as a competitive alternative to apps like Uber for services or TaskRabbit.

## 1.3 Title

QuickFix: A Location Based Application for Establishing the Client-Server Relationship

## 1.4 Objectives

For the purpose of developing a convenient desktop application with which the service providers can create profiles to advertise their services.

For the clients who are seeking a range of services according to their area and type of service

required.

For location intelligent recommendations of the clients and for easy management of the service booking by clients.

To develop a feedback mechanism in form of rating and reviewing service providers to ensure the clients give credible feedback.

## 1.5 Problem Statement

In urban societies, identifying and hiring reliable service providers remains a major challenge. Traditional methods such as word-of-mouth recommendations or directory searches are often time-consuming and inefficient. Clients frequently encounter mismatches in service expectations, which impacts overall satisfaction and trust.

QuickFix addresses this gap by leveraging technology to offer a user-friendly platform where clients can connect with trusted service providers in their vicinity. By integrating location-based intelligence and a robust feedback mechanism, QuickFix ensures that users receive services tailored to their needs. Additionally, the system's scalability and ease of use cater to the growing demand for convenient and reliable service platforms.

**Feasibility:** The increasing reliance on digital solutions for everyday tasks highlights the feasibility of QuickFix. With smartphones and desktops becoming central to user experiences, the platform aligns perfectly with current technological trends and consumer behavior.

## 2. Overall Description

### 2.1 Product Perspective

QuickFix is a new, self-contained application designed to provide end-to-end service management for clients and service providers. Unlike conventional service directories or general-purpose platforms, QuickFix introduces the following key differentiators:

1. **Location Intelligence:** Real-time recommendations based on the client's geographical location.
2. **Streamlined User Experience:** Simplified booking processes and automated notifications ensure smooth client-provider interactions.
3. **Feedback Mechanism:** Transparent rating and review system to uphold service quality.
4. **System Integration:** The application interfaces with payment gateways, notification services, and dispute management systems to provide a holistic solution.

The following diagram outlines the major system components and interactions: Include a simplified system diagram here, showing modules like Client, Service Provider, Booking Service, Notification System, Payment System, and Feedback System.

## 2.2 Product Functions

QuickFix enables a variety of functionalities tailored to both clients and service providers:

- **Client Features:**
  - Search for and book services based on type and location.
  - Receive notifications for booking confirmations, job status updates, and completion prompts.
  - Provide feedback and rate service providers.
- **Service Provider Features:**
  - Create and manage service listings.
  - Update job statuses (e.g., "In Progress," "Completed").
  - Respond to booking requests and maintain availability.
- **Core System Features:**
  - Location-based recommendations to ensure proximity-based efficiency.
  - Integrated payment processing for seamless transactions.
  - History tracking for past services, disputes, and feedback.
  - Messaging system for direct communication between clients and service providers.

These features collectively contribute to achieving QuickFix's overarching goals of accessibility, reliability, and user satisfaction.

## 2.3 List of Use Cases

- Send Notification
- View Profile
- Booking
- Location-Based Recommendation
- Manage Payments
- Registration
- View History
- Job status updates
- Provide Feedback
- Create a Service
- Manage Dispute Resolution

## 2.4 Extended Use Cases:

The extended usecases are listed below

## Faizan Liaqat

### Use Case - Send Notification

**Use Case:** Send Notification

**Primary Actor:** Client

**Stakeholders and Interests:** Service Provider

**Scope:** QuickFix Application

**Level:** User-goal

**Precondition:** The client has successfully booked a service or the service provider has updated the service status.

**Success Guarantee:** The client and service provider receive timely notifications related to service requests.

**Trigger:** A client books a service or the service provider updates the job status (accepted, on the way, completed, etc.).

**Main Success Scenario:**

Actor	System
1. The client books a service using the QuickFix app	
	2. The system automatically sends a booking confirmation notification to the service provider.
3. The service provider accepts the service request and updates the job status.	
	4. The system sends a job acceptance notification to the client.
5. The service provider updates the status.	
	6. The system notifies the client that the provider is en route.
7. The service provider marks the job as completed	

	8. The client receives a notification indicating the job is finished, with a prompt to rate and review the service.
--	---

**Extensions :**

- 4a If the service provider declines the job, the system sends a decline notification to the client and suggests alternate providers.  
 5a If the provider does not update the job status within a specified time, the system sends a reminder notification to the provider.  
 6a If the client cancels the job before the provider arrives, the system sends a cancellation notification to the provider.

**Use Case - View Profile****Use Case:** View Profile**Primary Actor:** Buyer, Seller**Stakeholders and Interests:** Admin, Banks, Taxation Institutes**Scope:** QuickFix Application**Level:** User-goal**Precondition:** The user (buyer or seller) must be logged in to their QuickFix account.**Success Guarantee:** The user successfully views their profile details.**Main Success Scenario:**

Actor	System
1. The user logs into their QuickFix account.	
	2. The system verifies the user's login credentials
3. The user locates the "View Profile" option in the main menu and selects it.	
5. The user successfully views their profile information.	4. The system retrieves and displays the user's profile information

**Extensions :**

- 2a If the login credentials are incorrect, the system displays an error message and prompts the user to try again.  
 3a If the "View Profile" option is not available due to a system error, the user is informed

with a system notification, and the error is logged for troubleshooting.  
 4a If the profile information fails to load (e.g., due to a database error), the system displays an error message, and the user is prompted to try again later.

### Use Case - Booking

**Use Case:** Booking

**Primary Actor:** Buyer

**Stakeholders and Interests:** Seller, Admin

**Scope:** QuickFix Application

**Level:** User-goal

**Precondition:** The user (buyer) must be logged in to their QuickFix account

**Success Guarantee:** The user successfully books a service with a seller.

#### Main Success Scenario:

Actor	System
1. The user logs into their QuickFix account.	
	2. The system verifies the user's login credentials
3. The user navigates to the "Booking" option and selects it.	
	4. The system presents a list of available sellers based on location and service type.
5. The user selects a seller and submits a booking request (including preferred time and service details). 1.	6. The system checks the availability of the selected seller for the requested service.
	7. The system notifies the seller of the booking request.
8. The seller accepts the job request.	1.

	9. The system confirms the booking with the buyer and provides details (e.g., time, contact information, price).
10. The user (buyer) successfully books the service.	

**Extensions :**

- 2a If the login credentials are incorrect, the system displays an error message and prompts the user to try again.
- 6a If the seller is unavailable for the requested time, the system notifies the user and suggests alternative times or providers.
- 7a If the seller does not respond within a specific time window, the system automatically cancels the request and notifies the buyer with suggested alternatives.
- 8a If the seller declines the job request, the system notifies the user and suggests alternate available sellers.
- 9a If there is an error in confirming the booking, the system retries the confirmation or prompts the user to try again later.

**Use Case - Location-Based Recommendation****Use Case:** Location-Based Recommendation**Primary Actor:** Buyer**Stakeholders and Interests:** Seller**Scope:** QuickFix Application**Level:** User-goal**Precondition:** The user (buyer) must be logged in, and location services must be enabled.**Success Guarantee:** The user successfully views services offered by nearby sellers.**Main Success Scenario:**

Actor	System
1. The user logs into their QuickFix account.	
	2. The system verifies the user's login credentials
3. The user navigates to the "Services" option and selects it.	
	4. The system displays the services page.

<p>5. The buyer navigates to the "Nearby Sellers" option and selects it..</p> <p>8. The buyer successfully views nearby sellers and their profiles.</p>	<p>6. The system checks the buyer's current location.</p> <p>7. The system displays a list of sellers near the buyer, based on the service selected.</p>
---	--

**Extensions :**

- 2a If the login credentials are incorrect, the system displays an error message and prompts the user to try again.
- 5a If location services are not enabled or permissions are denied, the system displays a message requesting the user to enable location services.
- 6a If the system cannot determine the buyer's location, the system asks the user to manually input their location.
- 7a If no sellers are available nearby, the system displays a message suggesting expanding the search radius or trying again later.
- 7b If there is an error retrieving the seller list (e.g., due to server failure), the system prompts the user to retry or displays a fallback list of sellers.

**Ali Awan****Use Case - Manage Payments**

- a. **Use Case Name:** Manage Payments
- b. **Scope:** QuickFix Application
- c. **Level:** User Level Goal
- d. **Primary Actor:** Buyer
- e. **Stakeholders and Interests:** buyer, seller, payment verification system, tax agencies, company
- f. **Preconditions:**
  - 1. User must be logged in to system.
  - 2. Buyer must have selected a service.
  - 3. Buyer must have chosen a payment method
- g. **Postconditions:**  
Payment is Successful

**h. Main Success Scenario:**

Actor	System
1. Buyer is logged in	
2. Buyer selects a service	
3. Buyer selects payment method	

4. If card option is added, buyer adds card credentials	
	5. System verifies credentials

6. User confirms payment

7. Payment service verifies the payment

8. System displays confirmation message to buyer

9. Seller receives payment

i. **Extensions:**

5a: credential verification failed

7a: payment service unable to verify payment

**Use Case - Registration**

j. **Use Case Name:** Registration

k. **Scope:** QuickFix Application

l. **Level:** User Level Goal

m. **Primary Actor:** Buyer and Seller (User)

n. **Stakeholders and Interests:** buyer, seller

o. **Preconditions:**

User must have an email and phone number.

p. **Postconditions:**

User Registers in to the system

q. **Main Success Scenario:**

Actors	System
1. User provides a credentials such as name, email, address, phone number etc.	
2. User sets a username and password.	
	3. System verifies the details
	4. Confirmation message is shown to user

5. User is registered.

**r. Extensions:**

- 2a: username already in use
- 3a: details verification failed

**Use Case - View History**

**s. Use Case Name:** View History

**t. Scope:** QuickFix Application

**u. Level:** User Level Goal

**v. Primary Actor:** Buyer and Seller

**w. Stakeholders and Interests:** buyer, seller, company

**x. Preconditions:**

- 1. User(Buyer/seller) must be logged in to system.
- 2. User must have some purchase (incase of buyer) or selling(incase of seller)

**y. Postconditions:**

User has viewed their history

**z. Main Success Scenario:**

Actor	System
1. User is logged in	
2. User navigates through view history	
	3. A. If a user is buyer, the system will display buyed services in history B. If a user is seller, the system will display sold services.

4. User can select a past service to view more details

**aa.Extensions:**

3a: system fails to show history

4a: System cannot show details of past services

**Use Case - Job status updates**

**bb. Use Case Name:** Job status update

**cc. Scope:** QuickFix Application

**dd. Level:** User Level Goal

**ee. Primary Actor:** Seller

**ff. Stakeholders and Interests:** seller, buyer, admin, company

**gg. Preconditions:**

- 1. Seller must be logged in to system.
- 2. Seller must have selected a job for which he wants to update the status.

**hh. Postconditions:**

Job status is updated

**ii. Main Success Scenario:**

Actors	System
1. The Seller logs into the system and navigates to the "My Jobs" section.	
2. The Seller selects a specific job to update.	
	3. The system displays the current status of the job (e.g., "Pending", "In Progress", "Completed").
4. The Seller chooses to update the status and selects a new status from available options (e.g., "In Progress", "Completed").	5. The system updates the job status.

