HOTEL WEBSITE

Problem Statement:

A hotel website is a critical tool for hotel, motel, or Airbnb owners to attract guests and showcase their services and amenities. Traditional methods of promoting hotels, such as brochures or word-of-mouth, have limitations, including :

1. Limited Reach: Physical brochures and flyers have a limited audience and cannot reach potential guests globally.
2. Lack of Real-Time Updates: Printed materials cannot be updated in real-time, leading to outdated information about room availability, pricing, or amenities.
3. Poor User Experience: Guests may find it difficult to visualize the hotel's features, rooms, and services without an interactive platform.
4. Manual Booking Processes: Traditional booking methods (phone or email) are time-consuming and prone to errors, leading to a poor guest experience.
5. Data Management Issues: Tracking guest inquiries or preferences manually is prone to errors and inefficiencies.

To overcome these issues in traditional Hotel Booking System we can develop a solution with these features:

1. Digital Showcase of Services and Amenities: Display rooms, facilities (e.g., pool, garden), and services (e.g., spa, restaurant) with high-quality images and descriptions.
2. Real-Time Room Availability and Booking: Allow guests to check room availability and book rooms online in real-time.
3. Customizable Web Pages: Enable hotel owners to customize web pages to reflect their brand and unique offerings.
4. Interactive Gallery: Create an interactive gallery of rooms, amenities, and special features to attract potential guests.
5. Online Payment Integration: Support secure online payments for bookings.
6. Guest Reviews and Ratings: Allow guests to leave reviews and ratings to build trust and credibility.
7. Responsive Design: Ensure the website is accessible on all devices (desktop, tablet, mobile).

INITIAL REQUIREMENT DOCUMENT

Title of Project	**HOTEL WEBSITE**
Stockholder involved to capture Requirement	1) Hotel Owners 2) Project Leader 3) System Admin 4) Desk Staff 5) End Users (Guests)
Techniques used for Requirement Capturing	1) Brainstorming 2) Interviewing
Name of the person along with designation	Aman Chauhan (2K24/SWE/08)
Date	February 2025
Version	1.0

Consolidated list of initial requirement:-

1. The website shall be accessible via any device (desktop, tablet, mobile) using a web browser..
2. Users (guests) shall be able to create accounts, log in, and log out securely.
3. The website shall display an interactive gallery of rooms, amenities, and special features (e.g. pool, garden).
4. Guests shall be able to view room availability and book rooms online in real-time.
5. The website shall support multiple payment methods (credit/debit cards, mobile payments, etc.).
6. Hotel owners shall be able to customize web pages (e.g. add/remove rooms, update pricing, add promotions).
7. Guests shall be able to leave reviews and ratings for their stay.
8. Guests shall be able to cancel bookings within a specified timeframe.
9. The website shall provide a search functionality for guests to filter rooms by price, amenities, and availability.
10. The website shall send booking confirmation emails to guests.
11. The website shall display contact information, location, and directions to the hotel.
12. The website shall have an admin panel for hotel owners to manage bookings, rooms, and guest information.
13. The website shall generate reports on bookings, revenue, and guest feedback.
14. The system should protect sensitive customer information and financial information from any unauthorized access.
15. Front desk staff and hotel administration shall be able to manage banquet hall reservations and t assignments.

SOFTWARE REQUIREMENTS SPECIFICATION

HOTEL BOOKING SYSTEM

Version: 1.0
Prepared By: Aman Chauhan (24/SWE/08)

Date: February 25th, 2024

1. INTRODUCTION

The Hotel Website is designed to provide an attractive and user-friendly platform for hotel owners to showcase their services and amenities, while allowing guests to book rooms, view galleries, and interact with the hotel online. The website will include the following features:

- Interactive Gallery: High-quality images and descriptions of rooms, amenities, and special features.
- Real-Time Booking: Guests can check room availability and book rooms online.
- Customizable Web Pages: Hotel owners can customize web pages to reflect their brand.
- Guest Reviews and Ratings: Guests can leave reviews and ratings to build trust.
- Responsive Design: The website will be accessible on all devices (desktop, tablet, mobile).

1.1. PURPOSE

The purpose of this document is to provide a detailed description of the Hotel Booking Website. It outlines the system's purpose, features, functionality, and interfaces, as well as the constraints under which it will operate. This document serves as a guide for both the users of the system and potential developers involved in its design and implementation.

1.2. SCOPE

- The product is titled Hotel Website (HW).
- The website will perform the following tasks:

Do's:

- The software will allow users to View room availability and book rooms online.
- Users can explore an interactive gallery of rooms and amenities.
- The Users can leave reviews and ratings.
- The software will allow users to make secure online payments.
- Hotel owners will be able to Customize web pages
- Hotel owners will be able to Manage bookings, rooms, and guest information.
- The system will generate reports on bookings, revenue, and guest feedback.

Don't:

- The system will not provide in-person concierge services or SMS notifications.

1.3. Definition, Acronym, Abbreviation

1. HBS: Hotel Booking System
2. SRS: Software requirement specification

3. RAM: Random access memory
4. DBMS: Database Management System
5. UI: User Interface
6. Admin: System Administrator
7. HTTPS: Hypertext Transfer Protocol Secure

1.4. References

- Object-Oriented Software Engineering by Yogesh Singh & Ruchika Malhotra, PHI Learning Pvt. Ltd., 2012.
- Software Engineering by K.K. Aggarwal & Yogesh Singh, New Age Publishing House, 3rd Edition, 2008.
- IEEE Recommended Practice for Software Requirements Specifications—IEEE Std. 830-1998.

1.5. Overview

The SRS contains an analysis of the requirements necessary to help easy design.
The overall description provides interface requirements for the Hotel Booking System, product perspective, hardware interfaces, software interfaces, communication interface, memory constraints, product functions, user characteristics and other constraints.
Succeeding pages illustrate the characteristics of typical naive users accessing the system along with legal and functional constraints enforced that affect Hotel Management System in any fashion.

2. Overall Description

The Hotel Website will manage room bookings, guest information, and hotel details. The system will allow guests to browse rooms, make reservations, and leave reviews. Hotel owners will have access to an admin panel to manage bookings, update room details, and generate reports.

The system facilitates seamless booking of Hotels, allowing users to book, or cancel Hotel rooms. The Hotel Booking System automates room reservations and provides real-time availability of rooms and amenities. Customers can access the system online to check room availability, make bookings, and view past transactions.

The system administrator is responsible for maintaining the following information:

- Room details (type, availability, and pricing)
- Customer details (regular and new)
- Service and amenity details (e.g., pool, garden, spa)

The administrator and support staff will perform the following functions:

- Add or update room and amenity information
- Monitor room availability
- Manage customer queries
- Record and address customer feedback

- Generate reports, including:
  - List of bookings by customers
  - Daily occupancy and revenue statistics
  - Special offers availed
  - Total available rooms
  - List of registered users and their booking history

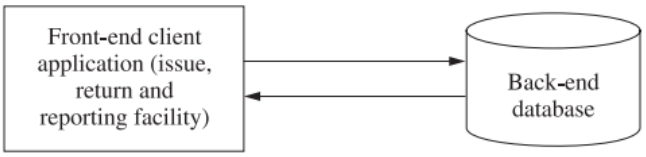
The system provides customers with the following capabilities:

- **Search for Available Rooms**:
  - Location-specific (e.g., city or area)
  - Date and duration of stay
  - Room type (e.g., deluxe, suite, standard)
  - Pricing and amenities
- **Reserve a Room**:
  - Select the duration of stay and room type.
  - Add special requests (e.g., extra bed, non-smoking room).
- **Cancel a Booking**:
  - View and cancel active bookings based on system policies.

2.1. Product Perspective

The Hotel Booking Website shall be developed using a client/server architecture and will be compatible with major web browsers and mobile devices. The front-end of the system will be developed using modern web technologies such as HTML, CSS, JavaScript, and frameworks like ReactJS for an interactive user experience. The back-end will be developed using a robust technology stack like Node.js, and the database will be managed using MongoDB.

This architecture ensures scalability, reliability, and seamless integration with third-party APIs (e.g., payment gateways and mapping services).



2.1.1 System Interfaces

None

2.1.2 User Interfaces

The Hotel Booking Website will provide the following user-friendly and intuitive interfaces:

(a) **Login/Registration:**

- Allow only authorized users to access the system through a valid login ID and password.
- Provide a registration option for new customers to create an account.

(b) **Room Details:**

- Maintain room details, such as type (e.g., suite, deluxe), amenities (e.g., pool, spa), and availability.
- Admin Functionality: Add, update, or delete room and amenity details.
- Customer Functionality: View available rooms and their features.

(c) **Customer Profile Management:**

- Manage customer details, including personal information, booking history, and special preferences.

(d) **Booking Management:**

- Handle room reservations, including booking, modifications, and cancellations.

(e) **Promotions and Offers Management:**

- Allow admins to create and update special offers, discounts, and seasonal promotions.

(f) **Reports and Analytics:**

- Generate reports for administrative purposes, such as:
    - List of bookings by customers
    - Daily occupancy and revenue statistics
    - Popular room types and amenities used
    - List of registered users and their booking history

2.1.3 Hardware Interfaces

(a) A computer with at least 2 GB RAM.

(b) Screen resolution of at least 640 × 480 or above.

(c) Computer systems will be in the networked environment as it is a multi-user system.

