**Software Requirements and Design Document**

**for**

Hotelify

**Prepared by**

**Ibraheem Farrukh**

**Aoun Jee**

**Sohaib Sattar**

**CodeSmith**

**27/11/2024**

**Table of Contents**

**Table of Contents**

**1.** **Introduction**

1.1 Purpose

1.2 Product Scope

1.3 Title

1.4 Objectives

1.5 Problem Statement

**2.** **Overall Description**

2.1 Product Perspective

2.2 Product Functions

2.3 List of Use Cases

2.4 Extended Use Cases

2.5 Use Case Diagram

**3.** **Other Nonfunctional Requirements**

3.1 Performance Requirements

3.2 Safety Requirements

3.3 Security Requirements

3.4 Software Quality Attributes

3.5 Business Rules

3.6 Operating Environment

3.7 User Interfaces

**4.** **Domain Model**

**5.** **System Sequence Diagram**

**6.** **Sequence Diagram**

**7.** **Class Diagram**

**8. Component Diagram**

**9.** **Package Diagram**

**10.** **Deployment Diagram**

# **Introduction**

## **Purpose**

This document defines the requirements for Hotelify, a hotel management system designed to ease the booking process of hotels. It is tailor fitted for the layperson to use to their needs whether it be room booking, restaurant booking or event hall booking, Hotelify can do it all! This document covers all the major aspects of the application from its components to its uses to its working.

## **Product Scope**

Hotelify aims to ease the problem that is managing hotels. With this application whether you are a Guest, an Employee or an Admin you can access your needs easily. It is designed to streamline productivity and create ease of use for whomever is using it. Its benefits include a single centralized application to access all the important parts of a hotel and a simple and easy to use interface.

## **Title**

Hotelify: Enhancing Guest Experiences and Streamlining Hotel Operations through an Integrated Management System

## **Objectives**

The primary objectives of this project are :

1. Improve the efficiency of hotel staff by automating routine tasks such as room management, service request handling, and bill generation.

2. Enable staff to track room status, housekeeping tasks, and maintenance requests in real-time.

3. Create a centralized system to unify all hotel operations, including room reservations, restaurant and hall bookings, service management, and billing.

4. Provide a seamless and convenient platform for hotel guests to manage their bookings, service requests, and feedback.

## **Problem Statement**

The hospitality industry often struggles with disorganized hotel services, leading to poor guest experiences and inefficient operations. Hotels face challenges in managing room bookings, service requests, restaurant reservations, and event scheduling due to fragmented systems that fail to communicate with each other. This results in delayed services, overbooked rooms, and increased operational costs, all of which negatively impact guest satisfaction. Additionally, hotel staff members waste valuable time dealing with manual tasks and coordinating multiple systems, hindering their ability to provide quick and efficient service.

The Hotelify system is designed to address these inefficiencies by providing an integrated management platform that unifies all hotel services. The system offers real-time updates on room availability, streamlines service requests, and simplifies reservation management for rooms, restaurants, and event halls. This ensures that the hotel can offer a seamless guest experience, reduce wait times, and improve overall operational efficiency. Furthermore, by automating repetitive tasks, Hotelify mitigates the need for manual processes, enabling staff to focus on higher-value tasks, ultimately enhancing guest satisfaction and boosting hotel productivity.

# **Overall Description**

## **Product Perspective**

Hotelify is a new, self-contained product developed to address inefficiencies and challenges in the hospitality industry. It is designed from the ground up as an integrated hotel management system that unifies various hotel operations, such as room booking, service requests, restaurant and event reservations, and billing, into a single platform. The system aims to provide both hotel staff and guests with an intuitive and seamless experience while improving operational efficiency and customer satisfaction.

Unlike existing hotel management systems, which often rely on multiple fragmented solutions, Hotelify provides an all-in-one solution, eliminating the need for various disconnected systems. It is not a replacement for any existing system but instead an innovative approach to solving common problems in hotel management, including double bookings, delayed services, and inefficient communication between hotel departments.

## **Product Functions**

The Hotelify system must perform several key functions to meet the needs of both guests and hotel staff. These functions are essential for managing reservations, billing, and service requests in a seamless and efficient manner. Below is a high-level summary of the major functions the product must perform:

1. Book a Room:
   * Allow guests to select and book available rooms based on their preferences (e.g., room type, check-in/check-out dates).
   * Confirm booking and generate a unique reservation ID.
   * Update room availability in real-time.
2. Add Guest:
   * Allow hotel staff to add guest information into the system (e.g., name, contact details, email).
   * Manage guest profiles for future reference (e.g., booking history, special requests).
3. Pay Bill:
   * Generate a bill for guests at check-out based on their room, services, and any applicable discounts.
   * Allow guests to view their bill and make payments via various methods (e.g., credit card, cash).
   * Update payment status and finalize the transaction.
4. Book Restaurant Table:
   * Allow guests to reserve a table at the restaurant based on available time slots and table capacity.
   * Confirm reservation and update the table's availability.
   * Send reservation details (time, table number) to the guest and restaurant staff.
5. Book Hall:
   * Allow guests to reserve event halls for meetings, conferences, or special events.
   * Provide available time slots for booking and manage the hall's capacity.
   * Confirm reservations and send event details to the guest and hotel staff.