6. The Buyer is notified of the job status change via confirmation message in the system

7. The job status is reflected in both the Seller and Buyer's dashboard.

**jj. Extensions:**

**5a:** The system fails to update job status

**Hamza Nadeem**

### Use Case - Provide Feedback

**A. Use Case Name:** Provide Feedback

**B. Scope:** QuickFix Application

**C. Level:** User goal

**D. Primary Actor:** Buyer- buyer providing feedback regarding service

**E. Stakeholders and Interests:**

- 1. Buyer- buyer providing feedback regarding service
- 2. Seller - interested in receiving feedback to improve service quality

**F. Preconditions:**

- 1. The Buyer logs into the system.
- 2. The service for which feedback is being provided has been utilised.
- 3. The feedback form is available and accessible.

**G. Postconditions:**

Feedback is submitted and recorded in the system . The Buyer receives an acknowledgment notification.

**H. Main Success Scenario:**

Actor	System
1.The Buyer/Seller logs into QuickFix.	2.The system authenticates the Buyer/Seller.
3.The Buyer/Seller navigates to the "Feedback" section.	
	4.The system provides a feedback form with various input fields.
5.The Buyer/Seller fills the form.	
6.The Buyer/Seller clicks on the "Submit" button and submits the form.	7.The system validates the feedback form fields.

8.The system stores the feedback in the database.

9.The system sends a confirmation notification to the Buyer/Seller when the feedback is successfully submitted.

**I. Extensions:**

7a.If the Buyer/Seller doesn't complete all required fields, the system prompts the user to fill in missing fields before submission.

8a.If the feedback submission fails due to a system error, the system displays an error message and offers an option to retry.

8b.If the feedback data fails to save in the database, the system notifies the admin to investigates the issue.

**Use Case - Create a Service**

**A. Use Case Name:** Create a Service

**B. Scope:** QuickFix Application

**C. Level:** User goal

**D. Primary Actor:** Seller- interests in selling a service

**E. Stakeholders and Interests:**

Buyer- Interested in viewing available services and finding the right seller.

Seller- Interested in showcasing their services effectively to attract buyers.

**F. Preconditions:**

1. The Seller or Buyer logs into the system.
2. Services Details are shown on the Service page.

**G. Postconditions:**

The Seller successfully lists their service, and it is available for buyers to view.

**H. Main Success Scenario:**

Actor	System
1.The Seller logs into QuickFix.	
3.The Seller navigates to the "Service" section.	2.The system authenticates the Seller.
4.The Seller creates a new Service.	
5.The Seller uploads all the required information regarding his service.	
6.The Seller clicks on the submit button.	

7.The System verifies the information and approves the gig.

**I. Extensions:**

5a.If the Seller doesn't complete all required fields, the system prompts the user to fill in missing fields before submission.

7a. If The service details does not meet requirement , the system disapproves the Service

**Use Case - Send and Receive Messages**

**A. Use Case Name:** Send and Receive Messages

**B. Scope:** QuickFix Application

**C. Level:** User goal

**D. Primary Actor:**

Buyer- sends and receive message

**E. Stakeholders and Interests:**

Buyer- sends and receive message

Seller- interested in maintaining clear communication with buyers

Support Team- uses messages to assist customers with issues

**F. Preconditions:**

1. The buyer is logged into the system.

2. The messaging functionality is enabled and available.

3. The buyer has access to the seller to initiate a conversation or continue an existing one.

**G. Postconditions:**

Messages are successfully sent and received, and the conversation is updated in the system.

**H. Main Success Scenario:**

Actor	System
1.The Buyer/Seller logs into QuickFix.	2.The system authenticates the Buyer/Seller.

3.The Buyer/Seller navigates to the "Messages" section.	
4.The Buyer/Seller selects a conversation.	
	5.The system retrieves and displays the selected conversation.
6.The Buyer/Seller types a message in the message box.	
7.Buyer/Seller clicks on the "Send" button. 10.Buyer/Seller clicks on the notification.	8.The system sends the message to the Buyer/Seller.  9.The system triggers a notification for the Buyer/Seller about the new message.
	11.The system opens the "Messages" section and displays the Buyer's message to the Buyer/Seller.

**I. Extensions:**

**8a.** If the message fails to send due to connectivity issues, the system notifies the buyer of the failure and allows the message to be saved in a draft state for later sending.

**Use Case - Manage Dispute Resolution**

**A. Use Case Name:** Manage Dispute Resolution

**B. Scope:** QuickFix Application

**C. Level:** User goal

**D. Primary Actor:**

Buyer- wants to resolve the dispute efficiently

Seller- wants to resolve the dispute to maintain customer satisfaction

**E. Stakeholders and Interests:**

Buyer- wants to resolve the dispute efficiently

Seller- wants to resolve the dispute to maintain customer satisfaction

Dispute Resolution Team- Responsible for resolving the disputes

**F. Preconditions:**

1. The Buyer logs into the QuickFix.
2. The Buyer has bought the service.
3. A dispute-worthy event has occurred..

**G. Postconditions:**

The dispute is resolved and recorded in the system, and both parties receive notifications of the outcome.

**H. Main Success Scenario:**

Actor	System
1.The Buyer/Seller logs into QuickFix.	
3.The Buyer/Seller navigates to the "Dispute Resolution" section.	2.The system authenticates the Buyer/Seller.
	4.System displays a form with fields for "Subject", "Description", "Order Id", and an option to attach screenshots of the conversation.
5.Buyer/Seller writes the "Subject", "Description", "Order Id", and attaches screenshots	

6.Buyer/Seller submits the form

7.System saves the form data and generates a Dispute ID.

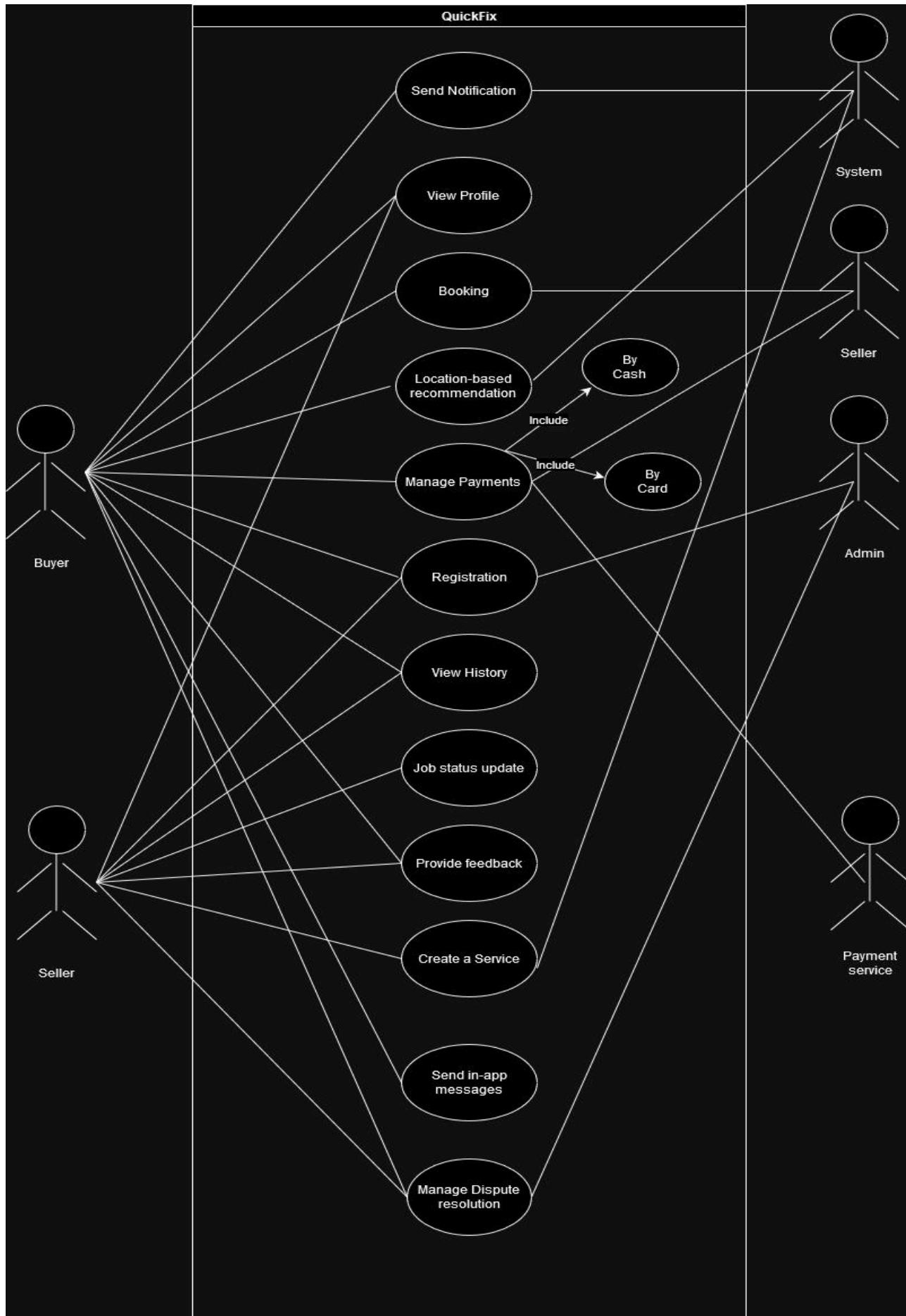
8.Buyer/Seller receives a confirmation notification

9.Buyer/Seller receives the Dispute resolve email within 3 business days

**I. Extensions:**

- 5a.The Buyer/Seller uploads a screenshot or document that exceeds the file size limit,the system prompts the Buyer/Seller to reduce the file size and try again.
- 6a.The Buyer/Seller fails to provide all required information, the system prompts the Buyer/Seller to complete missing fields before proceeding.
  - 9a.The Buyer/Seller does not receive a resolution within the promised timeframe,the system automatically sends a follow-up notification to both parties, indicating the delay and estimated resolution time.

## **2.5 Use Case Diagram**



## 3. Other Nonfunctional Requirements

### 3.1 Performance Requirements

- Response time for location-based service recommendations must not exceed **2 seconds** for 95% of queries.
- Notifications for booking confirmation, job updates, and service completion must be delivered within **1 second** of the triggering event.
- The application should support **24/7 uptime**, with downtime not exceeding **0.1% per month**, ensuring continuous availability.

### 3.2 Safety Requirements

- The application must prevent unauthorized data modification to avoid harm to user accounts, bookings, or payments.
- A fail-safe mechanism should ensure that incomplete payment transactions do not result in service activation or financial errors.
- Error messages displayed to users should not expose sensitive system information to prevent misuse or misinterpretation.

### 3.3 Security Requirements

- Restrict access to admin functions to authorized personnel with role-based access control.
- Ensure service providers cannot view client-specific sensitive information beyond job details.
- Maintain logs of all access attempts, successful or otherwise, for auditing purposes.