2.1.4 Software Interfaces

(a) MS-Windows Operating System

(b) ReactJs for designing front-end

(c) MongoDB for back-end

2.1.5 Communication Interfaces

Communication is via local area network (LAN).

2.1.6 Memory Constraints

At least 2GB RAM and 500 MB space of hard disk will be required to run the software.

2.1.7 Operations

None

2.1.8 Site Adaptation Requirements

The terminal at the client site will have to support the hardware and software interfaces specified in sections 2.1.3 and 2.1.4, respectively.

2.2. Product Functions

A system is to be implemented which can run on the web server.

- **Login and User Management:**
  - The system shall generate unique login IDs and passwords for users, administrators, and operators.
  - Administrators shall be able to manage room details, user accounts, and transaction records.
- **Room Management:**
  - Administrators can add, update, or remove room and amenity details.
  - The system shall maintain information about room availability, type, amenities, and pricing.
- **Room Booking:**
  - Customers can book rooms by selecting check-in and check-out dates, room type, and additional services.
  - The system shall allow customers to cancel confirmed reservations within a specified timeframe.
- **Guest User Access:**
  - A guest user can only check the availability of rooms and amenities but cannot make reservations.
- **Real-Time Availability:**
  - The system shall provide real-time updates on room availability based on type, date, and duration.
- **Personalized Recommendations:**
  - The system will recommend rooms and packages based on the user's preferences, previous bookings, and availability.
- **Customer Profile Management:**
  - Customers can view and update their profiles, including contact details and preferences.

- Customers can view a history of all their past bookings.
- **Reports and Analytics:**
- Administrators shall maintain transaction records, including revenue reports and booking trends.
- Generate reports for room occupancy, customer preferences, and seasonal trends.
- **Special Features:**
- The system shall allow customers to add special requests (e.g., extra bed, non-smoking room).
- Highlight amenities like pools, gardens, spas, and other features in the room details.
- **Promotions and Offers:**
- The system shall allow administrators to create and manage special promotions and discount offers.

2.3 User Characteristics

- Qualification: At least matriculation and comfortable with English.
- Experience: Should be well versed/informed about the processes of the Hotel Management.
- Technical experience: Elementary knowledge of computers.

2.4 Constraints

- There will be only one administrator for the system, responsible for overall management, such as updating Hotel details, managing user accounts, and generating reports.
- The delete operation is only available to the administrator. To reduce system complexity, there is no automated check to validate deletions. The administrator must exercise caution when deleting records, as they are responsible for maintaining data consistency.
- Users will not be allowed to update primary keys, such as reservation IDs, once they are entered into the system. This ensures data integrity across all modules.
- The system does not currently support dynamic pricing based on real-time demand, holidays, or special events. Pricing updates must be performed manually by the administrator.
- The system requires an active internet connection to check real-time availability and process reservations. Offline access is not supported.

2.5 Assumptions And Dependencies

- The Hotel facility operator will provide the initial list of Hotel spaces, including location, availability, and pricing, which will be registered in the system before its operation.
- Customers are expected to provide accurate vehicle details (e.g., license plate number, vehicle type) during registration or reservation. The system depends on this information for Hotel guidance and automated entry/exit processes.

- The administrator is responsible for creating accounts for operator staff and managing customer accounts. This includes registering new users and ensuring all details are accurate and up-to-date.
- The system administrator will create and securely communicate login IDs and passwords to administrators and support staff to ensure no unauthorized access occurs.
- The system relies on accurate and timely updates to Hotel availability and transaction records. Delays or inaccuracies in data synchronization may impact user experience and system efficiency.

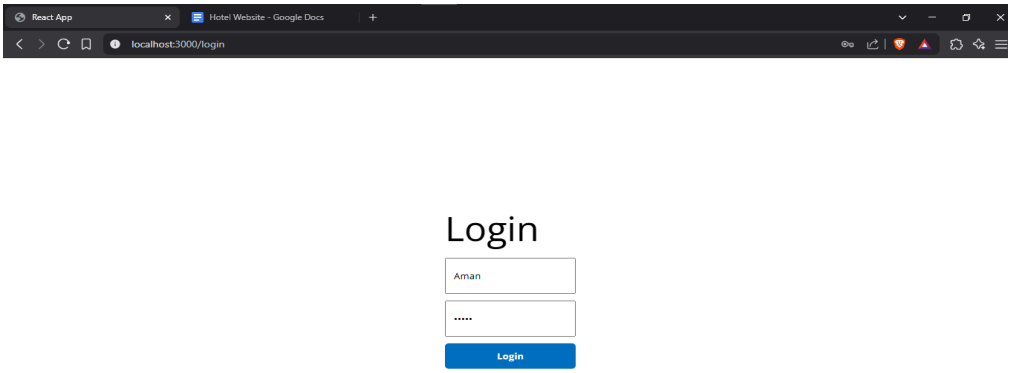
3. Specific Requirements

3.1 External Interface Requirements

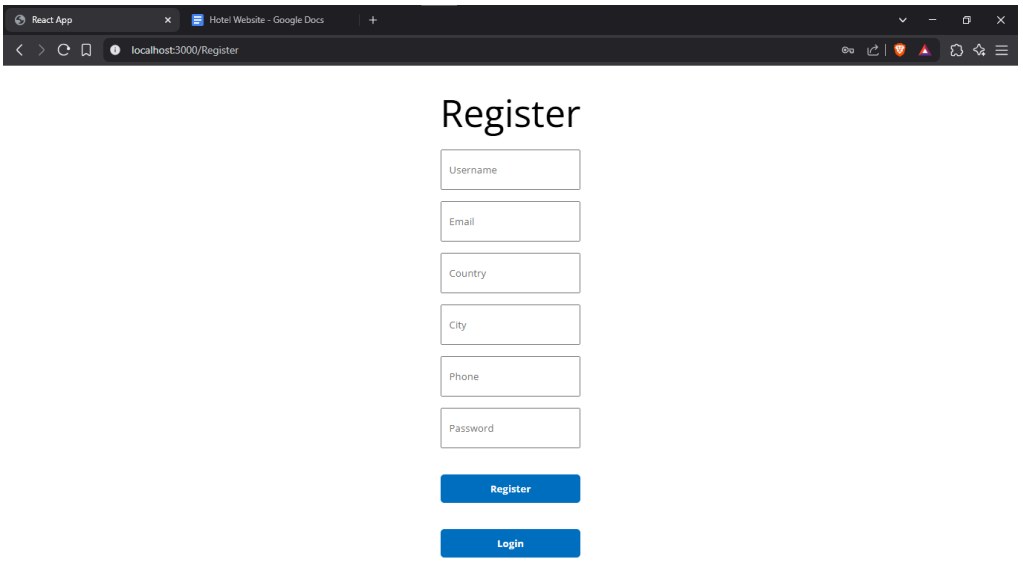
3.1.1 User Interfaces

The following user interfaces will be provided by the system.