## **List of Use Cases**

1. Book a Room

2. Modify Existing Bookings

3. Request Room Service

4. Make Restaurant Reservation

5. Generate Room Bill

6. View and Modify Guest Information

7. Manage Guest Requests

8. Check Room Status

9. Handle Room Maintenance Requests

10. Assign Room to Guest

11. View Bill and Make Payment

12. Schedule Event Hall

## **Extended Use Cases**

# 1. Book a Room

**Use Case Name** : Book a Room

**Scope :** Hotelify System

**Level** : User goal

**Primary Actor** : Guest

**Stakeholders & Interests :**

**Guest**: Wants to book a room quickly and with preferred amenities.

**Hotel Staff**: Need accurate information to prepare the room.

**Payment Gateway**: Requires successful payment transaction.

**Preconditions :**

Guest must be registered/logged in. Rooms must be available for selected dates.

**Postconditions :**

Booking confirmation is sent to the guest. Room is reserved in the system.

**Main Success Scenario**

**Actor Action**  **System Responsibility**

| 1. Guest logs into the system. |  |
| --- | --- |
| 2. Guest selects dates and room type. |  |
|  | 3. System shows available rooms. |
| 4. Guest selects a room and proceeds to payment. |  |
|  | 5. Payment is processed and confirmed. |
|  | 6. System sends booking confirmation to the guest. |

**Extensions :**

**5a**: If payment fails, guest is prompted to retry or use another payment method.

**6a**: If no rooms are available, system suggests alternate dates or room types.

# 2. Modify Existing Bookings

**Use Case Name :** Modify Existing Room Booking

**Scope :** Hotelify System

**Level :** User goal

**Primary Actor :** Guest

**Stakeholders & Interests :**

**Guest**: Needs to modify the booking (change dates, room type).

**Hotel Staff**: Must be informed about modifications to prepare room or adjust scheduling.

**Preconditions :**

Guest must have an existing booking.

**Postconditions :**

Booking is modified, and confirmation is sent.

**Main Success Scenario**

**Actor Action**   **System Responsibility**

| 1. Guest logs into the system. |  |
| --- | --- |
| 2. Guest selects "Manage Existing Booking". |  |
| 3. Guest modifies the booking (e.g., cancel booking, change dates, room type). |  |
|  | 4. System checks availability. |
|  | 5. System updates the booking and confirms changes. |
|  | 6. System sends updated booking confirmation to guest. |

**Extensions :**

**4a**: If the requested room type or dates are unavailable, the system suggests alternatives.

**5a**: If guest fails to confirm modifications, system reverts to original booking.

# 3. Request Room Service

**Use Case Name :** Request Room Service

**Scope :** Hotelify System

**Level :** User goal

**Primary Actor :** Guest

**Stakeholders & Interests :**

**Guest**: Wants to request room services (e.g., food, cleaning).

**Hotel Staff**: Needs to receive and fulfill service requests efficiently.

**Preconditions :**

Guest must be checked into the hotel.

**Postconditions :**

Service request is confirmed and assigned to hotel staff for fulfillment.

**Main Success Scenario**

**Actor Action**   **System Responsibility**

| 1. Guest logs into the system. |  |
| --- | --- |
| 2. Guest selects "Request Room Service". |  |
| 3. Guest selects desired service (e.g., food, laundry, cleaning). |  |
|  | 4. System confirms the request and notifies hotel staff. |
|  | 5. Request is acknowledged and fulfills the request. |
| 6. Guest is notified when the request is fulfilled. |  |

**Extensions :**

**4a**: If the service is unavailable (e.g., item out of stock), the system informs the guest and offers alternatives.

**5a**: If there is a delay in fulfilling the request, the system notifies the guest of the new estimated time.

# 4. Make Restaurant Reservation

**Use Case Name :** Make Restaurant Reservation

**Scope :** Hotelify System

**Level :** User goal

**Primary Actor :** Guest

**Stakeholders & Interests :**

● **Guest**: Wants to reserve a table at the hotel restaurant for a specific time.

● **Restaurant Staff**: Needs to know reservation details to prepare for the guest’s arrival.

● **Hotel Management**: Interested in maximizing restaurant occupancy and ensuring good guest experience.

**Preconditions :**

The guest must be checked in or have a booking with the hotel. The restaurant must have available tables for the selected time.

**Postconditions :**

The restaurant reservation is confirmed and stored in the system. Restaurant staff is notified.

**Main Success Scenario**

**Actor Action System Responsibility**

| 1. Guest logs into the system and navigates to restaurant reservations. |  |
| --- | --- |
|  | 2. System displays available restaurant slots and tables. |
| 3. Guest selects the date, time and number of people. |  |
|  | 4. System verifies availability of the selected time and options. |
| 5. Guest confirms the reservation details. |  |
|  | 6. System confirms the reservation and notifies reservation staff |
| 7. Guest receives a confirmation of the restaurant reservation. |  |

**Extensions :**

2a. If no tables are available, the system suggests alternate times or dates.