### 3.4 Software Quality Attributes

- **Reliability:** Ensure system uptime of **99.9%** and fault-tolerant design to recover from partial failures.
- **Usability:** The interface must prioritize ease of navigation with user-friendly features like tooltips, clear error messages, and accessible designs for various demographics.
- **Adaptability:** The application should support integration with emerging location services and payment gateways.
- **Maintainability:** Modular architecture to simplify debugging and feature addition.
- **Scalability:** Ability to support future expansions to larger regions or higher user loads without architecture overhaul.

### 3.5 Business Rules

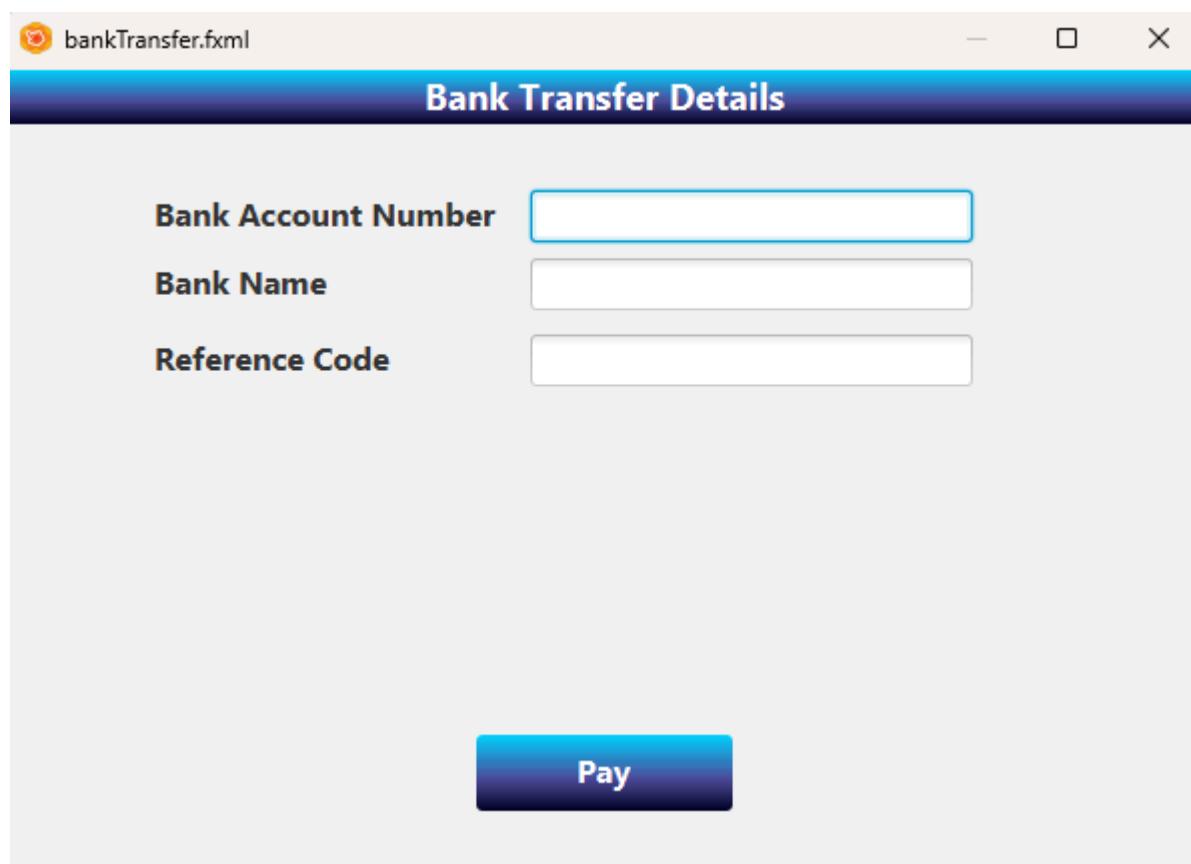
- Only registered clients can book services, provide feedback, and manage disputes.
- Service providers can only update job statuses or manage their listings after successful verification by the system.
- Disputes can only be raised for completed
- Feedback submission is restricted to clients who have utilized the service.

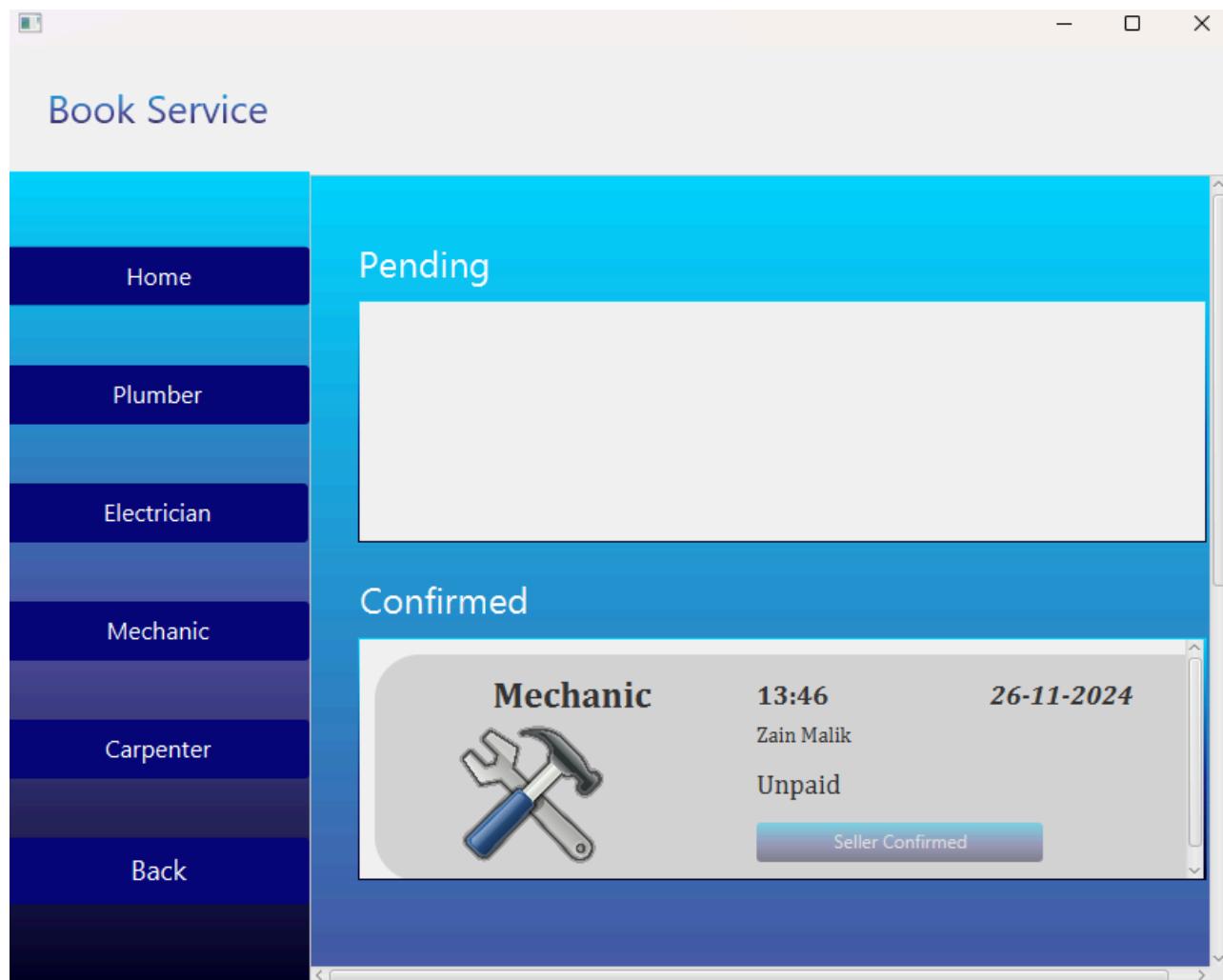
### 3.6 Operating Environment

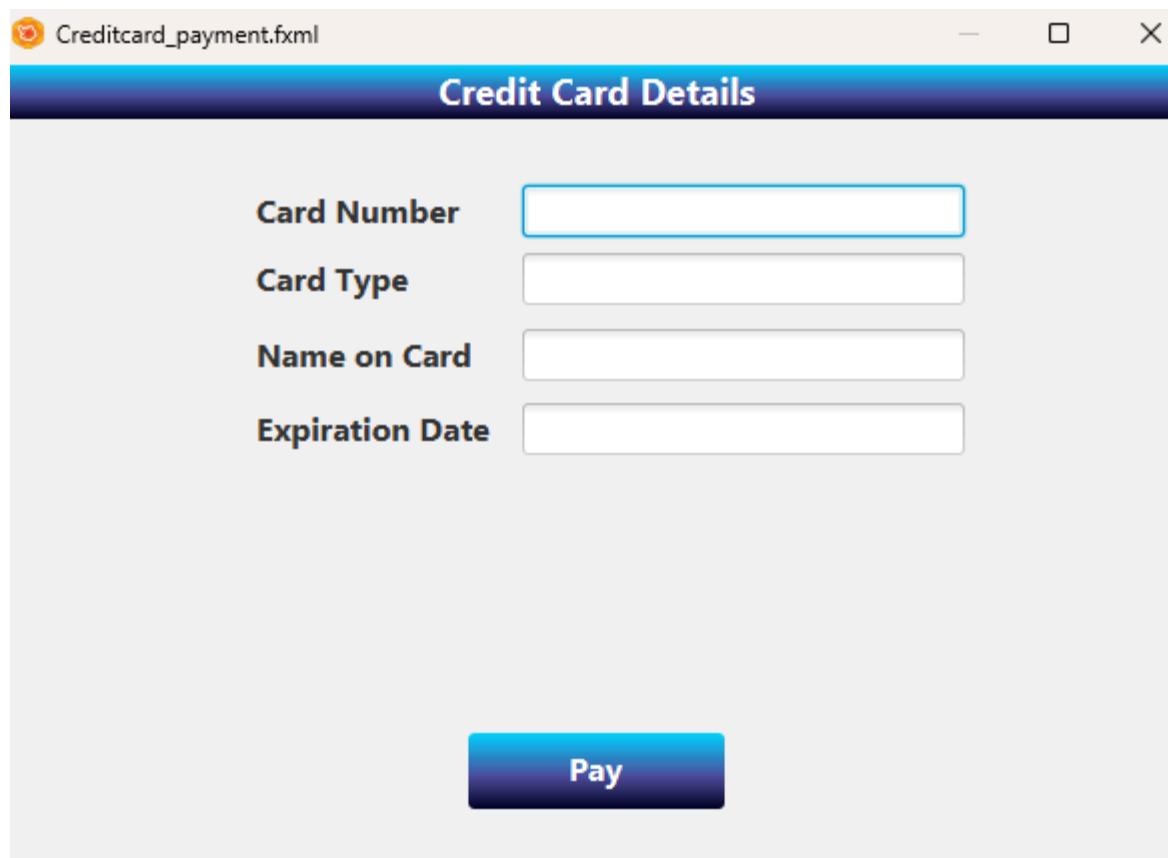
- **Hardware Requirements:**
  - Minimum: Intel i5 (or equivalent), 8 GB RAM, 256 GB SSD.
  - Recommended: Intel i7 (or equivalent), 16 GB RAM, 512 GB SSD.
- **Operating Systems:**
  - Windows 10/11
- **Software Dependencies:**
  - JavaFX Eclipse
  - Database: MySQL.

