Software Requirements and Design Document

for

Bookify

Afrah Syed (22i1008), Abdullah Aslam (22i0784), Abdur Raheem (22i0777)

Table of Contents

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

1. Introduction

1.1 Purpose

The purpose of this document is to outline the functional and non-functional requirements for "Bookify," a comprehensive booking management system. The focus is on facilitating seamless bookings across various services like hotels and restaurants, while enhancing user experience and integrating diverse features for both end-users and service providers.

1.2 Product Scope

Bookify is a unified booking platform, providing a one-stop solution for users to book various services, consolidating multiple apps into one. This platform aims to simplify the booking process by providing features such as notifications, cancellation management, and targeted offers, alongside tools for service providers to manage services and analytics.

1.3 Title

Bookify: A Unified Platform for Seamless Service Bookings

1.4 Objectives

- Simplify the process of booking diverse services through one platform.
- Improve user experience with intuitive and efficient navigation.
- Provide service providers tools for managing bookings, offering promotions, and viewing analytics.
- Minimize user frustration by integrating service options into a single interface.

1.5 Problem Statement

Existing booking platforms are limited to specific services and often require users to juggle multiple apps. Bookify addresses this by offering a unified system, allowing users to manage all bookings efficiently. This solution reduces errors, saves time, and provides value-added features for convenience.

2. Overall Description

2.1 Product Perspective

Bookify is a standalone platform that bridges the gap between multiple service categories, providing an all-in-one booking system. It integrates with existing service providers to streamline bookings and user interactions, with a responsive design for mobile and desktop compatibility.

2.2 Product Functions

- User registration and login.
- Service browsing and booking.
- Payment processing with multiple gateways.
- Notifications for service availability and reminders.
- Analytics for service providers.
- Customer support and targeted offers.

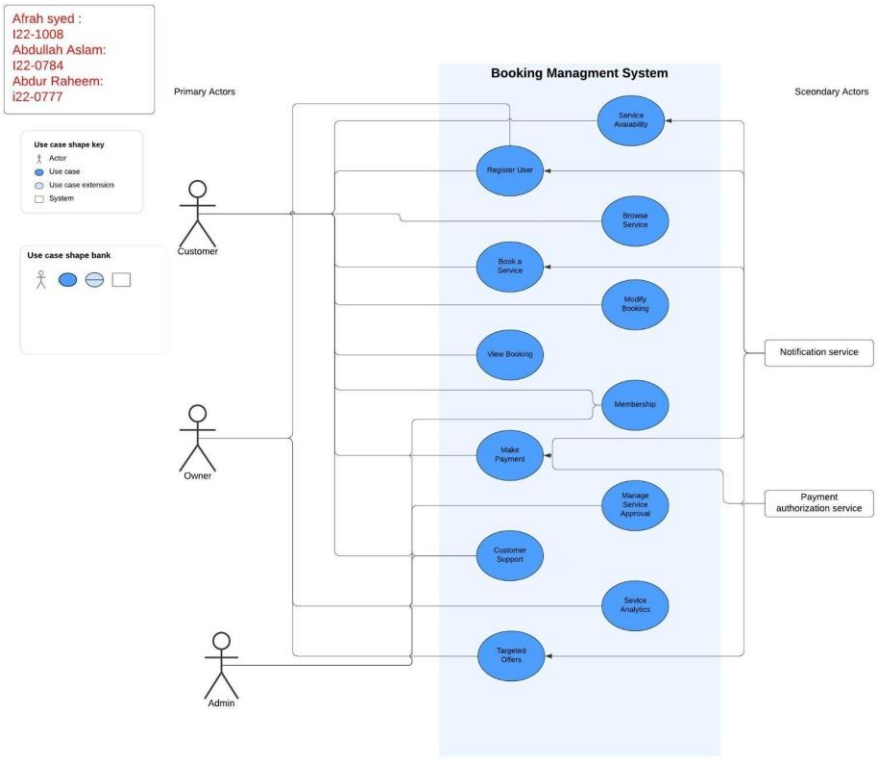
2.3 List of Use Cases

- View Booking
- Make Payment
- Customer Support
- Targeted Offers
- Service Availability Notifications
- View Service Analytics
- Manage Service Approvals
- Register User
- Browse Services
- Book a Service
- Modify Booking

2.4 Extended Use Cases

Detailed scenarios for each use case, including primary and alternate workflows, are provided in the use case document.

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

- The system should handle multiple users simultaneously without performance issues.
- Booking confirmations must be processed and displayed quickly.
- Notifications should be delivered promptly after an event occurs.
- Data retrieval, such as fetching bookings or service details, should be efficient and fast.
- The system should maintain optimal performance as the database grows in size.
- Login and authentication processes should ensure a seamless user experience.

3.2 Safety Requirements

- Safeguard data to prevent accidental loss or corruption, with **daily backups** and recovery mechanisms in place.
- Prevent double-booking by implementing real-time availability checks before confirming a booking.
- User actions that involve irreversible changes, such as cancellation, must prompt for confirmation.

3.3 Security Requirements

- Role-based access control to ensure users only access functionalities appropriate to their role (customer, provider, or admin).

- Limit database access to authorized backend services.
- Use a trusted payment gateway for processing transactions safely.