3a. If the guest cancels the process, no reservation is made, and the system returns to the main menu.

# 5. Generate Room Bill

**Use Case Name :** Generate Room Bill

**Scope :** Hotelify System

**Level :** User goal

**Primary Actor :** Hotel Staff

**Stakeholders & Interests**

● **Guest**: Wants an accurate final bill for services rendered during their stay.

● **Hotel Staff**: Needs to ensure all charges (room, food, services) are correct and accounted for.

● **Payment Gateway**: Processes the final bill payment.

**Preconditions :**

The guest must have used hotel services, and check-out should be initiated or nearing completion.

**Postconditions :**

The bill is generated, reviewed, and ready for payment.

**Main Success Scenario**

**Actor Action** **System Responsibility**

| 1. Hotel staff initiates the bill generation process. |  |
| --- | --- |
|  | 2. System compiles all charges (room, services, dining) associated with the guest's stay. |
| 3. Hotel staff reviews and confirms the charges. |  |
|  | 4. System generates the final bill and displays it for the guest to review. |
| 5. Guest reviews the bill and approves it for payment. |  |
|  | 6. System marks the bill as ready for payment and updates the guest profile. |

**Extensions :**

2a. If there are discrepancies in charges, staff can correct the bill before finalizing it.

4a. If the guest disputes the charges, the system allows for adjustments before approval.

# 6. View and Modify Guest Information

**Use Case Name :** View and Modify Guest Information

**Scope :** Hotelify System

**Level :** User goal

**Primary Actor :** Hotel Staff

**Stakeholders & Interests :**

● **Guest**: Wants their personal information to be accurate and up to date.

● **Hotel Staff**: Needs current guest data to provide personalized service and ensure smooth communication.

**Preconditions :**

The guest must have a registered profile in the system.

Hotel staff must have access and permission to modify guest information.

**Postconditions :**

Guest information is updated in the system.

**Main Success Scenario**

**Actor Action**   **System Responsibility**

| 1. Hotel staff selects the guest profile from the system. |  |
| --- | --- |
|  | 2. System retrieves the guest’s current information (e.g, name, contact, preferences) |
| 3. Hotel staff views the guest information. |  |
| 4. Hotel staff updates guest details if needed (e.g, phone number, preferences). |  |
|  | 5. System validates the input and updates the information. |
|  | 6. System saves the changes and confirms that the guest’s information has been modified. |

**Extensions :**

3a. If the system detects invalid data, it prompts the staff to correct the input.

4a. If the staff cancels the operation, no changes are saved, and the system retains the original information.

# 7. Manage Guest Requests

• **Scope**: Hotel Management System

• **Level**: User goal

• **Primary Actor**: Hotelify System

• **Stakeholders and Interests**:

o **Guest**: Expects timely fulfillment of requests.

o **Hotel**: Needs to track and manage guest requests efficiently.

• **Preconditions**: o Guest must be checked in.

• **Postconditions**: o Request is fulfilled or denied with reason.

**Main Success Scenario**:

| **User Side** | **System Side** |
| --- | --- |
| 1. Staff views guest requests. | 2. System displays all current requests. |
| 3. Staff assigns request to a department. | 4. System logs the assignment and notifies the relevant department. |
| 5. Staff fulfills or denies the request. | 6. System updates request status to fulfilled or denied. |
| 7. Staff notifies guest of request status. | 8. System sends notification to the guest. |

**Extensions**:

• 5a. Request cannot be fulfilled; system suggests alternative options or denies the request with a reason.

# 8. Check Room Status

• **Scope**: Hotelify System

• **Level**: User goal

• **Primary Actor**: Housekeeping Staff

• **Stakeholders and Interests**:

o **Hotel**: Needs to monitor room readiness for guests.

• **Preconditions**: o Staff must have access to the housekeeping system.

• **Postconditions**: o Room status is updated in the system.

**Main Success Scenario**:

| **User Side** | **System Side** |
| --- | --- |
| 1. Housekeeping staff logs into the system. | 2. System displays room status information. |
| 3. Staff views room occupancy and cleaning status. | 4. System updates room status based on the progress. |
| **User Side** | **System Side** |
| 5. Staff marks rooms as cleaned or reports an issue. | 6. System updates the room status accordingly. |

**Extensions**:

• 5a. Staff reports an issue with the room, system flags it for maintenance.

# 9. Handle Room Maintenance Requests

• **Scope**: Hotelify System

• **Level**: User goal

• **Primary Actor**: Hotel Staff (Maintenance)

• **Stakeholders and Interests**:

o **Guest**: Wants the room issues to be addressed promptly.

o **Hotel**: Needs to ensure rooms are well-maintained for guest satisfaction.

• **Preconditions**: o Guest or staff has reported a maintenance issue.

• **Postconditions**: o The maintenance issue is resolved, or an alternative is provided.