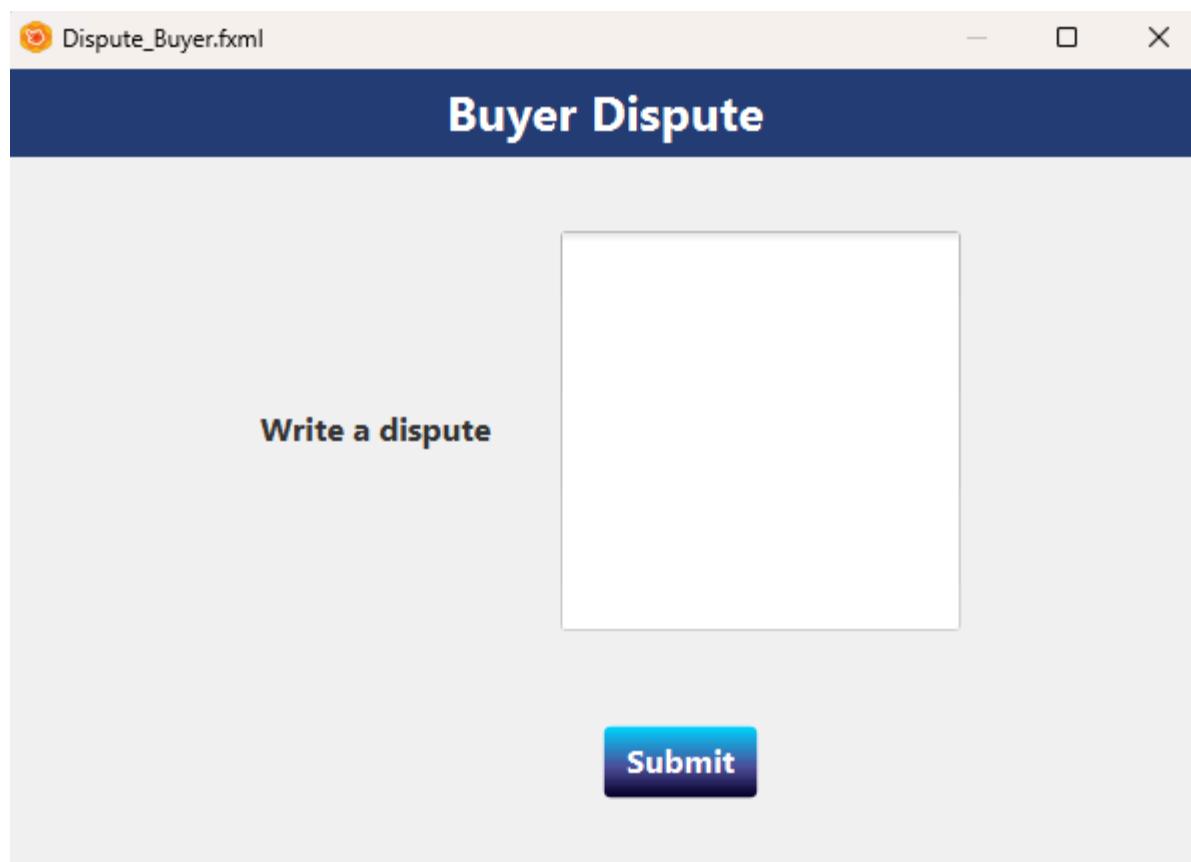
### 3.7 User Interfaces

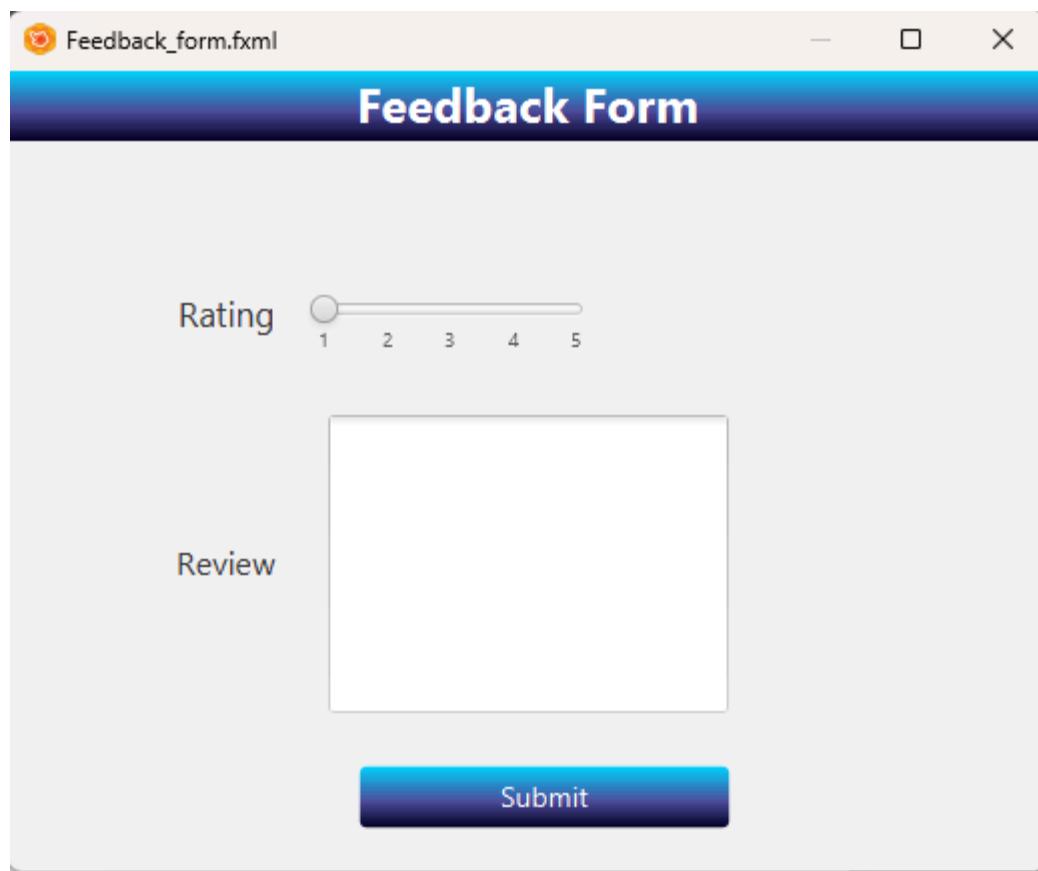
- **General Interface Standards:**
  - Clean and minimalistic design adhering to Material Design guidelines.
  - Consistent placement of navigation menus and action buttons across screens.
- **Features:**
  - **Home Screen:** Search bar for services and access to notifications.
  - **Service Booking:** Dropdown menus for selecting service types, locations, and times.
  - **Service Provider Dashboard:** Tools for creating, updating, and managing service listings.
  - **Admin Panel:** Controls for managing disputes.
- **Error Messages:**
  - Display user-friendly messages such as "Invalid credentials. Please try again." instead of technical errors.