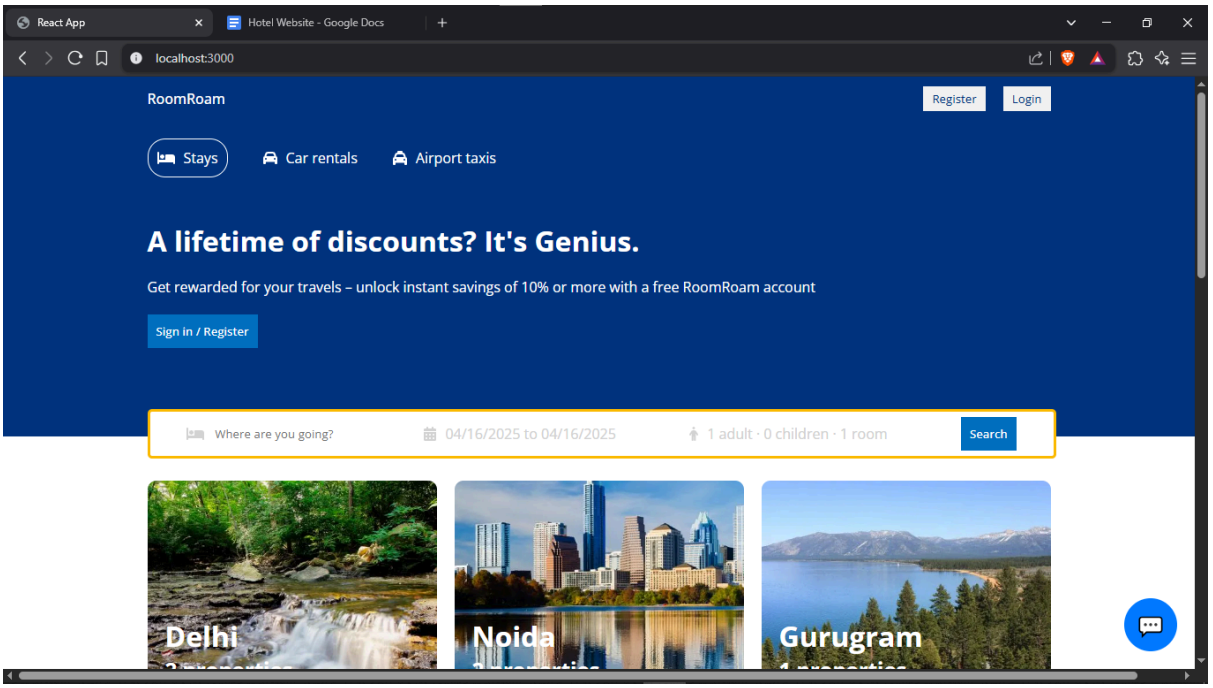
1) Login Page

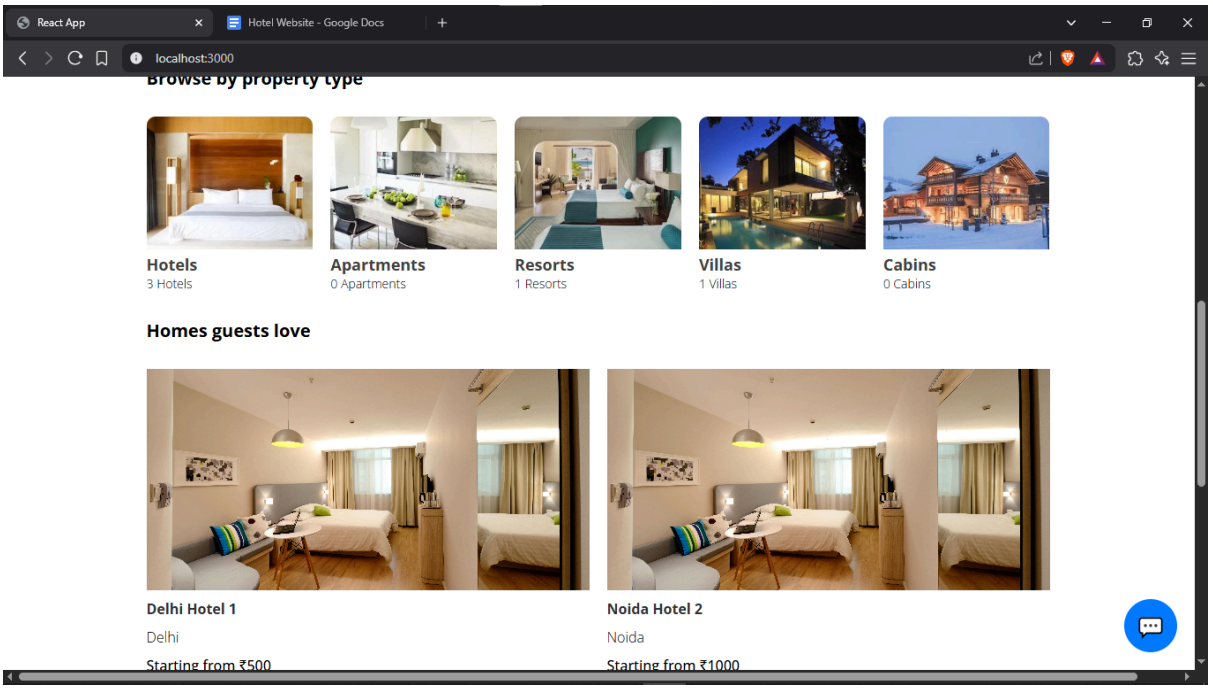


2) Register Page

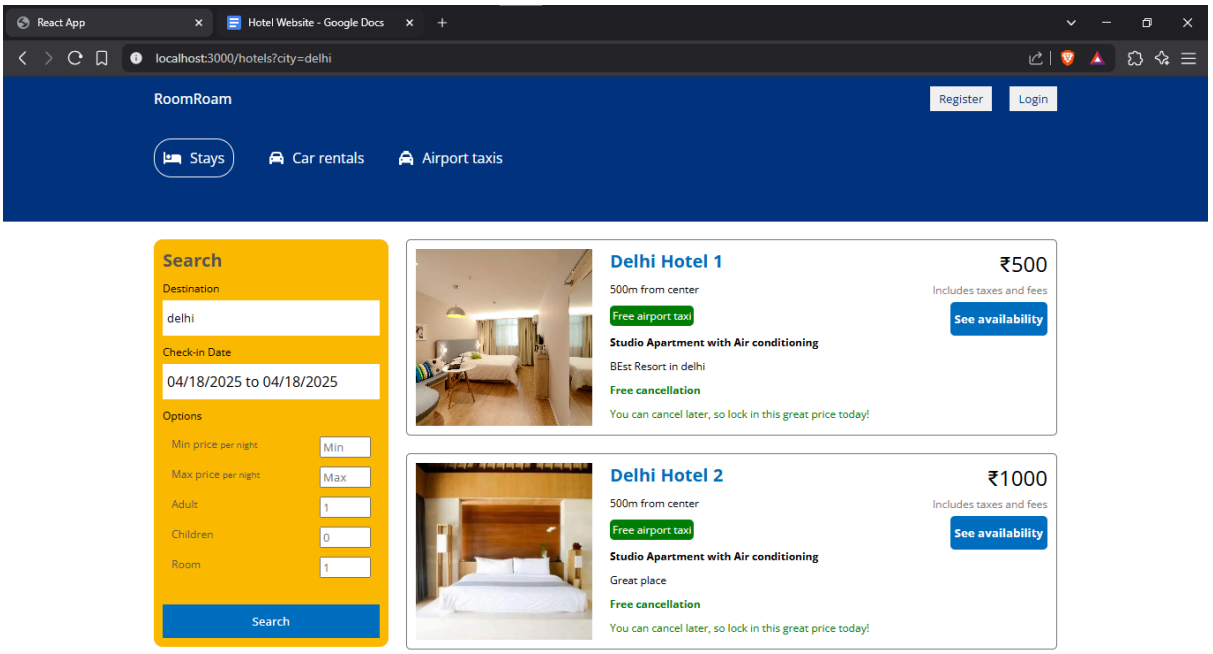


3) Home Page:

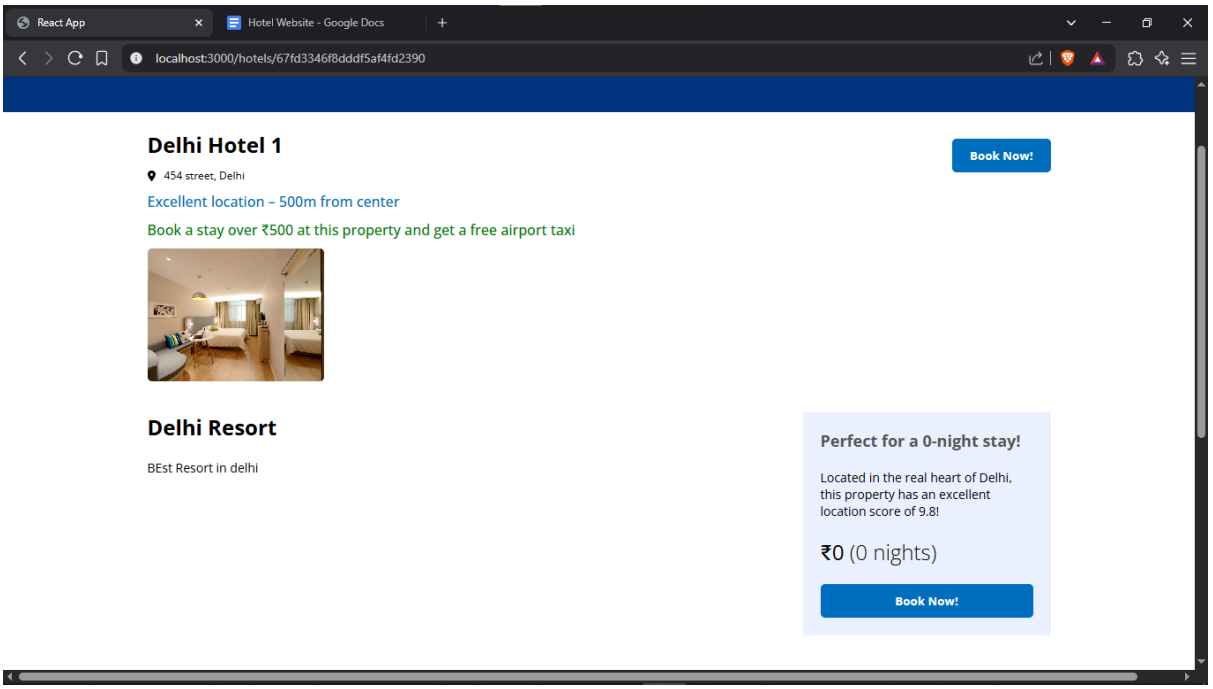

React App
Hotel Website - Google Docs
localhost:3000
RoomRoam
Register
Login
Stays
Car rentals
Airport taxis
A lifetime of discounts? It's Genius.
Get rewarded for your travels – unlock instant savings of 10% or more with a free RoomRoam account
Sign in / Register
Where are you going?
04/16/2025 to 04/16/2025
1 adult · 0 children · 1 room
Search
Delhi
Noida
Gurugram


React App
Hotel Website - Google Docs
localhost:3000
Browse by property type
Hotels
3 Hotels
Apartments
0 Apartments
Resorts
1 Resorts
Villas
1 Villas
Cabins
0 Cabins
Homes guests love
Delhi Hotel 1
Delhi
Starting from ₹500
Noida Hotel 2
Noida
Starting from ₹1000