**Main Success Scenario**:

| **User Side** | **System Side** |
| --- | --- |
| 1. Guest or staff reports a maintenance issue. | 2. System logs the request and assigns it to maintenance staff. |
| 3. Maintenance staff reviews the request. | 4. System displays the details of the issue. |
| 5. Maintenance staff resolves the issue. | 6. System updates room status and request completion. |
| 7. Staff notifies guest of the resolution. | 8. System sends notification to guest or staff about the completion. |

**Extensions**:

• 5a. Issue cannot be resolved immediately; alternative arrangements are made for the guest.

# 10. Assign Room to Guest

• **Scope**: Hotelify System

• **Level**: User goal

• **Primary Actor**: Front Desk Staff

• **Stakeholders and Interests**:

o **Guest**: Wants to have a room assigned quickly.

o **Hotel**: Needs to ensure guests are assigned rooms efficiently.

• **Preconditions**:

o Guest has a valid booking.

o Rooms must be available for assignment.

• **Postconditions**:

o Room is assigned to the guest.

o Room status is updated to "occupied." **Main Success Scenario**:

| **User Side** | **System Side** |
| --- | --- |
| 1. Front desk staff accesses the guest's booking. | 2. System retrieves the booking details. |
| 3. Staff selects a room for the guest. | 4. System shows available rooms for assignment. |
| 5. Staff assigns the selected room to the guest. | 6. System updates the room status to  "occupied." |
| 7. Staff provides the room key to the guest. | 8. System logs the check-in details for the guest. |

**Extensions**:

• 5a. No suitable rooms are available; guest is offered an upgrade or alternative.

# 11. View Bill and Make Payment

• **Scope**: Hotelify System

• **Level**: User goal

• **Primary Actor**: Guest

• **Stakeholders and Interests**:

o **Guest**: Wants to review the bill and make the final payment.

o **Hotel**: Wants to ensure payments are processed smoothly.

• **Preconditions**:

o Guest must be checked in.

o Guest must have charges for their stay.

• **Postconditions**: o Bill is reviewed and payment is completed.

**Main Success Scenario**:

| **User Side** | **System Side** |
| --- | --- |
| 1. Guest logs into the system. | 2. System retrieves guest account details. |
| 3. Guest views the detailed bill. | 4. System displays the guest’s bill. |
| 5. Guest selects a payment method. | 6. System processes the payment. |
| 7. Guest confirms the payment. | 8. System confirms the payment and updates guest account. |
| 9. Guest receives payment confirmation. | 10. System sends a confirmation to the guest. |

**Extensions**:

• 5a. Payment method is declined; guest selects an alternative payment method.

# 12. Schedule Event Hall

• **Use Case Name**: Schedule Event Hall

• **Scope**: Hotelify System

• **Level**: User goal

• **Primary Actor**: Event Coordinator

• **Stakeholders & Interests**:

o **Event Organizer**: Wants to book a conference room and ensure availability.

o **Hotel**: Aims to manage event space efficiently and avoid scheduling conflicts.

• **Preconditions**:

o Conference room availability must be up-to-date.

o Event coordinator has access to the booking system.

• **Postconditions**: o Conference room is scheduled for the event, and other staff are notified.

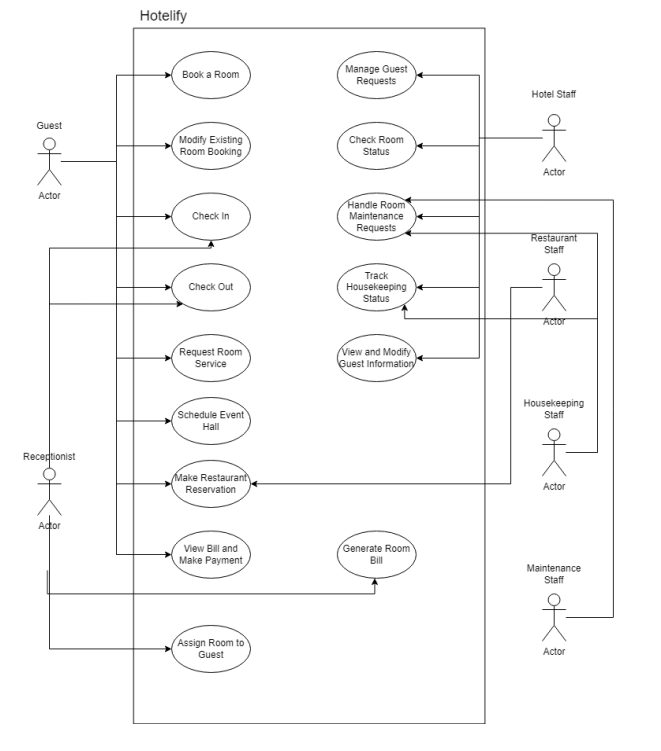
**Main Success Scenario**:

| **Actor Action** | **System Responsibility** |
| --- | --- |
| 1. Event coordinator requests to book a conference room. | 2. System displays available rooms and timeslots. |
| 3. Staff selects a room and sets the event date/time. | 4. System checks for conflicts and reserves the room for the event. |
| 5. Staff confirms the booking. | 6. System updates the room schedule and sends confirmation to the organizer. |
| 7. Staff shares event details with relevant hotel departments. | 8. System notifies housekeeping, catering, and other departments about the upcoming event. |

**Extensions**:

• 4a. If there is a scheduling conflict, the system suggests alternative rooms or time slots.