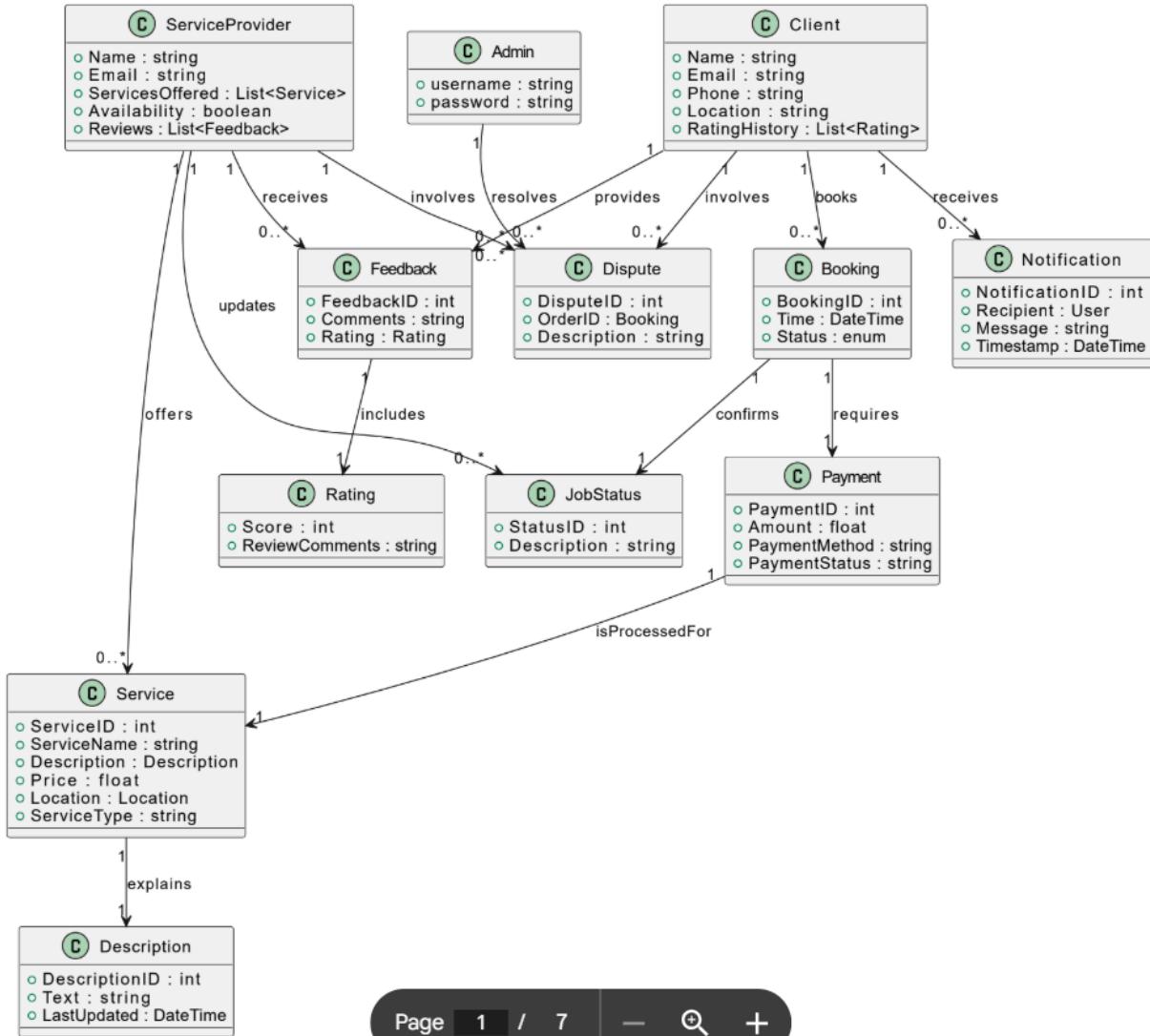




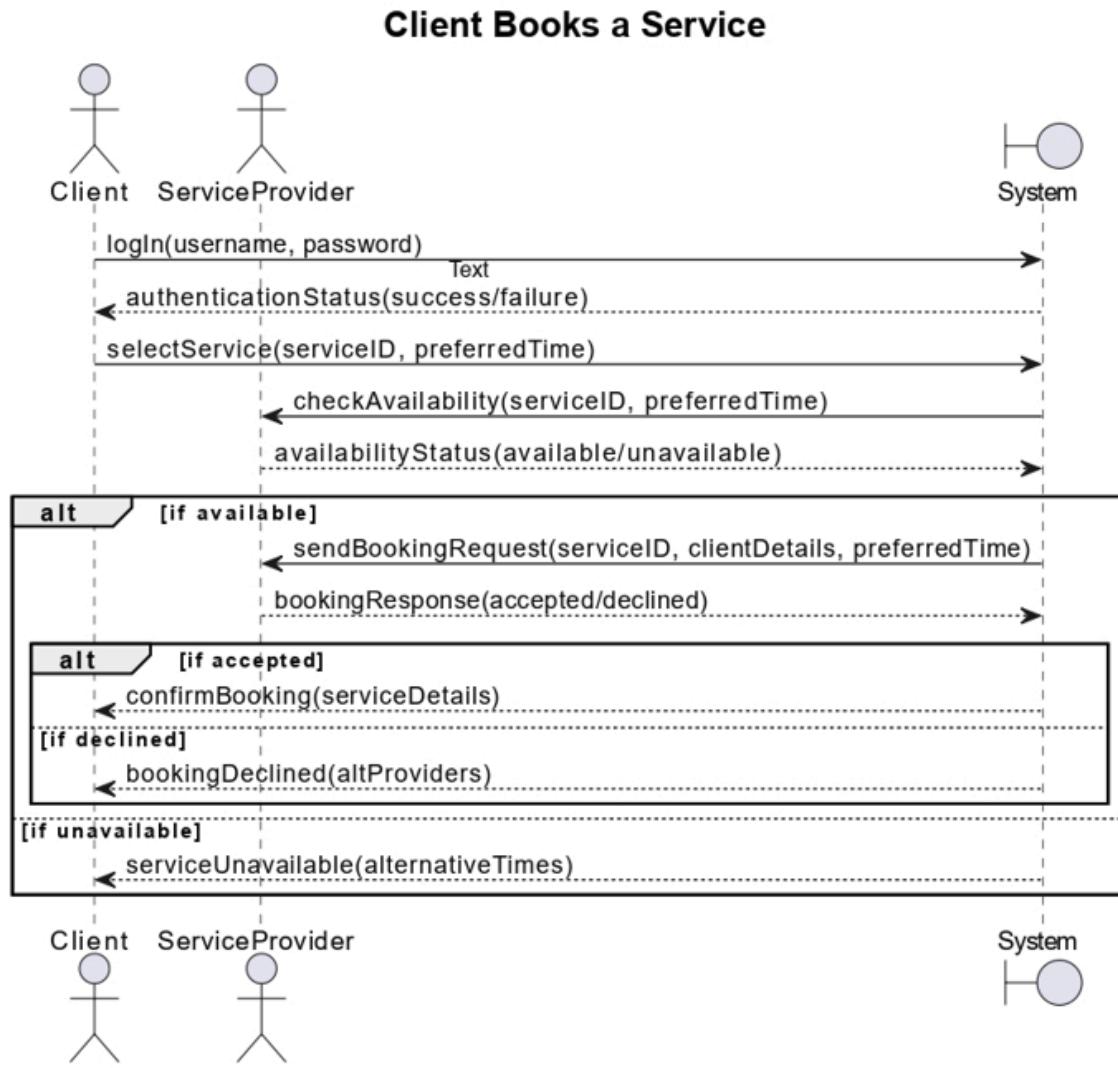


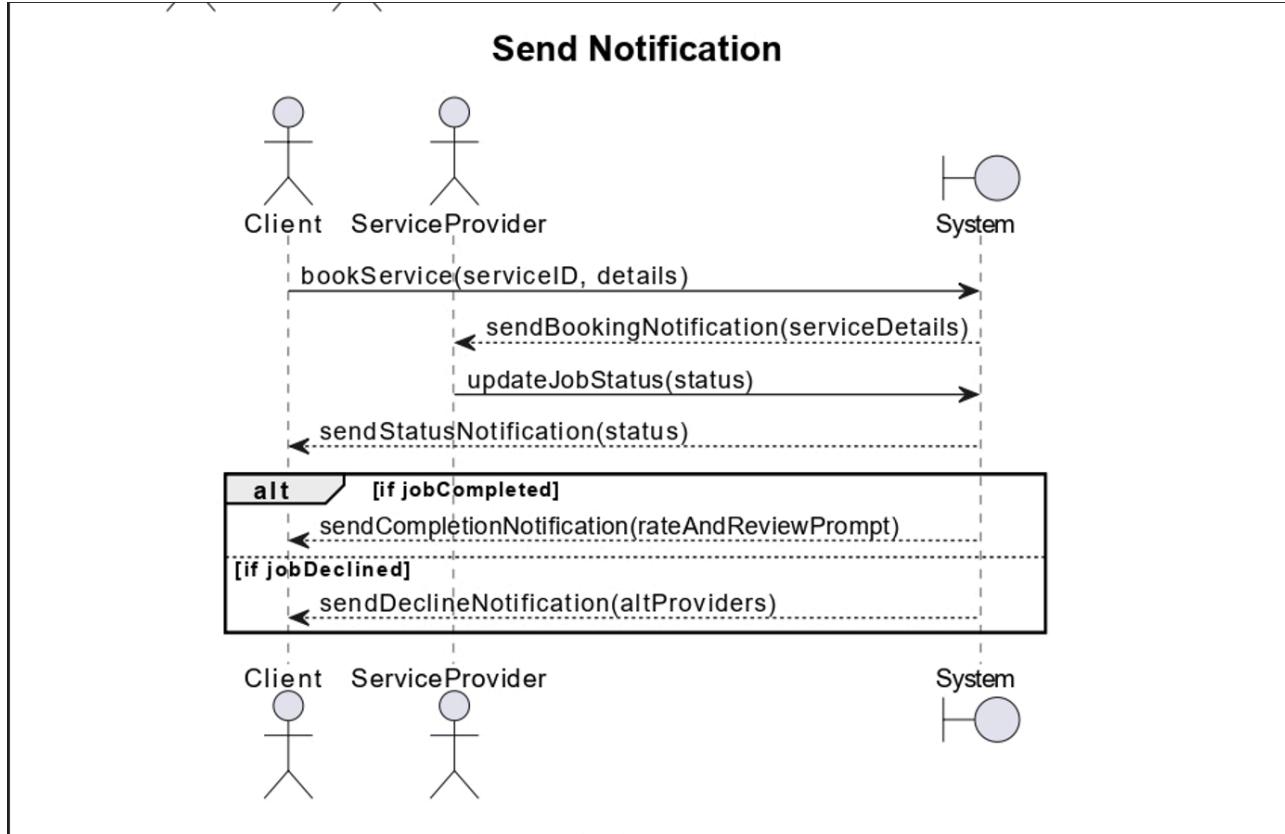


## 4. Domain Model

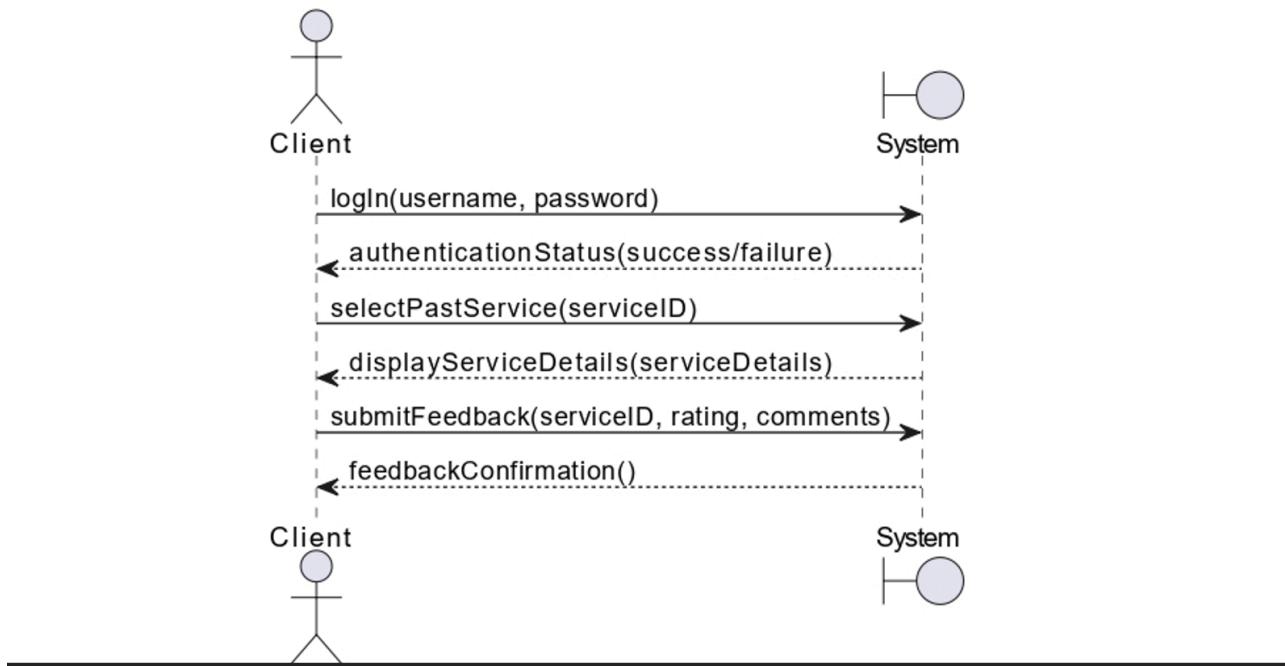


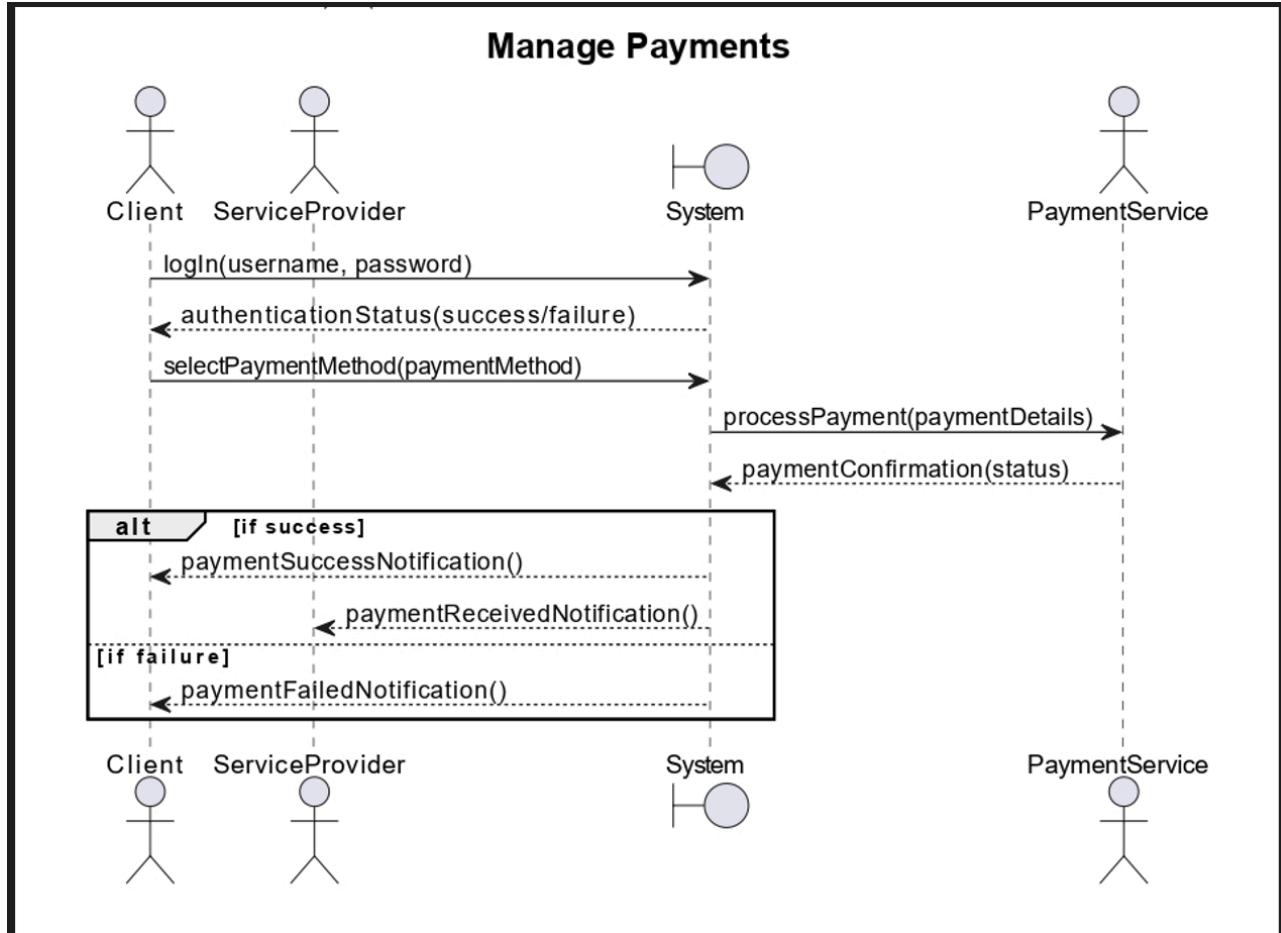
## 5. System Sequence Diagram



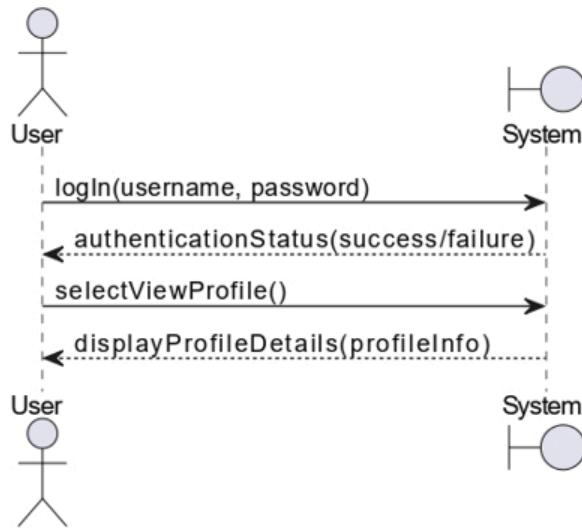