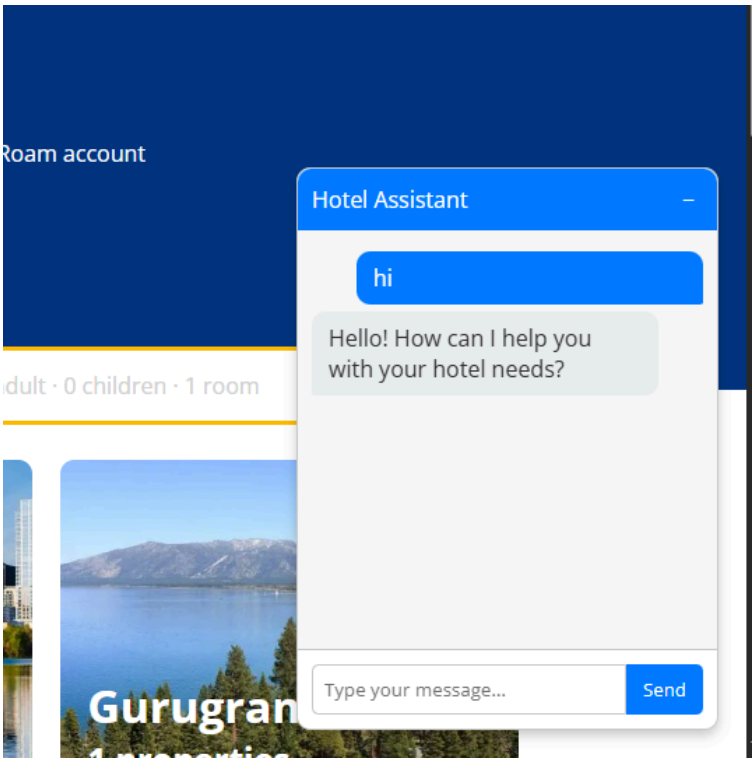
4) Hotel Browsing Page:



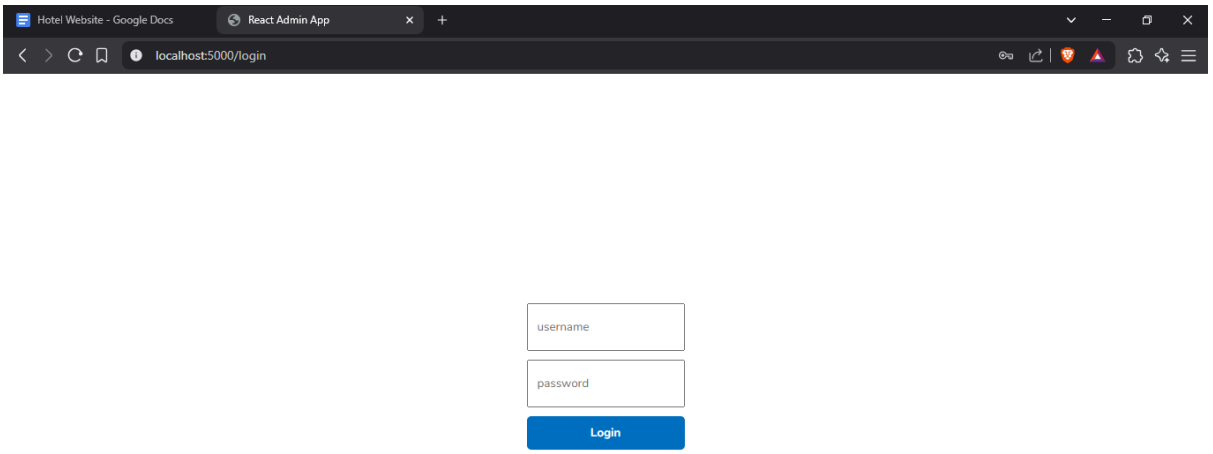
5) Hotel Booking Page:



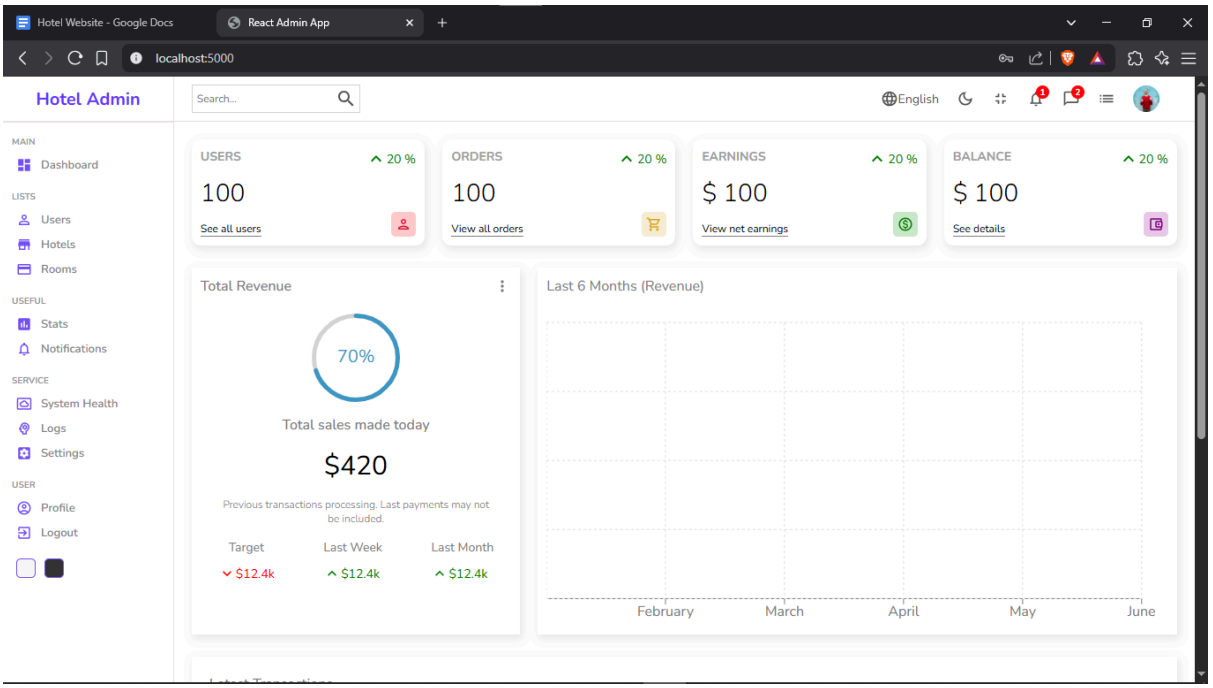
6) ChatBot Assistant:



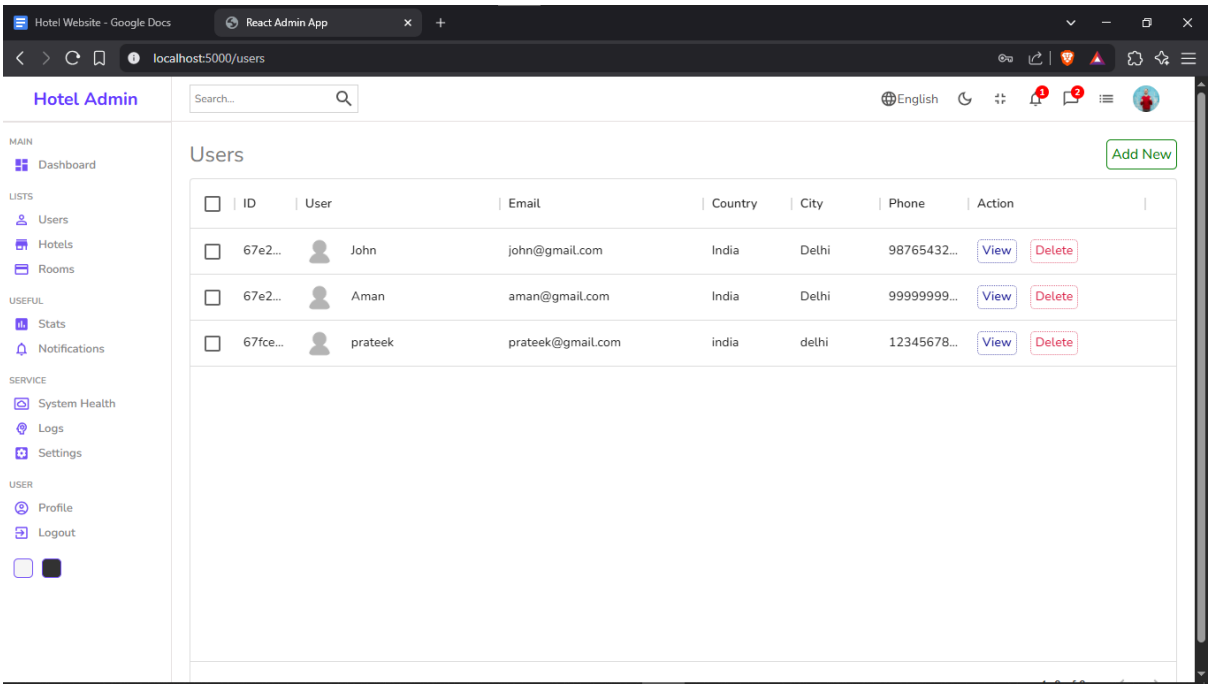
7) Admin Login Page:



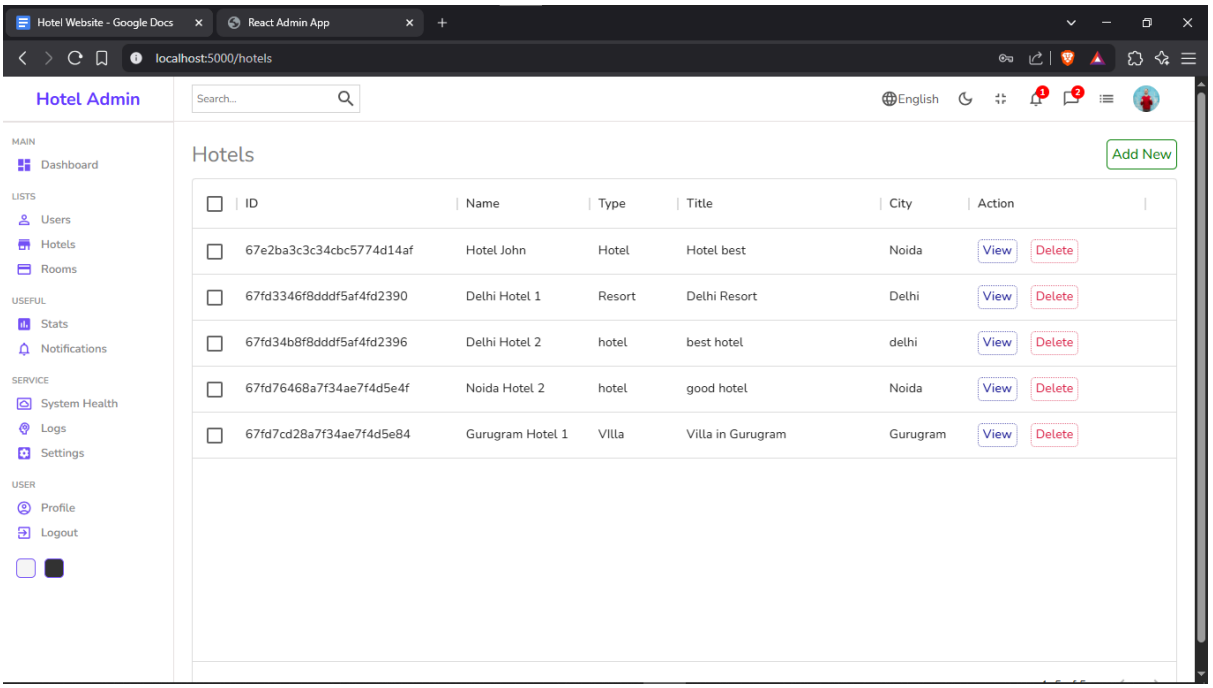
8) Admin Dashboard:



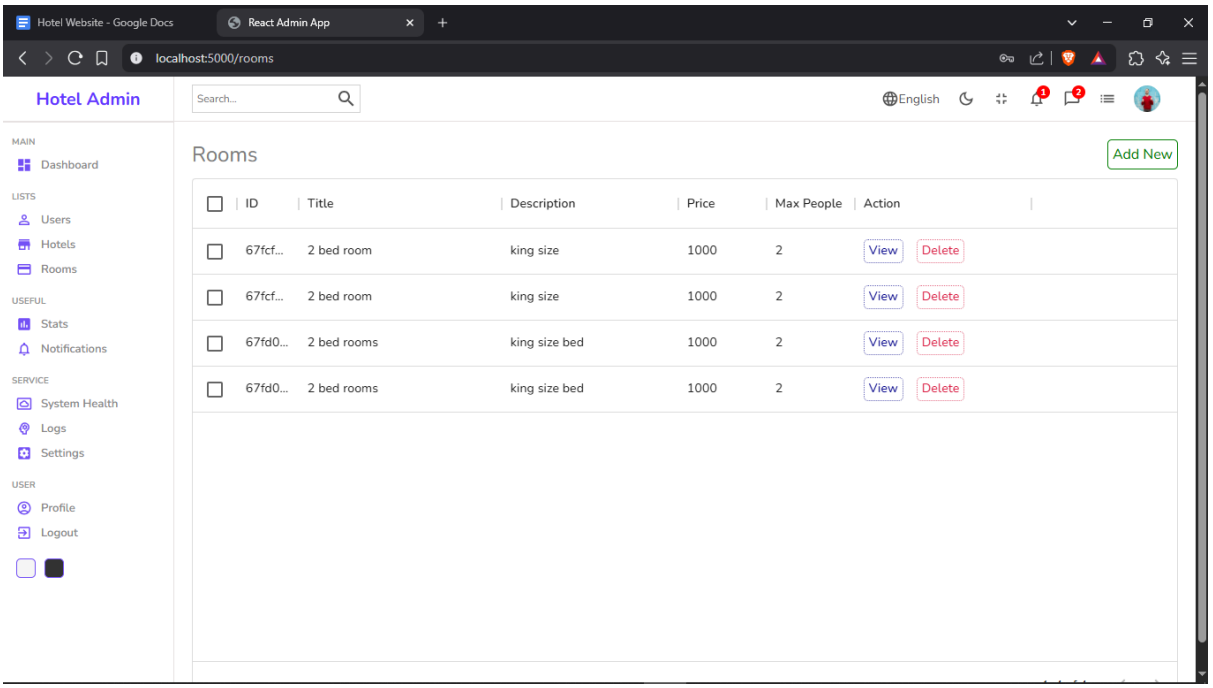
9) User's Page on Admin Side:



10) Hotels Page on Admin Side:



11) Rooms Page on Admin Side:



3.1.2 Hardware Interfacesloca

The minimum hardware requirements of Hotel Management System are a 1 Gigahertz CPU and 2 Gigabytes of RAM. The system must interface with the standard output device, keyboard and mouse to interact with this software.

3,1.3 Software Interfaces

As stated in section 2.1.4.

3,1.4 Communication Interfaces

Hotel Webiste requires an internet connection to work.

3.2 Functional Requirements

1. **Register Use Case Description**

Introduction This use case documents the steps that the users must follow in order to Register for the system to access functionalities based on their roles.
Actors: Guest, Admin
Preconditions: 1. The user must have access to the internet and a compatible web browser. 2. User does not have an existing account.
Postconditions: 1. A new user account is created and stored in the system. 2. The user can now log in to the system using their credentials.
Basic Flow: 1. User selects "Register" and fills in details. 2. System validates the email format and password strength. 3. System checks for duplicate email addresses. 4. Account is created, and a confirmation email is sent.
Alternative Flow: A1: Invalid Input 1. System detects invalid data, shows error. 2. The actor returns to the basic flow.
Special Requirements: None
Associated Use Case: None

2. **Login Use Case Description**

Introduction This use case documents the steps that the users must follow in order to log in to access functionalities based on their roles.
Actors: Guest, Admin
Preconditions: 1. The user must have access to the internet and a compatible web browser. 2. The administrator or Customer must have a valid username and password.
Postconditions: 1. The user will now log in to the system.
Basic Flow: 1. The user enters their login credentials. 2. The system verifies the credentials. 3. On successful login, the user is redirected to their dashboard. 4. Users access role-specific features.
Alternative Flow: A1: Invalid Input 1. System detects invalid data, shows error. 2. The actor returns to the basic flow.
Special Requirements: None
Associated Use Case: None

3. **Manage Hotel Reservation**

Introduction This use case describes the process of managing reservations, including making new reservations, modifying existing ones, canceling reservations, and viewing a list of reservations.
Actors: Guests, Receptionist, Admin
Preconditions: 1. The user must be logged in before this use case begins. 2. The system is operational, and the actor is authenticated. 3. Room availability data is up-to-date in the system.
Postconditions: 1. Reservations are successfully created, modified, or canceled, and all changes updated in the system's room inventory. 2. Group bookings are processed, and the reservation list is accessible for viewing by authorized actors.
Basic Flow: 1. Make Reservation 1. Guest or Receptionist initiates a reservation request by selecting dates, room type, and entering guest information. 2. The system checks room availability for the selected dates. 3. Actor confirms room selection and provides payment information if necessary.

4. The system confirms the reservation, generates a unique reservation ID, and sends a confirmation to the guest.

2. Modify Reservation

1. Guest or Receptionist initiates a request to modify an existing reservation by providing the reservation ID.
2. The system retrieves reservation details and checks room availability for the new dates or changes requested.
3. Actor updates reservation details, such as dates, room type, or number of guests.
4. The system saves the updated reservation details and sends a confirmation of changes to the guest.

3. Cancel Reservation

1. Guest or Receptionist initiates a cancellation request by entering the reservation ID.
2. The system retrieves the reservation details and verifies eligibility for cancellation based on the hotel's cancellation policy.
3. Actor confirms cancellation request.
4. The system processes the cancellation, updates room availability, and sends a cancellation confirmation to the guest.

Alternative Flow:
1. Room Unavailable for Selected Dates then the system notifies the actor and suggests alternative dates or room types.
2. Cancellation Not Allowed then the system notifies the actor and provides information on cancellation policies.

Special Requirements: Real-time updates to room availability to prevent double-booking.

Associated Use Case: Login.

4. Maintain Hotel Information

Introduction
This use case allows the actors to manage and view essential room information, including room availability, details,, maintenance requests.

Actors: Receptionist, Admin

Preconditions: 1 The actor is authenticated and authorized to perform relevant room mainten functions.
2 Room data is stored and accessible within the system.