## **Use Case Diagram**



# **Other Nonfunctional Requirements**

#### 3.1 Performance Requirements

Application Response Time:

The application should load any screen (e.g., dashboard, booking details) within 1 second. Database queries for retrieving reservation data, room availability, and guest details should execute within 500 milliseconds for datasets up to 100,000 records.

Startup Time:

The application must fully initialize and be ready for use within 5 seconds on the recommended hardware configuration.

Concurrency:

The system must support up to 50 simultaneous users working on the same database, performing actions like bookings, checking room availability, or making payments, without any noticeable performance degradation.

Improved Performance:

Caching mechanisms should be implemented to reduce frequent database calls. For example, frequently accessed data like room availability and guest history should be cached, reducing the number of database queries and improving performance. This caching should allow data retrieval in constant time (Big O(1)).

## **Safety Requirements**

**Data Backup:**

The application must ensure daily backups of the MySQL database to prevent data loss in case of unexpected hardware or software failures.

**Error Recovery:**

In case of a failure during an operation (e.g., during payment processing or booking), the system must log the error and allow users to retry the operation without any data loss. This ensures that no transaction or reservation is lost.

**Accidental Deletion Prevention:**

Critical actions, such as room deletion or service request cancellation, should prompt the user with confirmation dialogs to prevent accidental data loss or unintended changes.

## **Security Requirements**

**Authentication:**

Users must authenticate with unique credentials (username and password). The system should enforce strong password policies (minimum 8 characters, including alphanumeric and special characters).

**Data Encryption:**

Sensitive user information, such as passwords, must be stored and encrypted using secure hashing algorithms like SHA-256.

Data transferred between the application and the database should be encrypted using SSL/TLS to protect against man-in-the-middle attacks.

**Access Control:**

Role-based access control (RBAC) must be implemented to ensure that only authorized users (e.g., admins) can perform sensitive actions like modifying user data or managing **reservations.**

**Secure Local Storage:**

Any temporary data stored locally (e.g., cached user sessions or payment details) must be encrypted to prevent unauthorized access.

## **Software Quality Attributes**

**Reliability**:

The system should have an uptime of 99.8% during operational hours, ensuring that the application is available for use with minimal disruptions.

**Usability**:

The user interface must follow intuitive design principles, minimizing the need for training and ensuring that both hotel staff and guests can navigate the system easily. This will be achieved by adhering to industry-standard GUI guidelines in JavaFX.

**Maintainability**:

The system’s codebase should be modular and follow industry-standard practices. This will simplify future updates, bug fixes, and integration of new features.

**Portability**:

The application should be compatible with Windows 10/11 (64-bit) and macOS Monterey or later, allowing the software to run seamlessly across both major platforms.

**Scalability**:

Initially supporting 50 users, the system should be designed to scale with minimal changes to the architecture to accommodate up to 200 users in the future.

**Testability**:

The application must support unit and integration testing for critical components such as database operations, business logic, and user interfaces.

## **Business Rules**

**User Roles**:

Guests can only view and edit their own reservation details and submit service requests.

Employees can manage reservations, service requests, and room availability but cannot access sensitive guest information.

Admins have full access, including the ability to manage user accounts, modify system settings, and manage rooms, reservations, and payments.

**Reservation Confirmation**:

Room reservations should be confirmed immediately upon successful payment, with an email or notification sent to the guest.

**Service Requests**:

Guests should only be able to make one service request at a time unless the current request is resolved. Requests are tracked, and guests can view the status of their requests.

## **Operating Environment**

**Hardware Platform**:

* **Client System** (Running Hotelify):
  + **Minimum**:
    - 8 GB RAM
    - 2 GHz dual-core processor
    - 500 MB free storage
  + **Recommended**:
    - 16 GB RAM
    - 4-core processor
    - 1 GB free storage
* **Database Server**:
  + 16 GB RAM
  + 4-core processor
  + 500 GB SSD

**Operating System**:

* The application must be compatible with Windows 10/11 (64-bit) and macOS Monterey or later.