### Provide Feedback

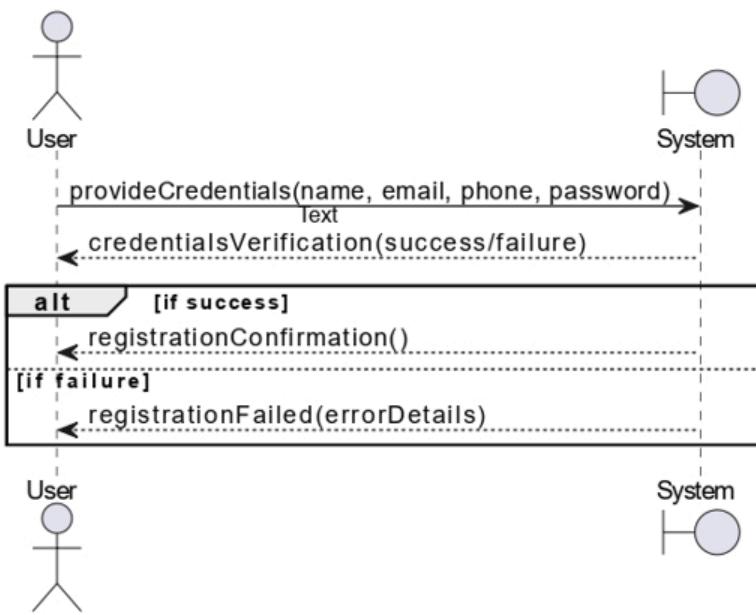




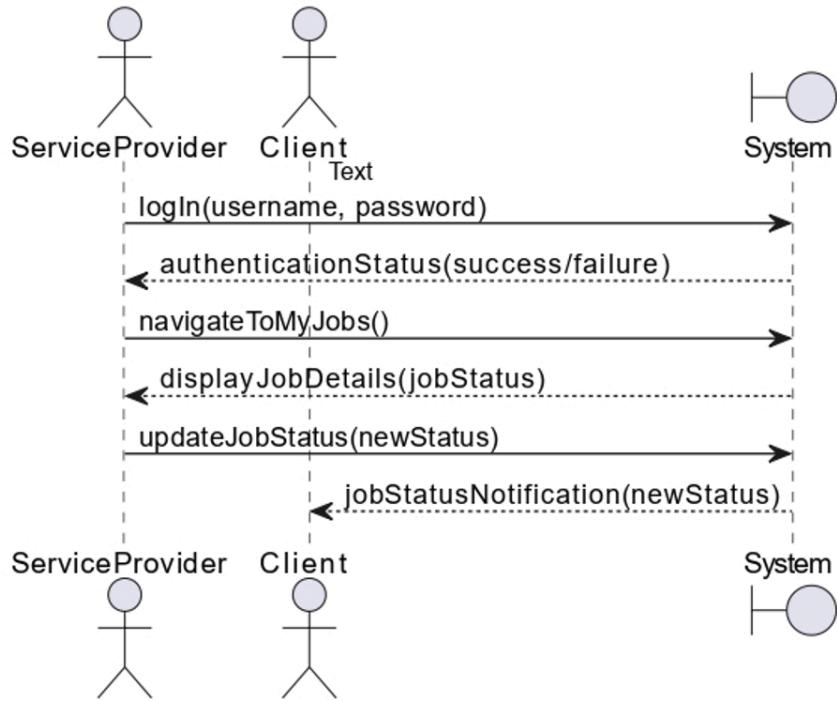
### View Profile



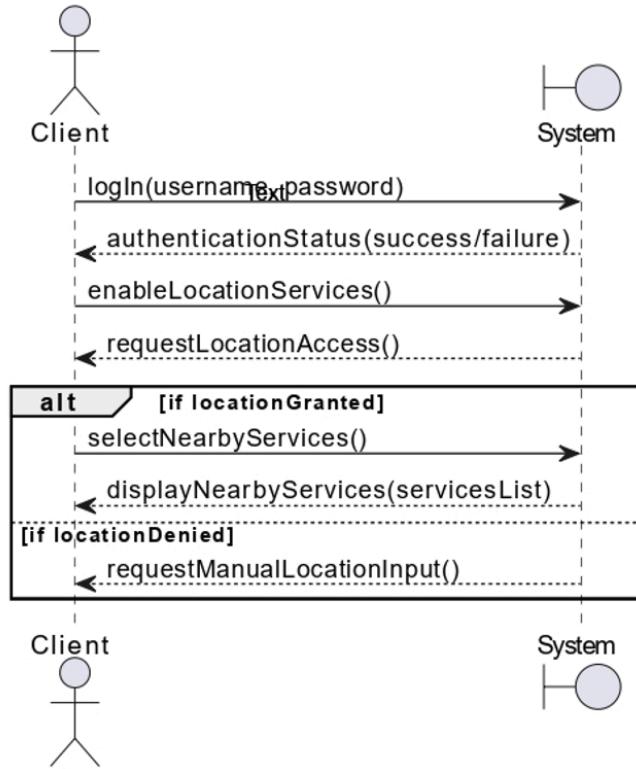
### Registration

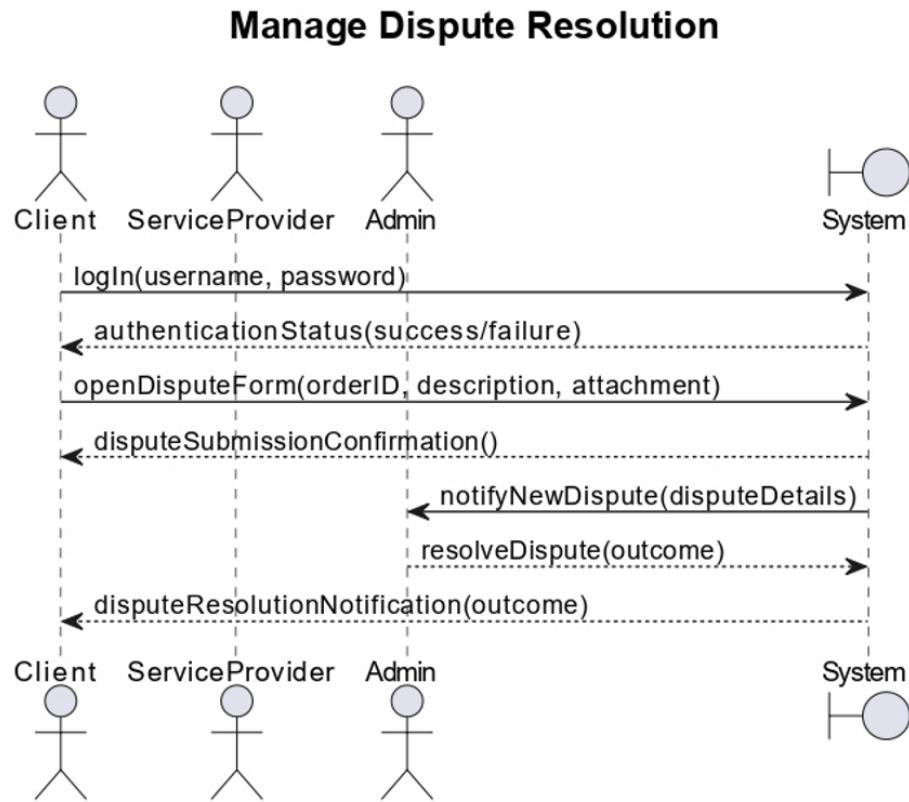


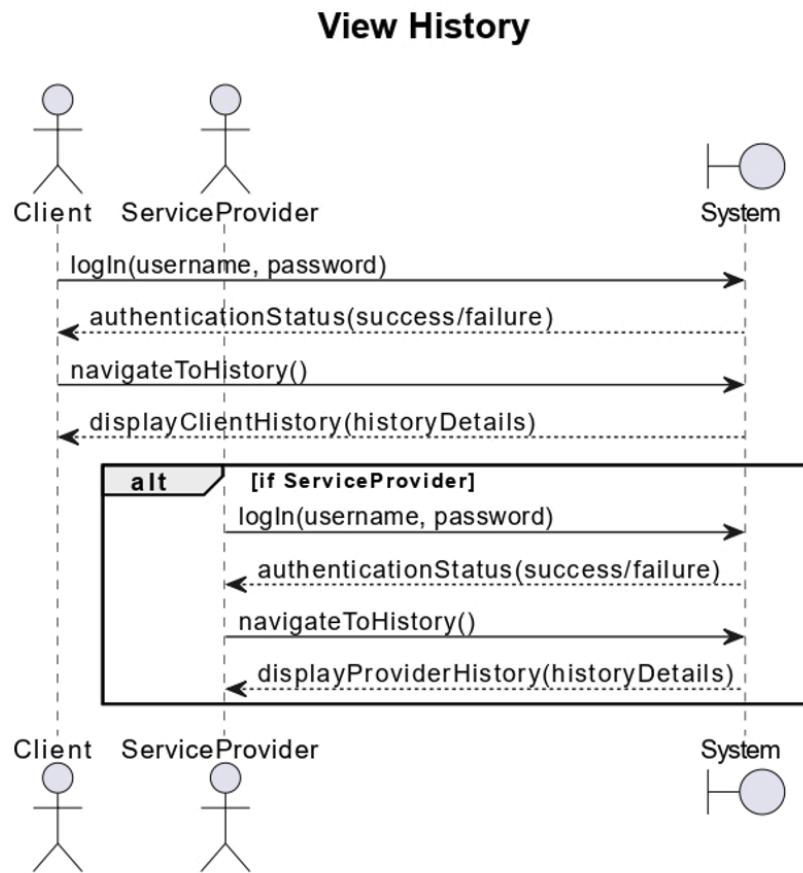
## Job Status Updates



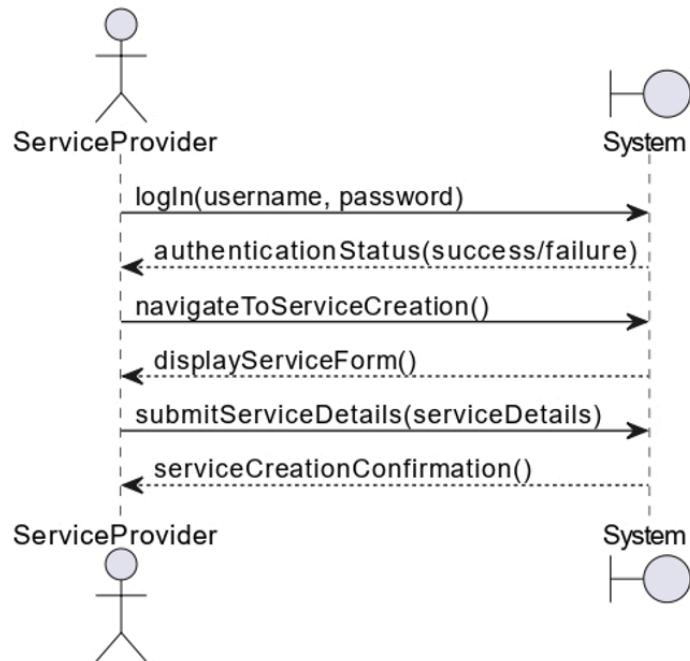