3.4 Software Quality Attributes

• Maintainability: Modular and scalable design for easy updates.
• Usability: Responsive UI optimized for all devices.
• Reliability: 99.9% uptime for critical booking operations.

3.5 Business Rules

• Customers must register before booking.
• Service providers must be approved by the admin to list services.

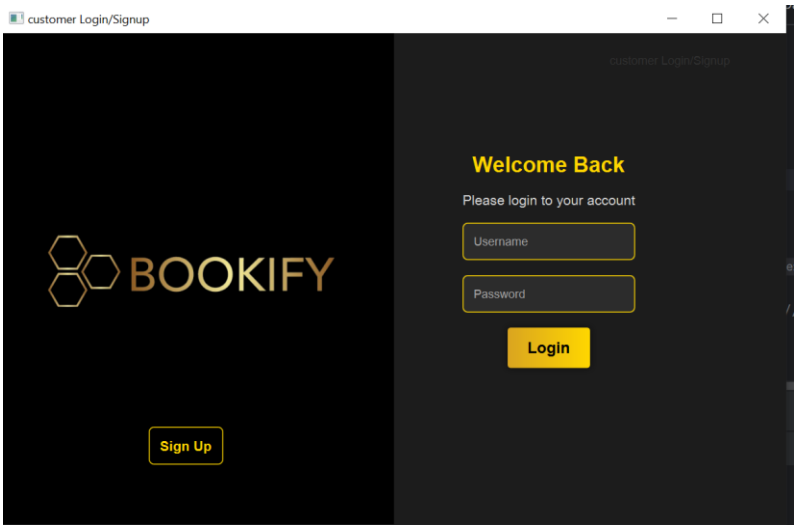
3.6 Operating Environment

• Compatible with standard desktops and laptops with at least **4GB RAM**, **2 GHz dual-core processor**, and **500MB free disk space**.
• Runs on **Windows 10 or later** and **macOS 10.15 or later**.
• Uses **JavaFX** for the user interface and **Java** for backend processing.
• Stores and manages data with **MySQL 8.0**.
• Extendable to other databases through a factory pattern.
• Requires **Java Runtime Environment (JRE)** and MySQL Client for smooth operation.