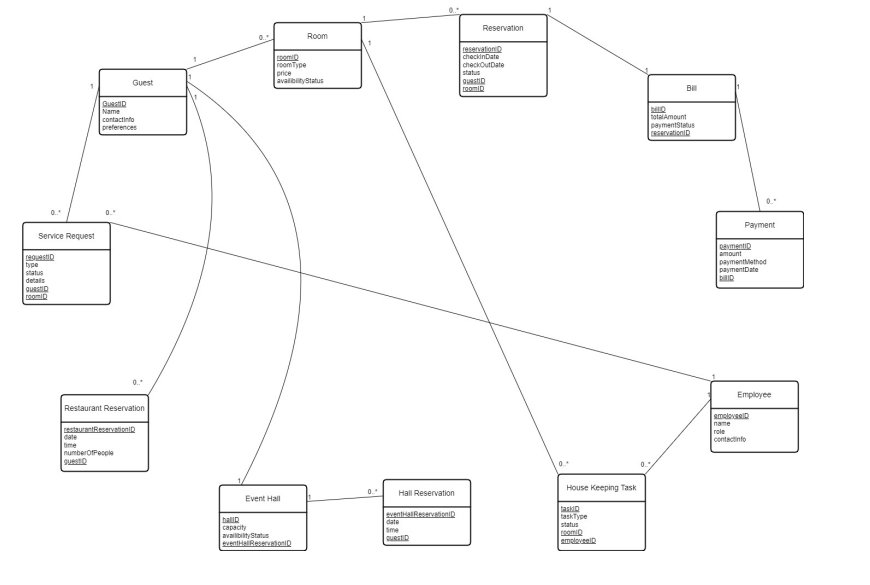
**Dependencies**:

* JavaFX SDK 17 for UI development.
* MySQL Connector for Java to facilitate database communication.
* Scene Builder for designing and modifying the user interfaces.