Postconditions: 1 Room availability and details are updated based on changes made.
2 Cleaning status, maintenance requests, and inventory logs are current and accurately reflect room status.

Basic Flow: 1 View Room Availability

1. Receptionist or Manager requests to view current room availability.
2. The system retrieves and displays a list of all rooms with details on occupancy status, room type, and any maintenance or cleaning statuses.
3. Actors can filter rooms based on occupancy, type, or status.
4. The system displays the filtered room list accordingly.

2 Add/Edit Room Details

1. Admin or Manager initiates a request to add a new room or edit existing room details, including room type, rate, or amenities.
2. The system displays a form to enter new room details or update current information.
3. Actor enters or modifies room information and submits the form.
4. The system saves changes and updates room records, making new or modified room information available across the system.

Alternative Flow:
1 Room Details Already Exist then the system alerts the actor and allows them to modify existing room details instead.

Special Requirements: 1 System must provide real-time updates to reflect accurate room availability.

Associated Use Case: Login, Manage Reservation, Maintain Guest Check-In and Check-Out

5. **Maintain User Information**

Introduction
This use case involves managing user accounts within the system. It includes user login and registration, role-based access control, managing guest profiles.

Actors: Receptionist, Admin, Guests

Preconditions: 1 The system is operational, and user data is securely stored in the system database.
2 For login, users must be registered with the system

Postconditions: 1 Access to the system's functions is restricted based on user roles.
2 Guest profiles, preferences, and feedback are accurately maintained and retrievable.
3 User accounts are created or updated with accurate information.

Basic Flow:

1. User navigates to the profile page.
2. User edits fields like phone number, address, password etc.
3. System validates the input.
4. Database updates the user's profile.

Alternative Flow:
1. Login Failed Due to Incorrect Credentials

2. Process fails dues to incorrect entry in the field
Special Requirements: None
Associated Use Case: Login

6. Guest Check-In

Introduction This use case describes the process of checking guests in to the hotel.
Actors: Guests, Receptionist, Admin
Preconditions: 1 The guest has a reservation in the system for check-in. 2 The guest has a valid payment method. 3 System has necessary guest records, room availability, and rate information stored.
Postconditions: 1 The guest is successfully checked in, and their room status is updated to "Occupied."
Basic Flow: 1 Check in 1. Guests arrive at the hotel to check in. Receptionist or Admin retrieves the reservation in the system. 2. The system verifies the guest's reservation details, room assignment, and billing information. 3. Receptionist confirms guest information, updates any additional preferences , and assigns a room key. 4. The system marks the room as "Occupied" and updates the guest's status to "Checked-In." 5. Receptionist or Admin provides the guest with the room key and confirms the check-in process. 6. The system sends a confirmation of successful check-in and, if applicable, records any required deposit.
Alternative Flow: **1.** If Guest Arrives without Reservation then the system prompts the Receptionist to create a new reservation if there is room availability.
Special Requirements: None
Associated Use Case: Manage Reservation, Maintain Room Information, Billing Transaction

7. Check-Out

Introduction This use case describes the process of checking guests out of the hotel.

Actors: Guests, Receptionist, Admin
Preconditions: 1 The guest has a reservation in the system. 2 The guest has a valid payment method. 3 System has necessary guest records, room availability, and rate information stored.
Postconditions: Upon checkout, the room is marked as "Available," and any pending billing guest records are finalized.
Basic Flow: 1. Guest approaches the Receptionist to check out of the hotel. Receptionist retrieves the guest's check-in record and any pending charges. 2. The system calculates the total billing amount, including room charges, additional services, and applicable taxes. 3. Receptionist presents the final bill to the guest for review. 4. System updates the bill as settled if payment is received, noting the payment method (e.g., credit card, cash). 5. Receptionist confirms the checkout and initiates the room status change. 6. System updates room status to "Available," and the guest's record is marked as "Checked-Out."
Alternative Flow: None
Special Requirements: None
Associated Use Case: Manage Reservation, Maintain Room Information, Billing Transaction

8. **Billing Transactions**

Introduction This use case describes the process of generating invoices, processing payments associated with guest stays.
Actors: Receptionist, Admin
Preconditions: Guest has an active reservation.
Postconditions: Payment is recorded, and the transaction is complete.
Basic Flow: 1. System generates an invoice (room charges, room service, etc.). 2. Guest selects a payment method (credit card, cash). 3. Payment gateway processes the transaction. 4. System updates the payment status and sends a receipt.
Alternative Flow:

If payment fails, the system retries or suggests another method.
Special Requirements: None
Associated Use Case: None

9. **Reporting and Analytics**

Introduction This use case describes the process by which Admins can generate various reports related to occupancy and revenue
Actors: Admin
Preconditions: 1 The system has up-to-date data related to room bookings, guest information financial transactions, and guest feedback. 2 The Admin has the necessary permissions to access the reporting and analytics tools.
Postconditions: 1 Reports on occupancy, revenue, and guest feedback are generated and avai for viewing, analysis, and export. 2 The generated reports are accurate and reflect the most recent data from the system.
Basic Flow: 1. Admin selects report type (e.g., monthly revenue). 2. System queries the database for the specified period. 3. Data is compiled into charts/tables. 4. Admin exports or shares the report.
Alternative Flow: 1. No data for selected period then the system alerts the Admin and displays a message indicating no occupancy data available.
Special Requirements: None
Associated Use Case: Billing Transactions

3.3 Performance Requirements

- Login response time: <2 seconds.
- System uptime: 99.9%, with maintenance schedules as necessary.
- The system must support at least 1,000 concurrent users.

3.4 Logical Database Requirements

- Database for customer, policy, claim, and transaction records.

- Audit log for tracking access and modifications.

3.5 Design Constraints

3.5.1 Standards Compliance

- SSL/TLS for secure data communication.
- The system must comply with local and international data protection regulations, including GDPR, for handling sensitive customer information

3.7 Software System Attributes

3.7.1 Reliability

- The system must have an uptime of at least 99.9%.
- Automatic failover mechanisms should be in place to ensure minimal downtime.

3.7.2 Availability

- 24/7 availability with a goal of 99.9% uptime.

3.7.3 Security

- Data encryption and audit logging.
- Regular security audits for compliance.
- Enforce strong password policies.
- Maintain security logs for a minimum of 5 years.
- All sensitive data must be encrypted with industry-standard encryption.
- Multi-factor authentication (MFA) must be enforced for all users.

3.7.4 Maintainability

- Modular design to enable updates and scalability.

3.7.5 Portability

- The system should be deployable on different operating systems (Windows, Linux).
- A mobile-responsive design must be provided to support access from smartphones and tablets.

3.8 Organizing the Specific Requirements

3.8.1 System Mode

- Supports continuous operation with failover protocols.

3.8.2 User Class

Roles with permissions are as follows:

- **Customer**: View Hotel Space and Reserve space.

- **Operator**: Control barriers and management in Hotel place.
- **Admin**: Full control over system configurations.

3.8.3 Objects

Core objects include **Customer**, Hotel, **Reservation**, **Transaction**, **Rooms** and **Administrator**.

3.8.4 Feature

Each feature corresponds to the functional requirements defined in Section 3.2.

3.8.5 Stimulus and Response

- **Stimulus**: Actions like login, and Hotel Management.
- **Response**: The system responds within 2-3 seconds for most actions.

3.8.6 Functional Hierarchy

All functions are organized by modules, aligned with specific requirements.

3.9 Additional Comments

HMS will offer a user-friendly interface, and future versions may include mobile compatibility.