## Location-Based Recommendation



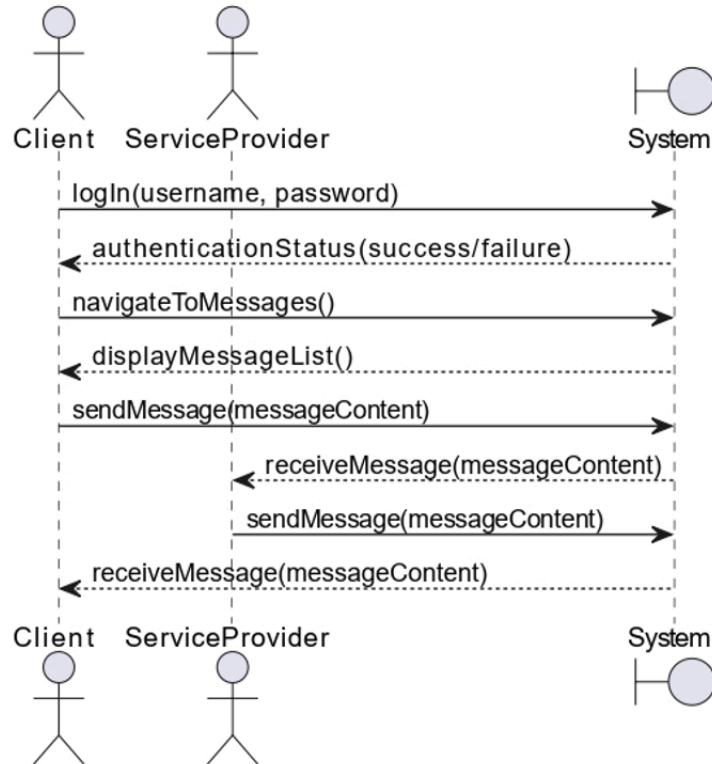




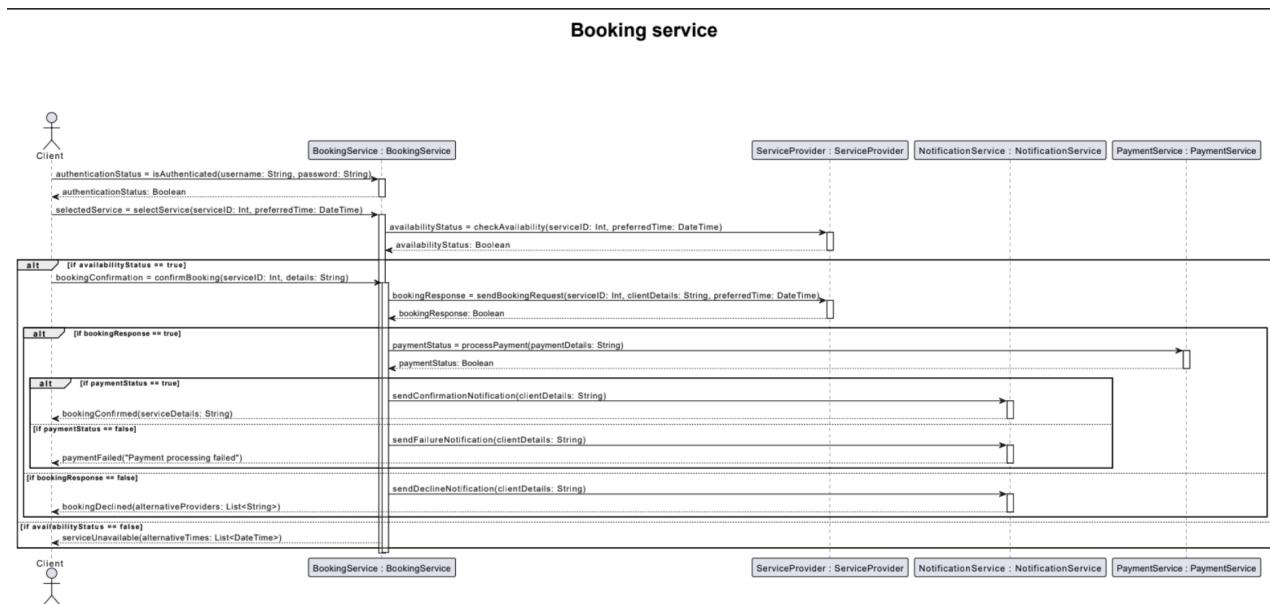
## Create a Service



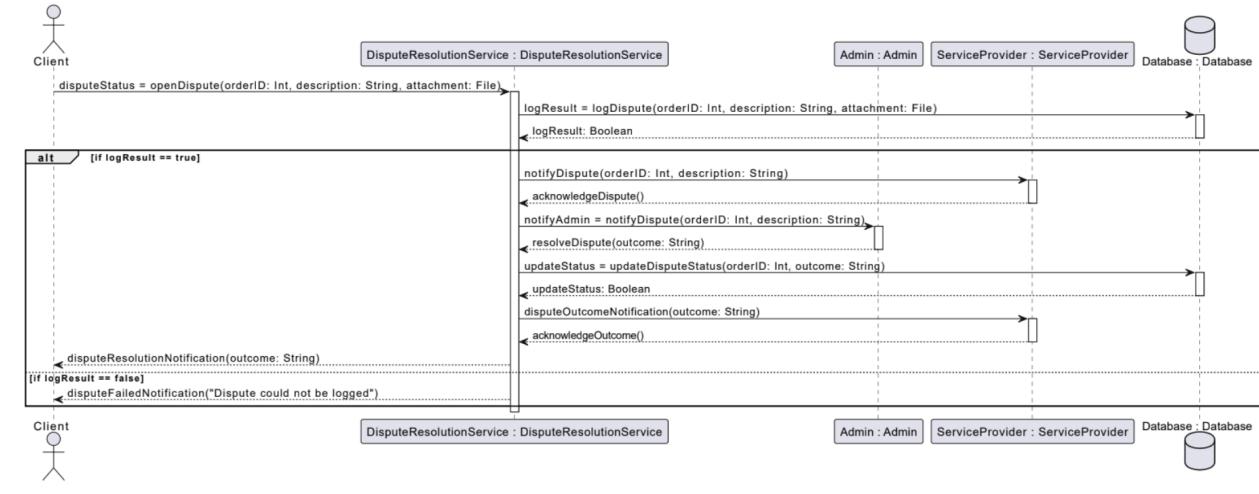
## Send and Receive Messages



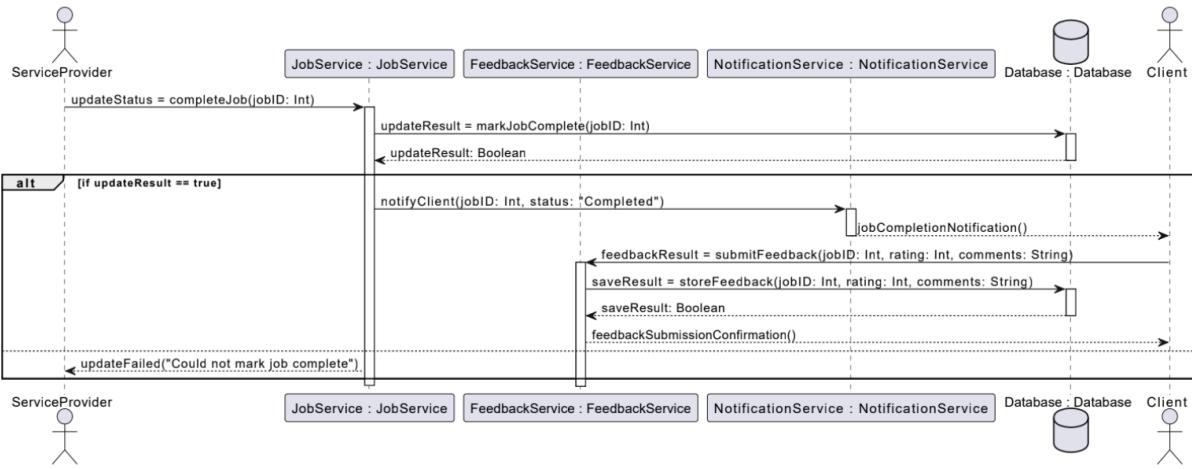
## 6. Sequence Diagram



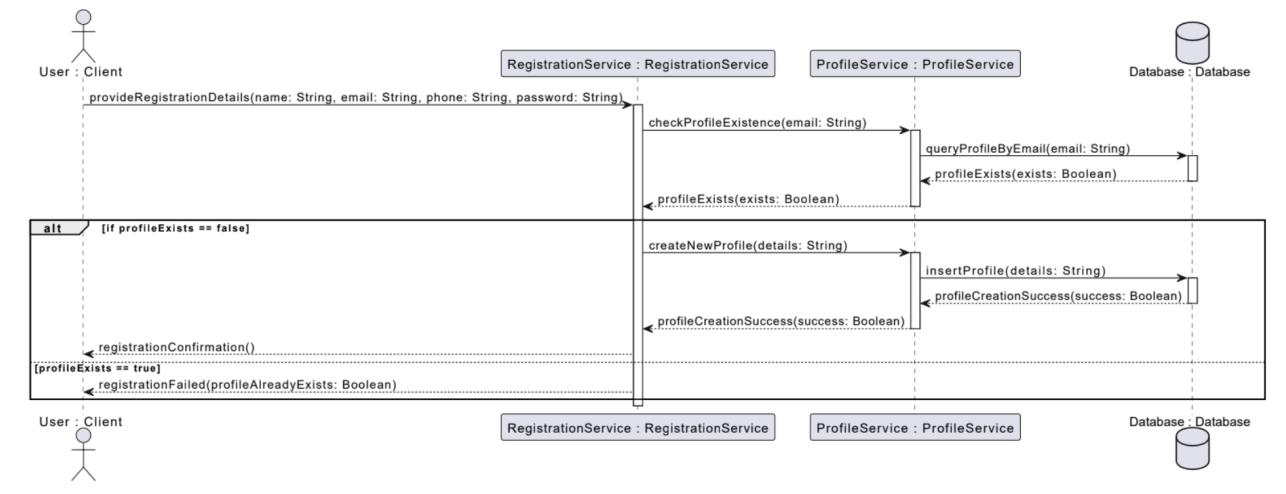