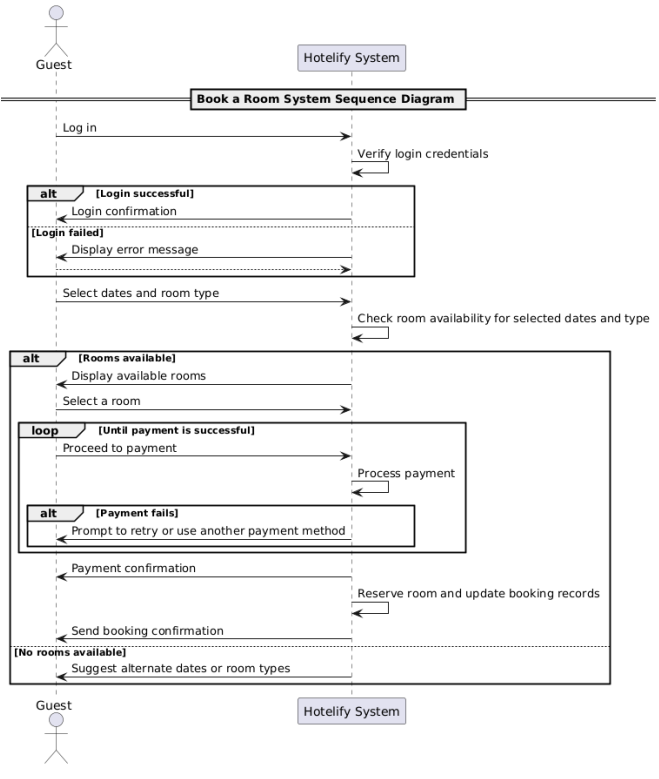
## **User Interfaces**

1. Guest Interface
2. Admin Interface
3. Employee Interface

# **Domain Model**

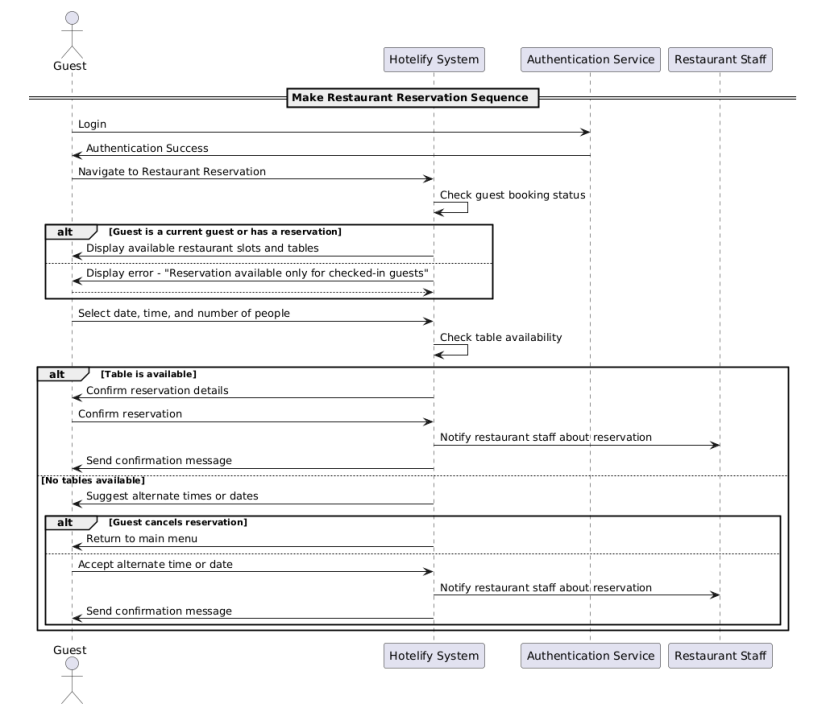


# **System Sequence Diagram**

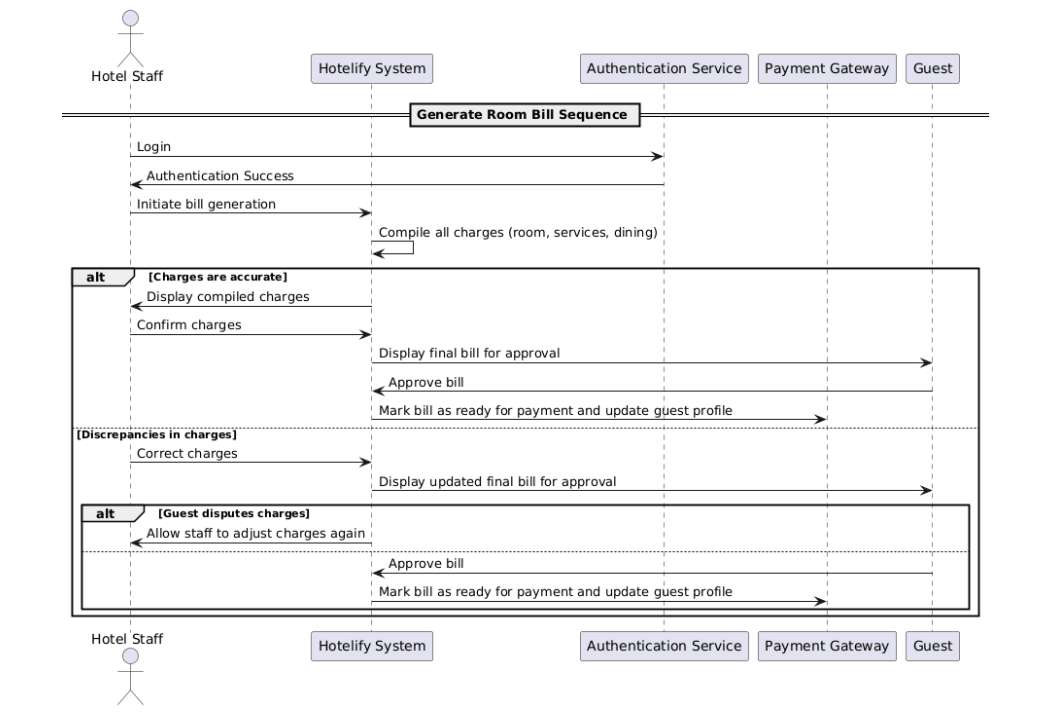