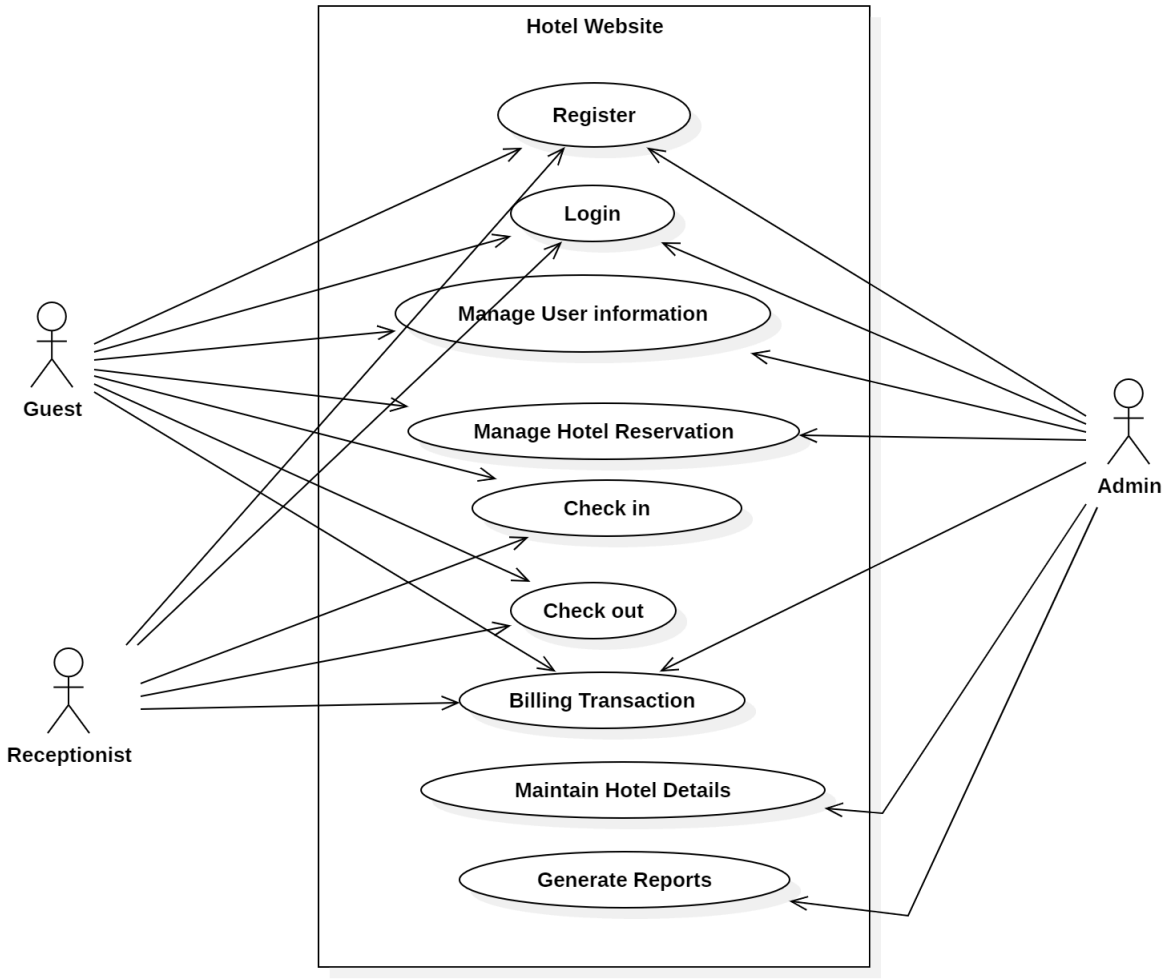
4. Supporting Information

Appendix A: Glossary

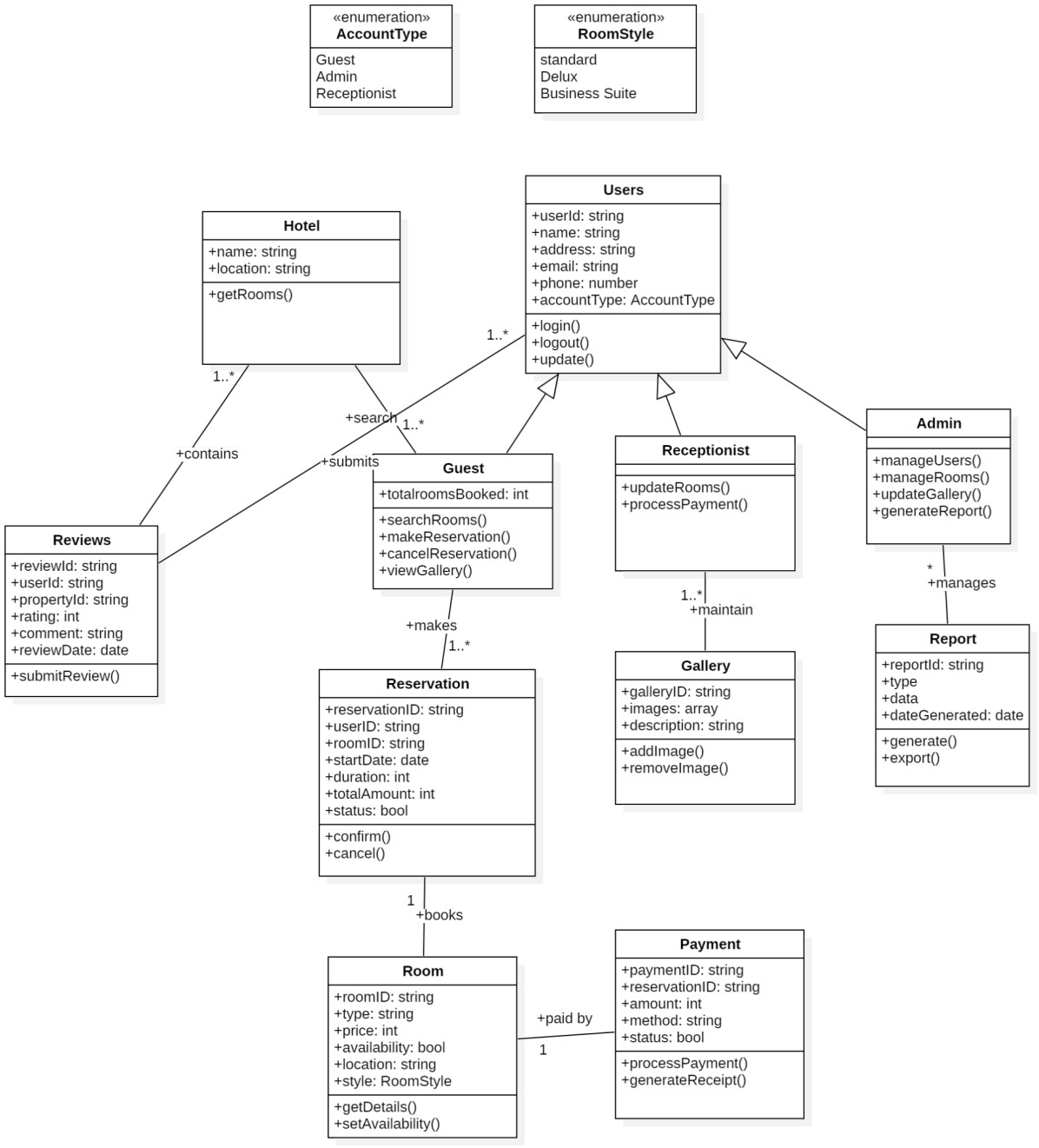
- **HMS:** HotelManagement System
- **GDPR:** General Data Protection Regulation
- **SSL/TLS:** Secure Sockets Layer / Transport Layer Security
- **IEEE:** Institute of Electrical and Electronics Engineers
- **RAM:** Random Access Memory
- **SRS:** Software Requirements Specification