## Dispute Resolution



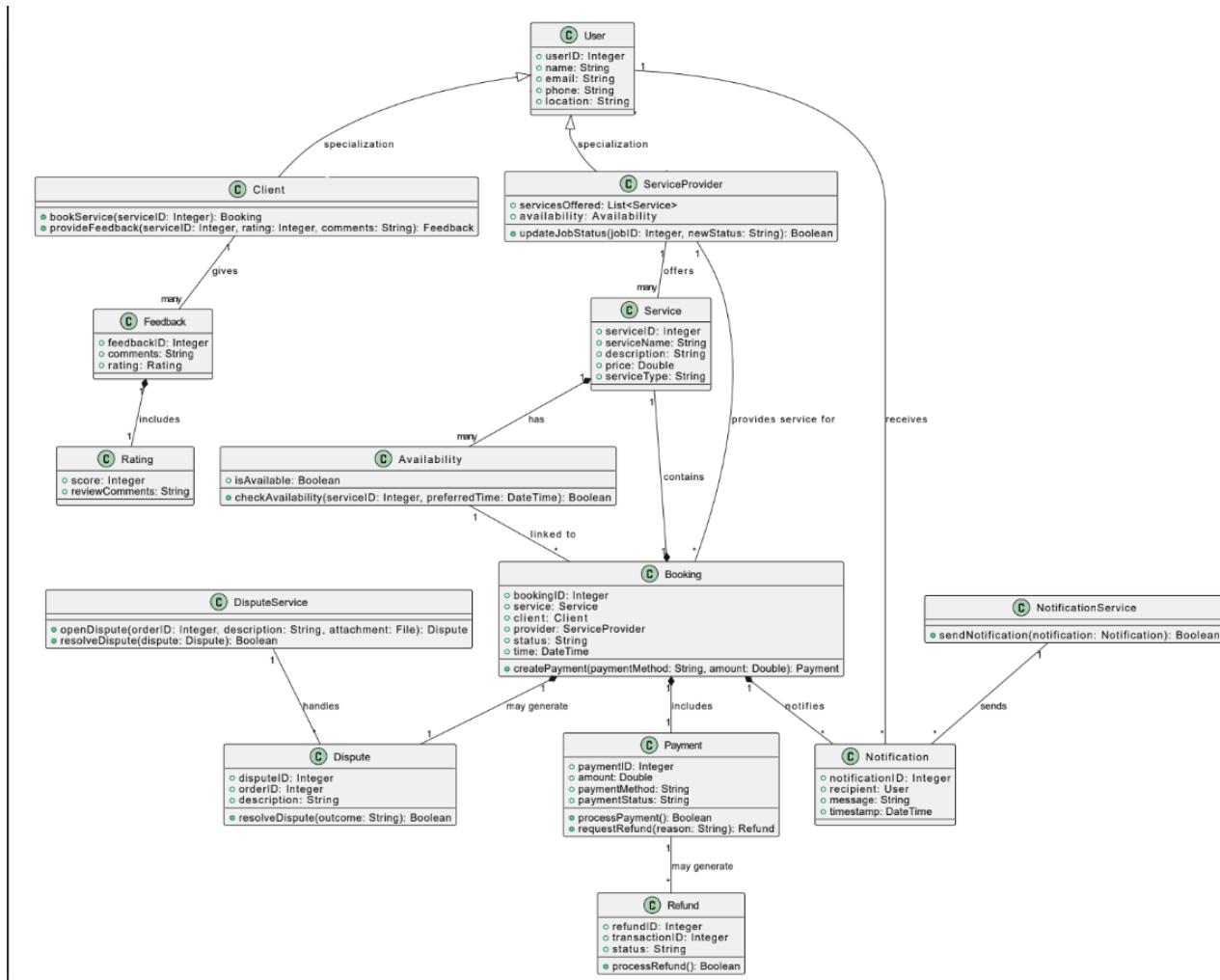
## Client Provides Feedback Sequence Diagram



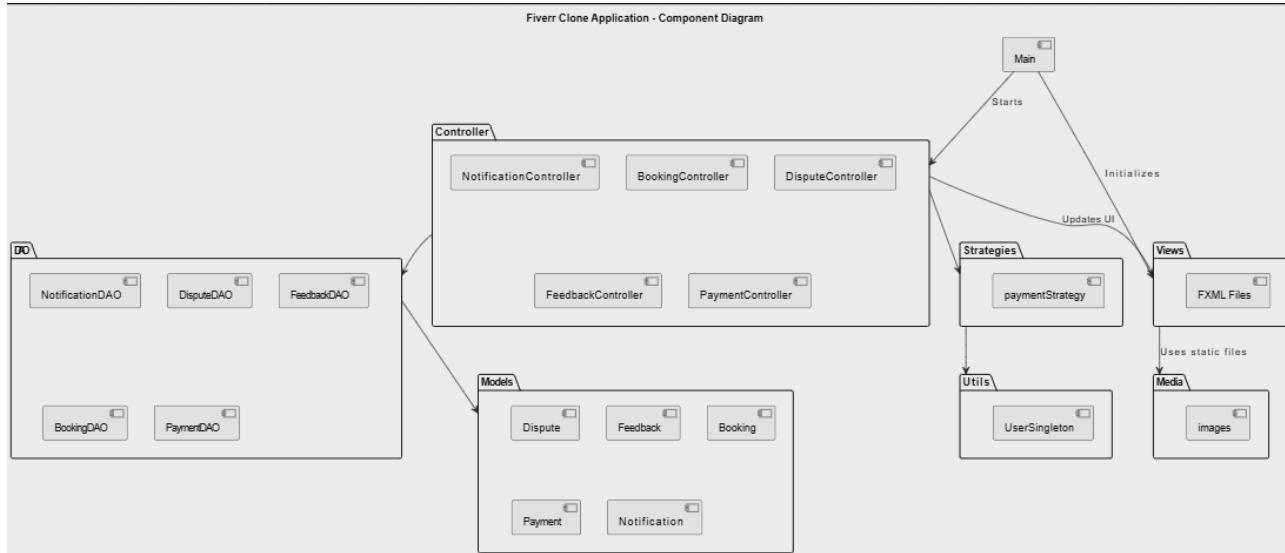
## Registration



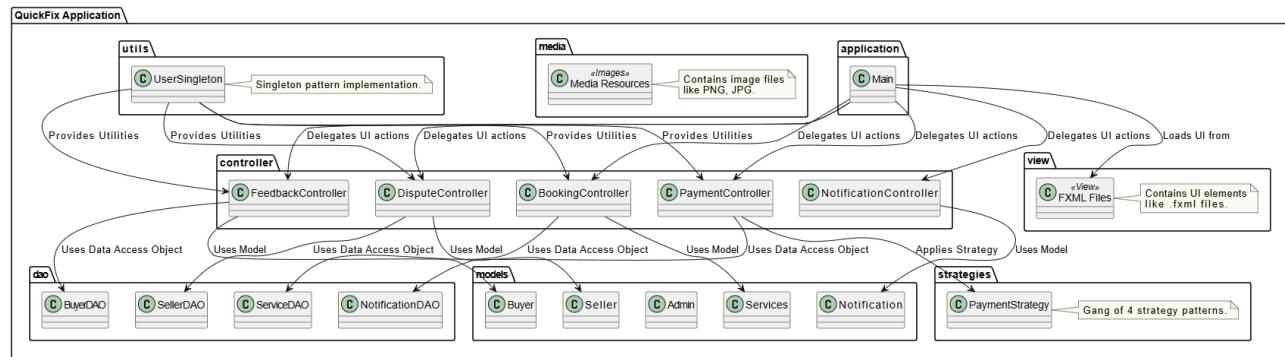
## 7. Class Diagram



## 8. Component Diagram



## 9. Package Diagram



## 10. Deployment Diagram