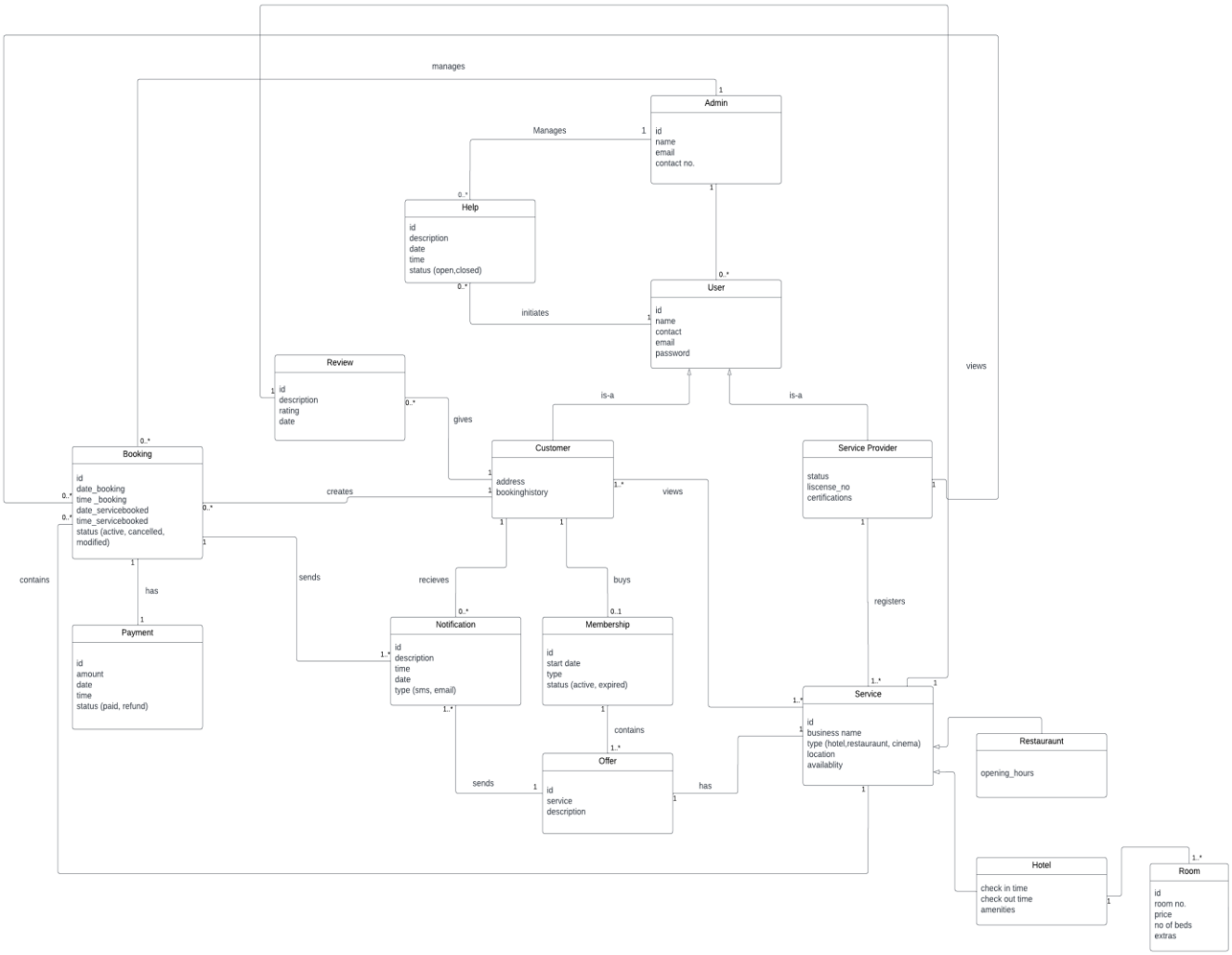
3.7 User Interfaces

- Clear menu with options for home, services, bookings, and support.
- Easy-to-use forms for login, registration, and booking.

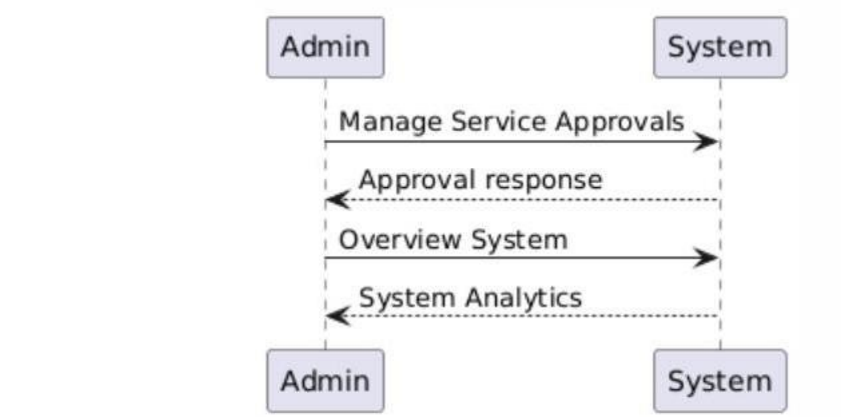
- The interface includes a dashboard for customers and service providers, with clearly labeled sections for browsing, bookings, and analytics.