# **Sequence Diagram**

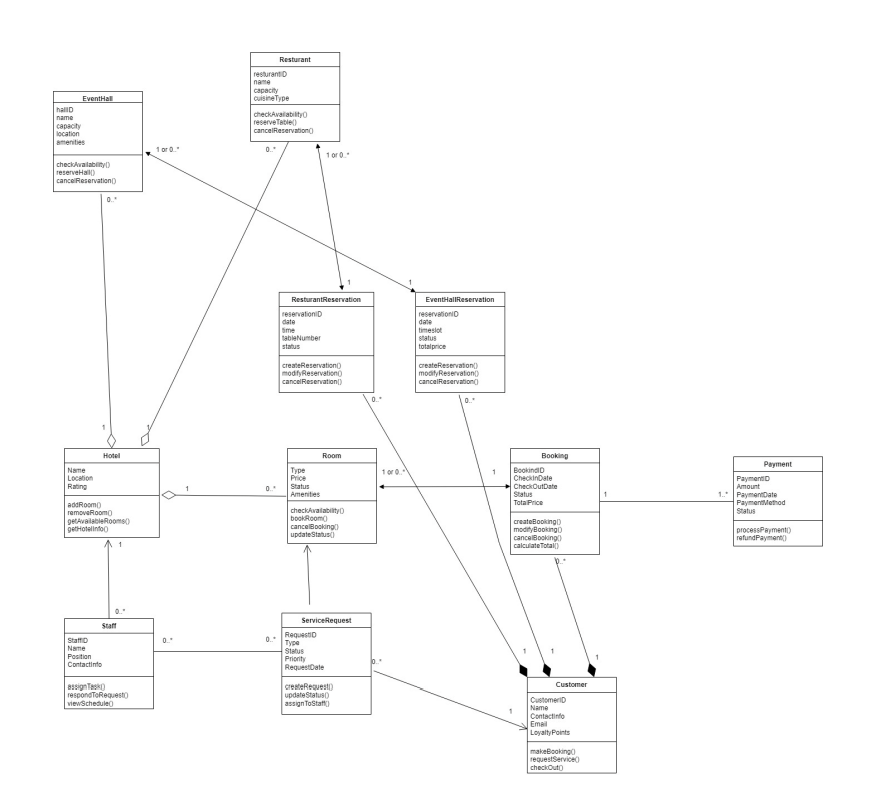
1. Make Restaurant Reservation



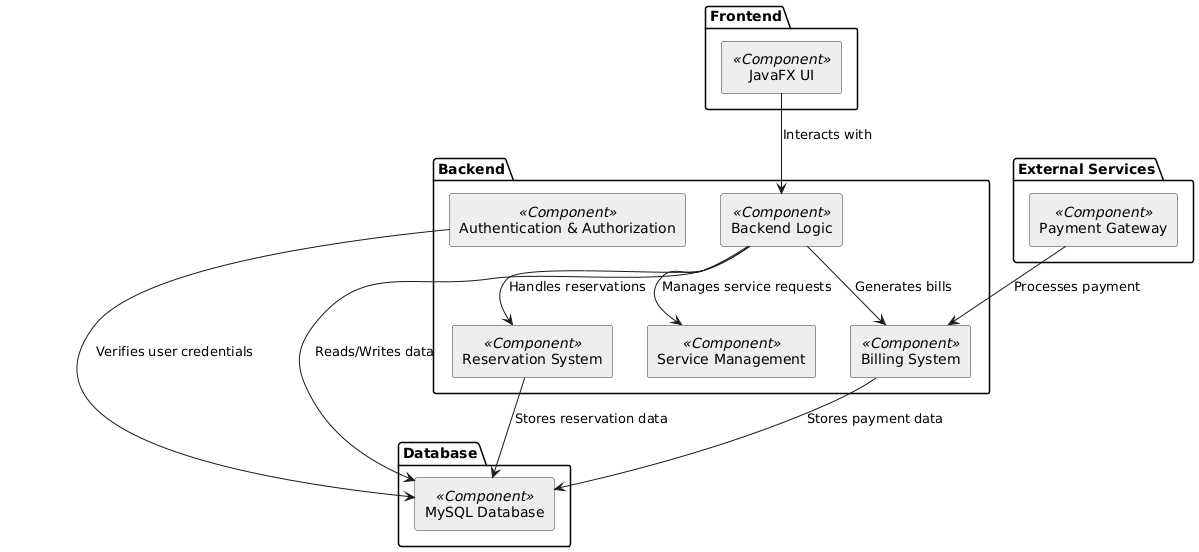
1. Generate Room Bill



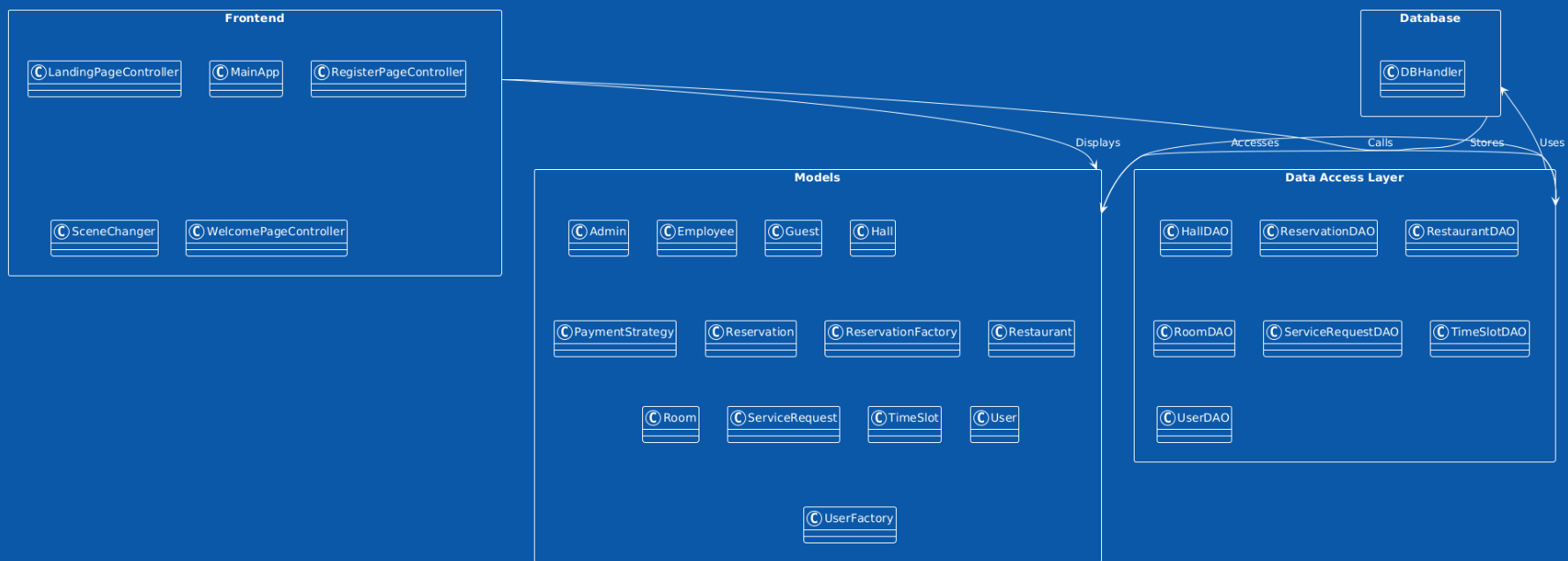
# **Class Diagram**



# **Component Diagram**



# **Package Diagram**



# **Deployment Diagram**