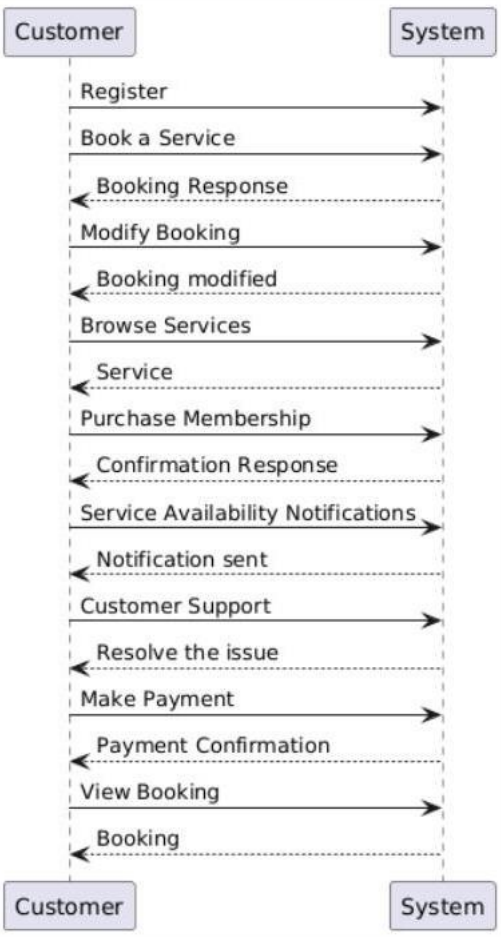


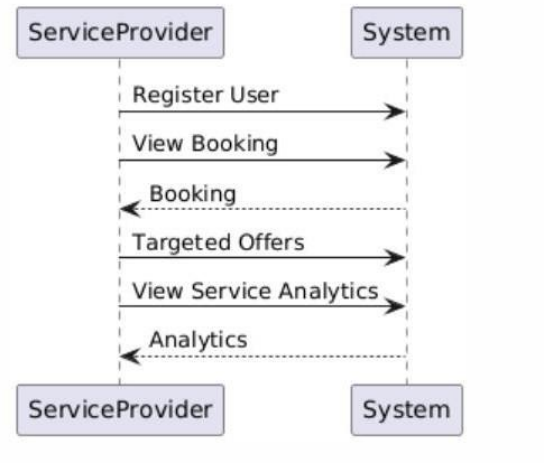
4. Domain Model



5. System Sequence Diagram

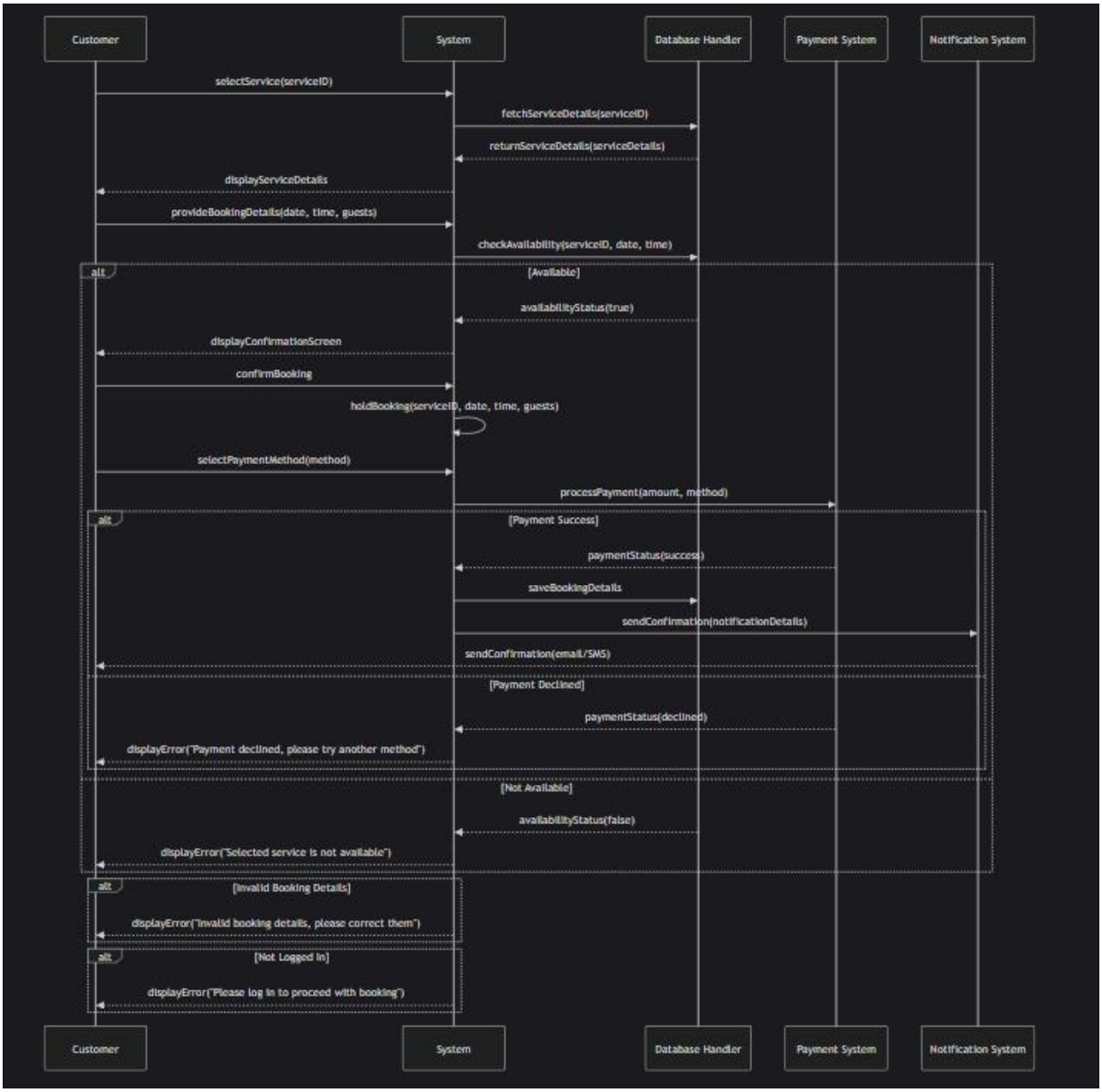




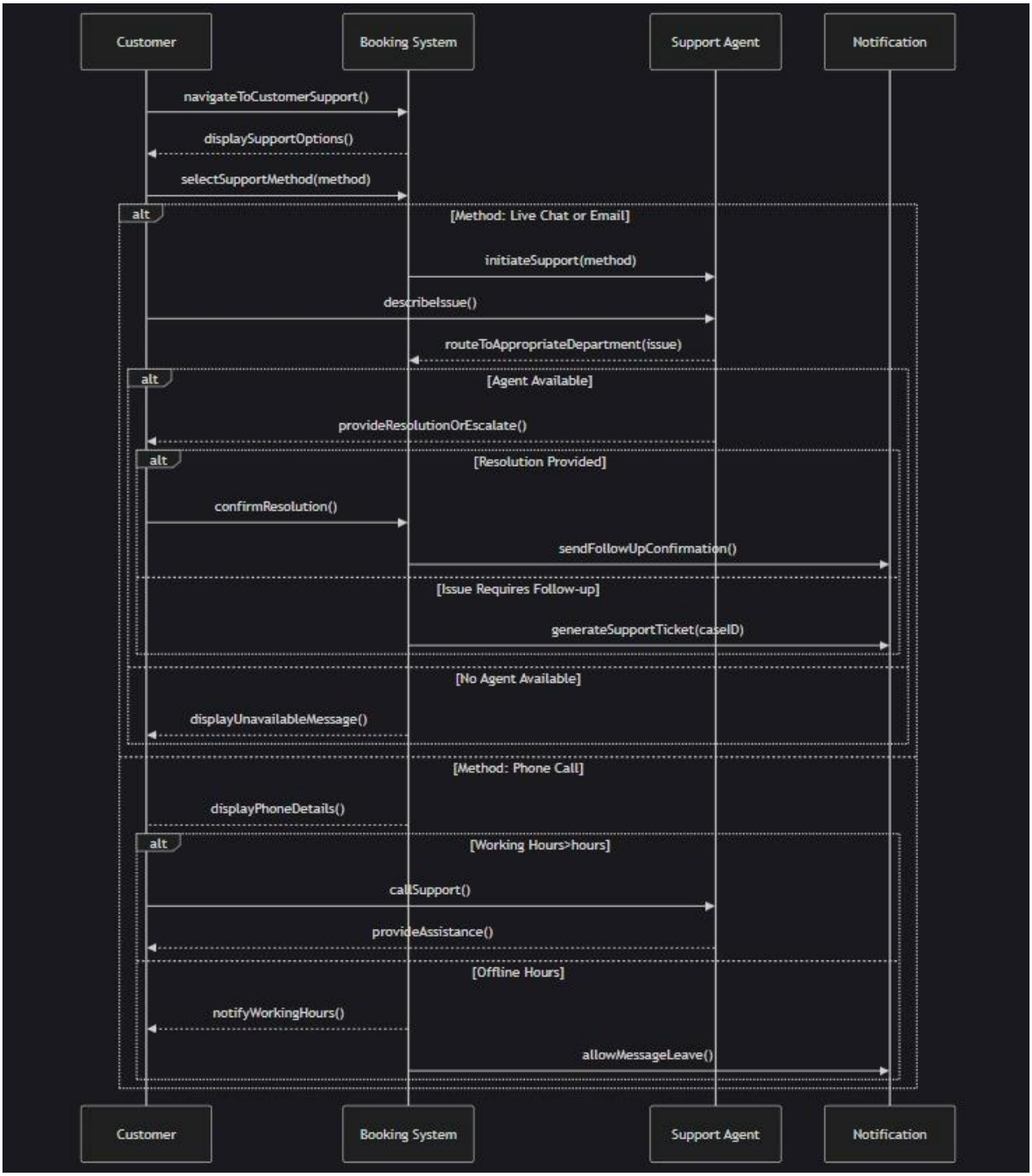


6. Sequence Diagram

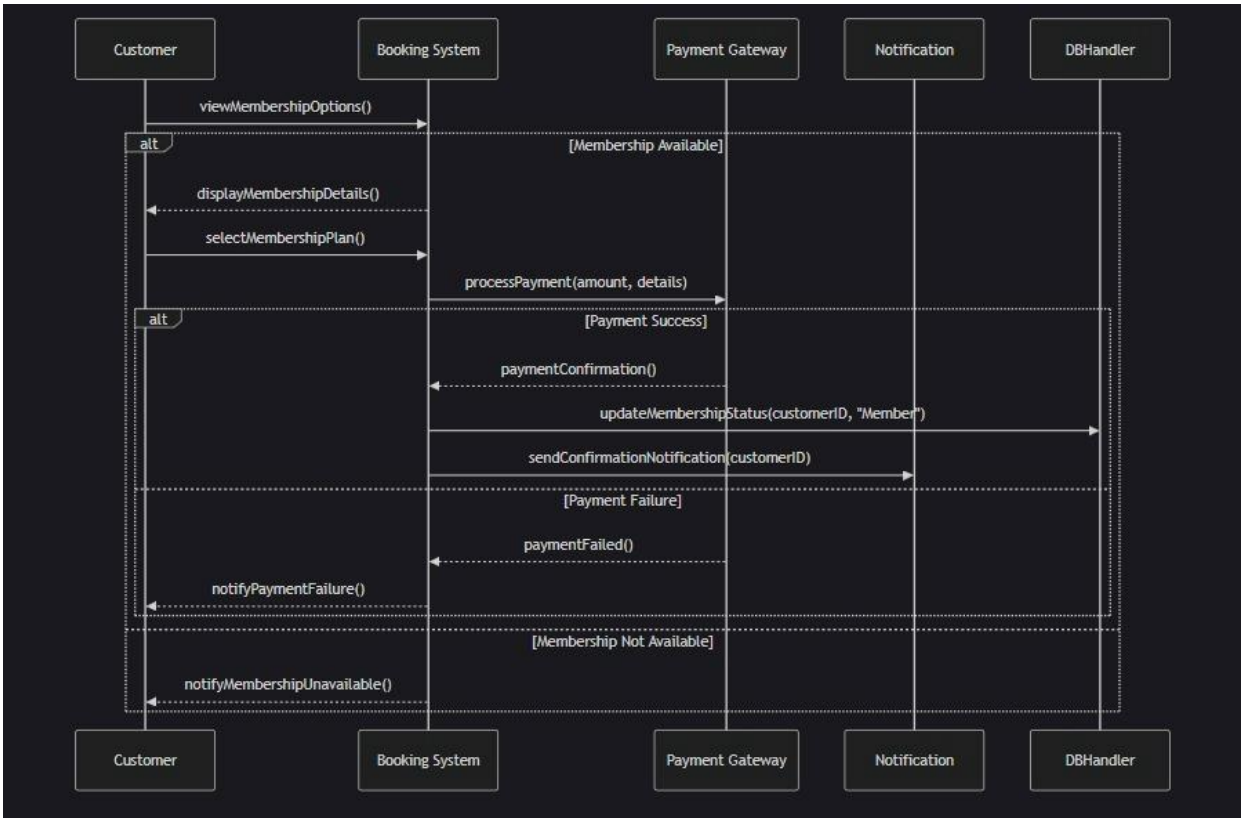
1) BOOK A SERVICE



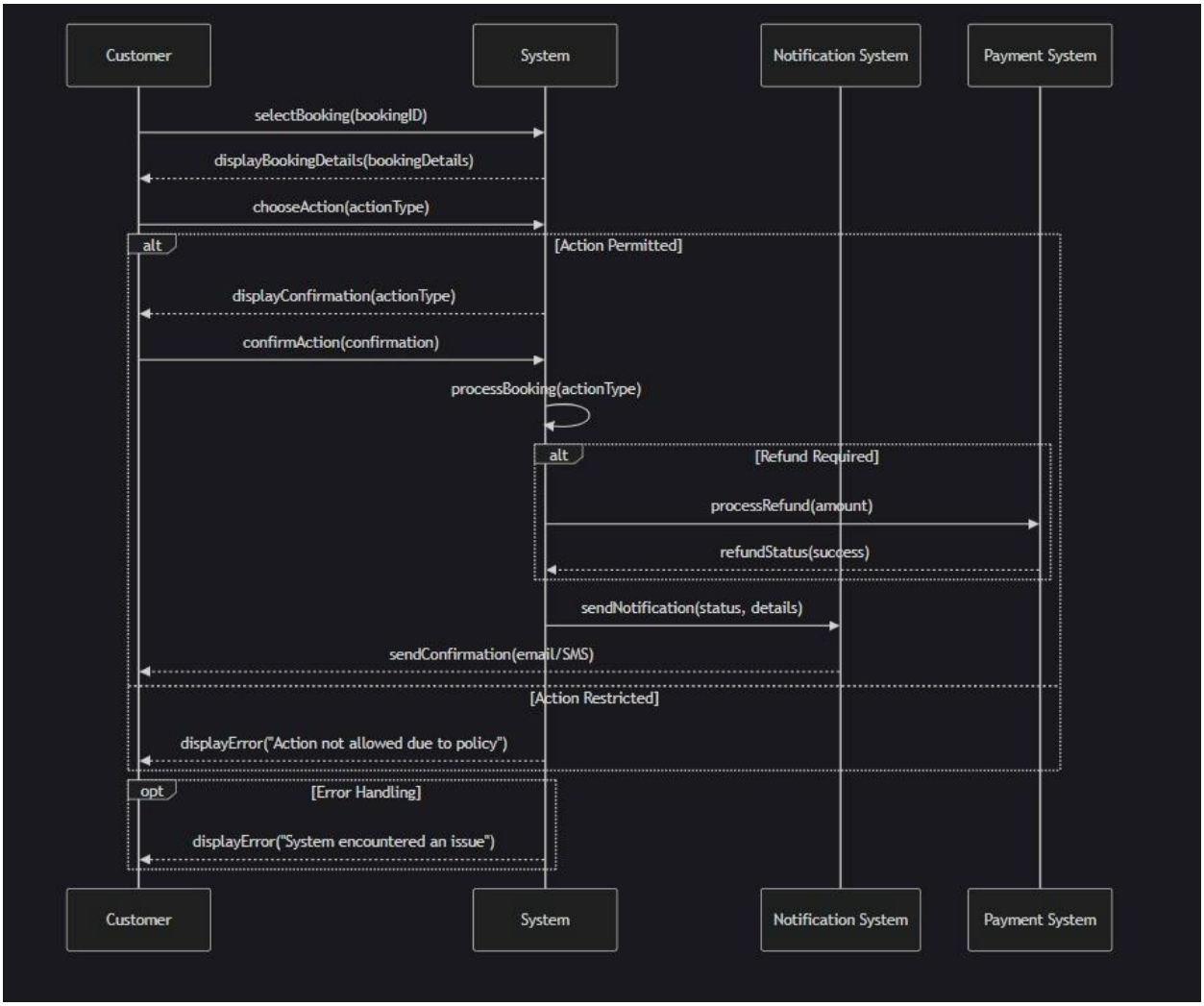
2) CUSTOMER SUPPORT



3) MEMBERSHIP

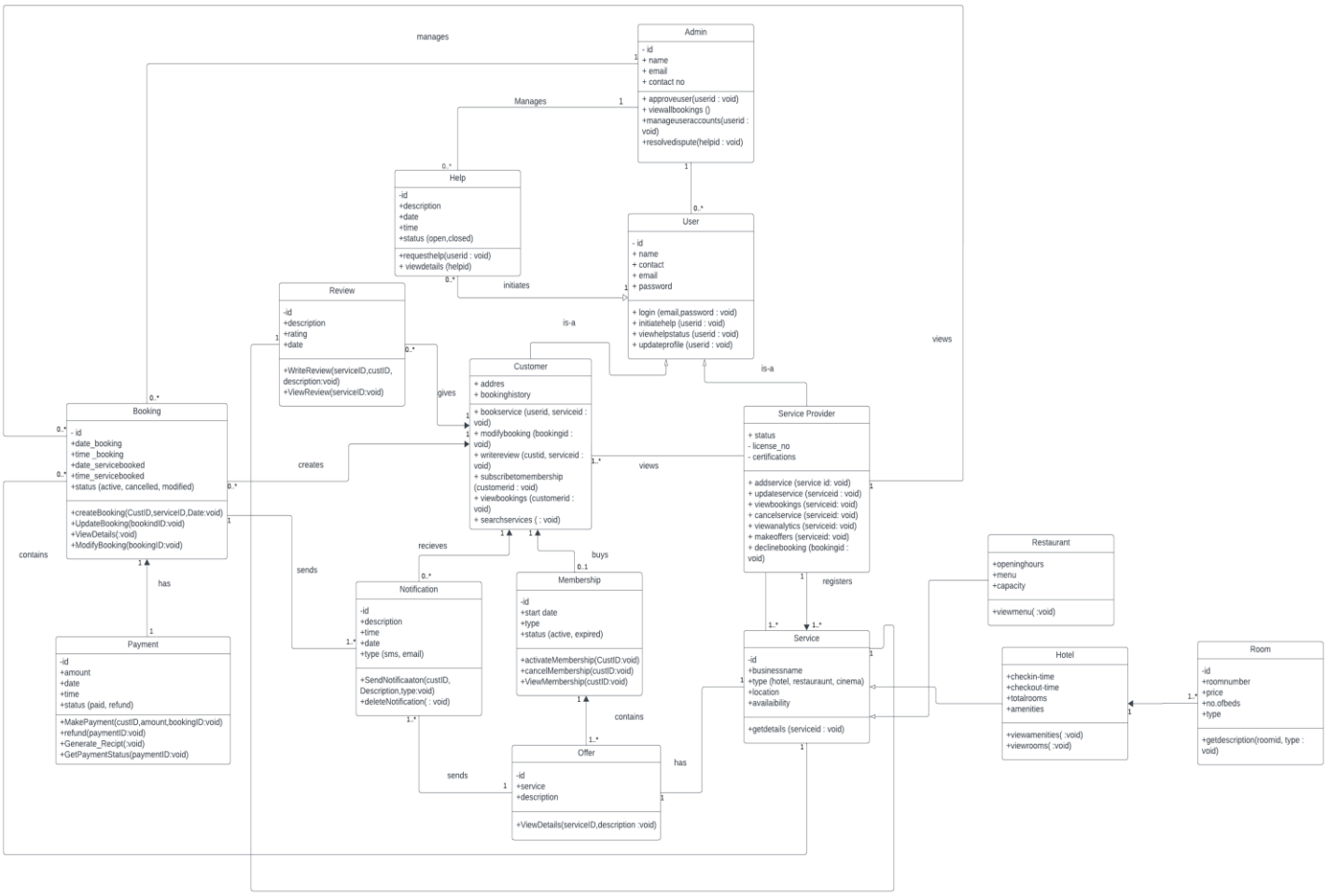


4) MODIFY BOOKING

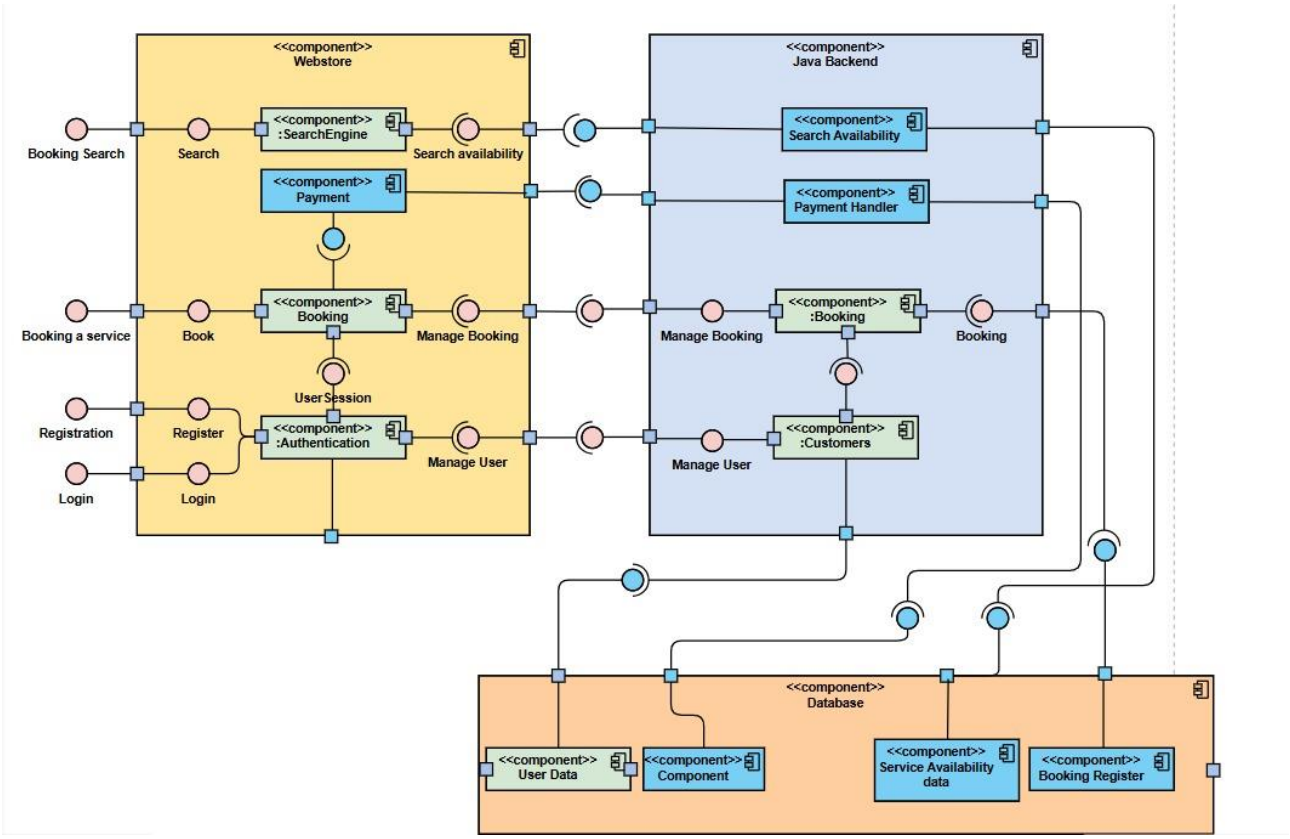


5) MAKE PAYMENT

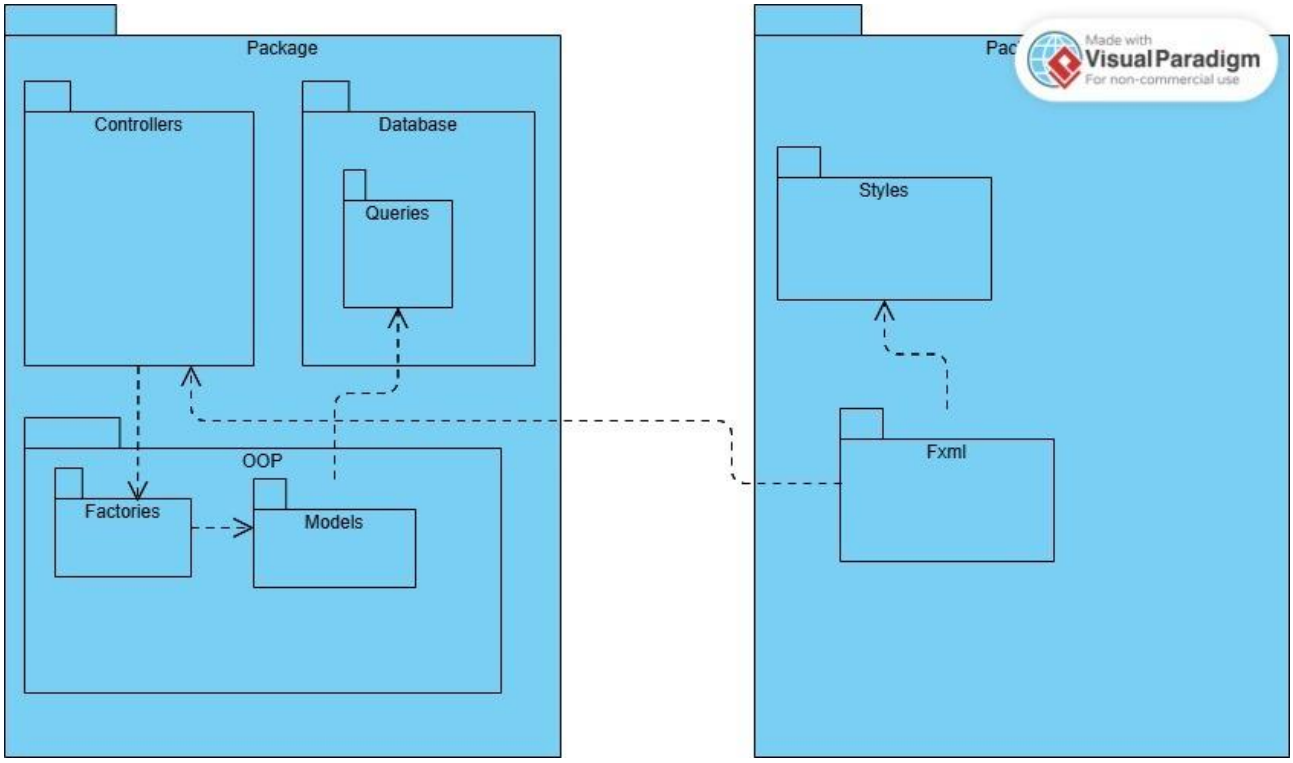
7. Class Diagram



8. Component Diagram


<<component>>
Webstore
<<component>>
Java Backend
Booking Search
Search
<<component>>
:SearchEngine
Search availability
<<component>>
Search Availability
<<component>>
Payment
<<component>>
Payment Handler
Booking a service
Book
<<component>>
Booking
Manage Booking
Manage Booking
<<component>>
:Booking
Booking
UserSession
Registration
Register
<<component>>
:Authentication
Manage User
Manage User
<<component>>
:Customers
Login
Login
<<component>>
Database
<<component>>
User Data
<<component>>
Component
<<component>>
Service Availability
data
<<component>>
Booking Register

9. Package Diagram



10. Deployment Diagram