Appendix B: Analysis Models

1. USE CASE DIAGRAM



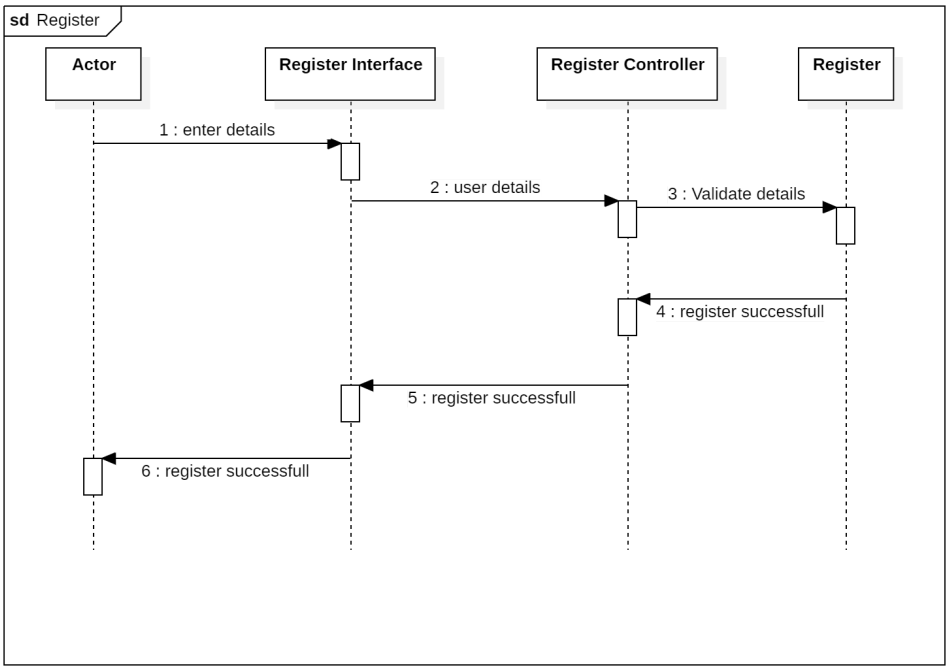
2. CLASS DIAGRAM



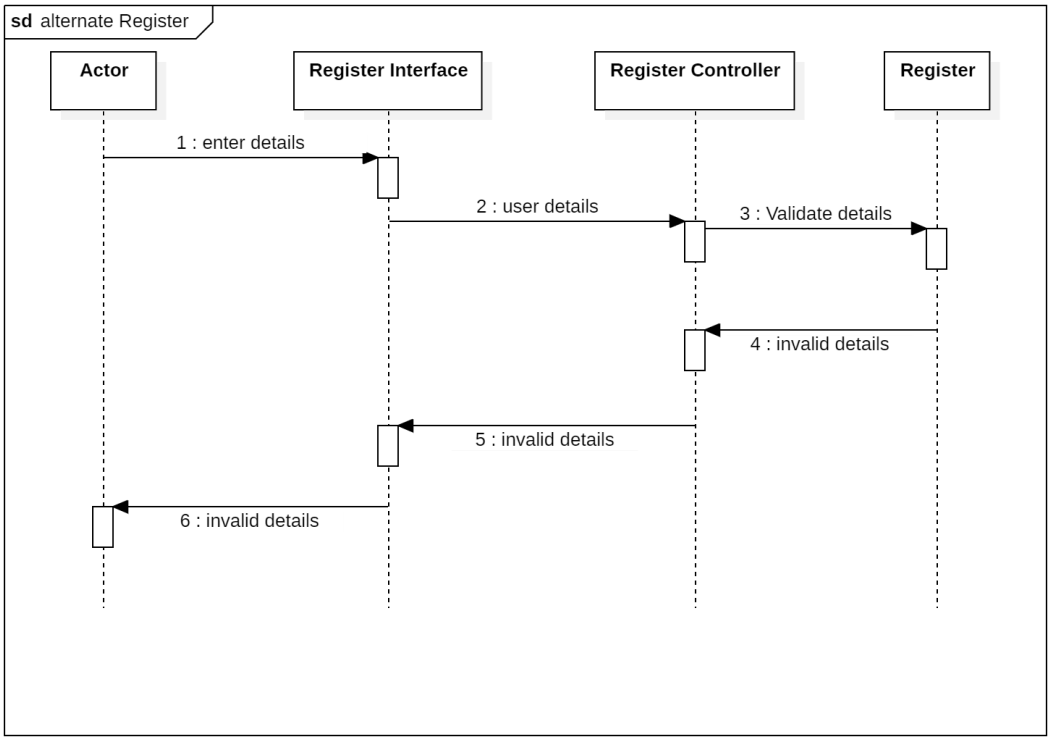
3. SEQUENCE DIAGRAM

3.1. REGISTER

3.1.1. BASIC FLOW

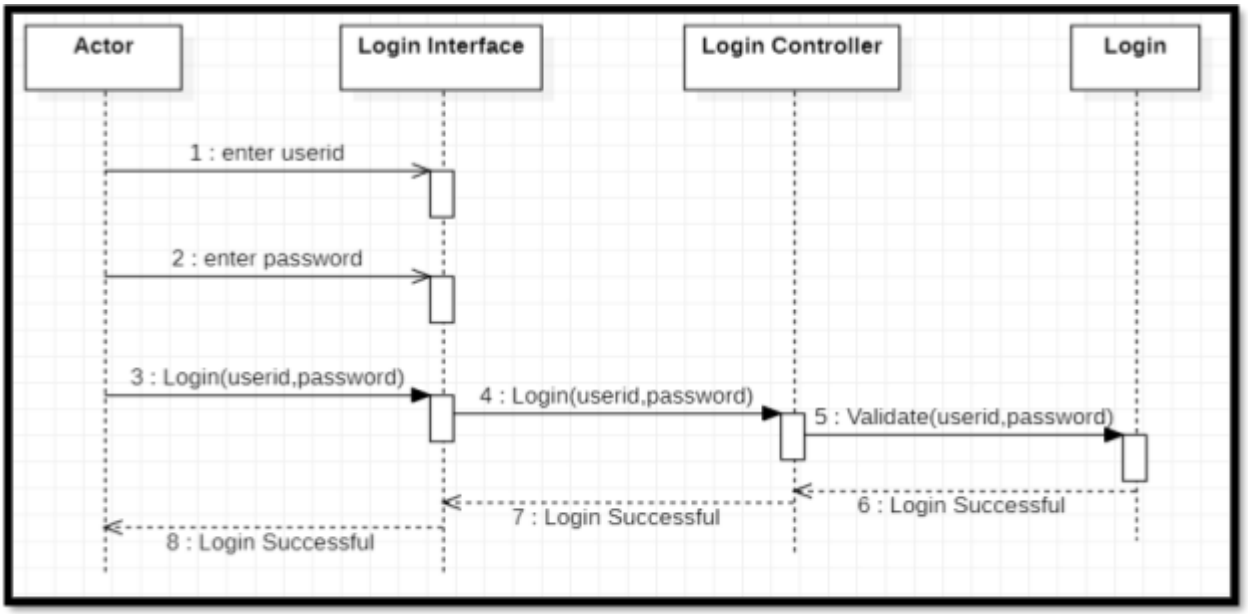


3.1.2. ALTERNATE FLOW

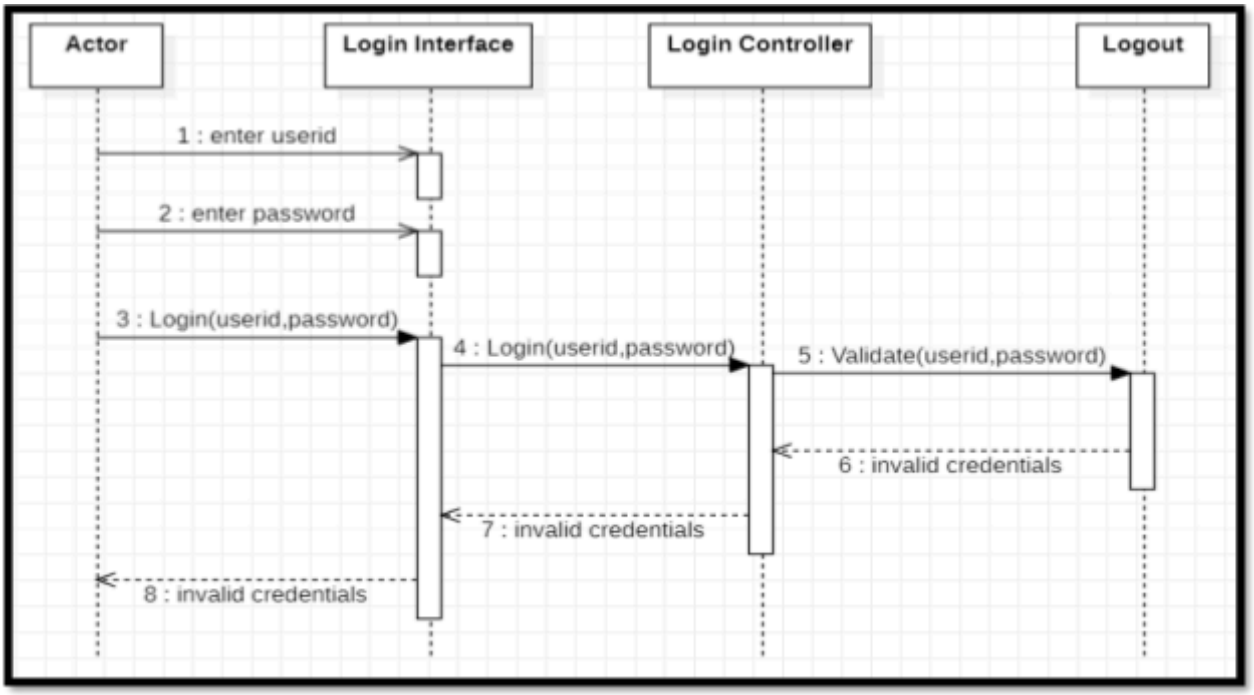


3.2. LOGIN

3.2.1. BASIC FLOW

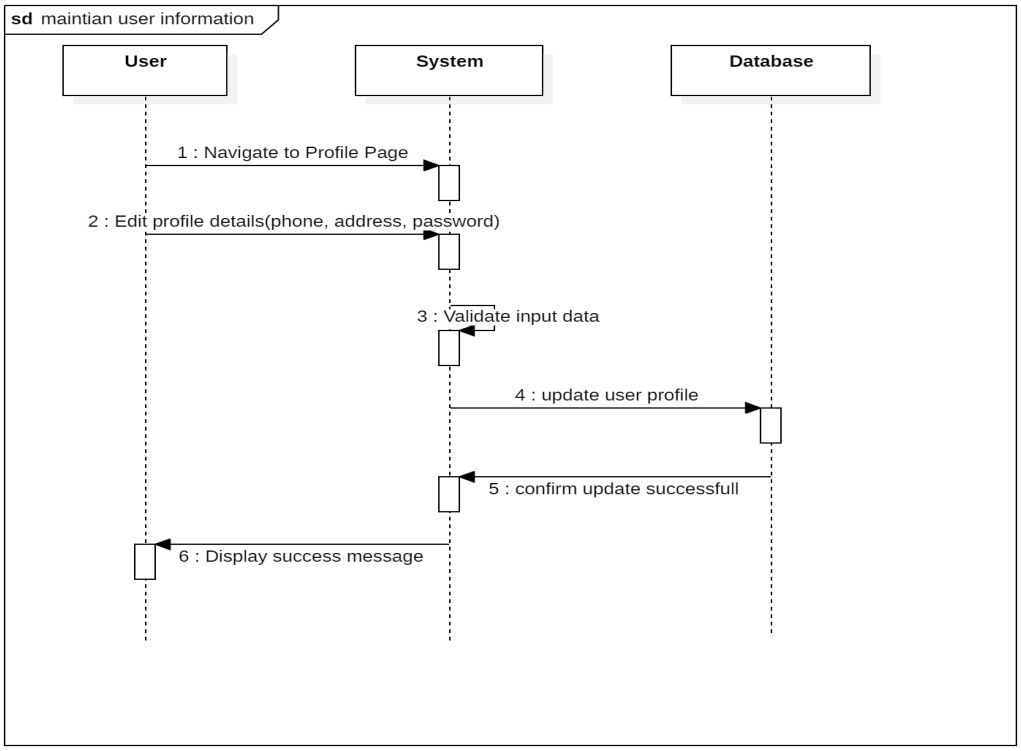


3.2.2. ALTERNATE FLOW

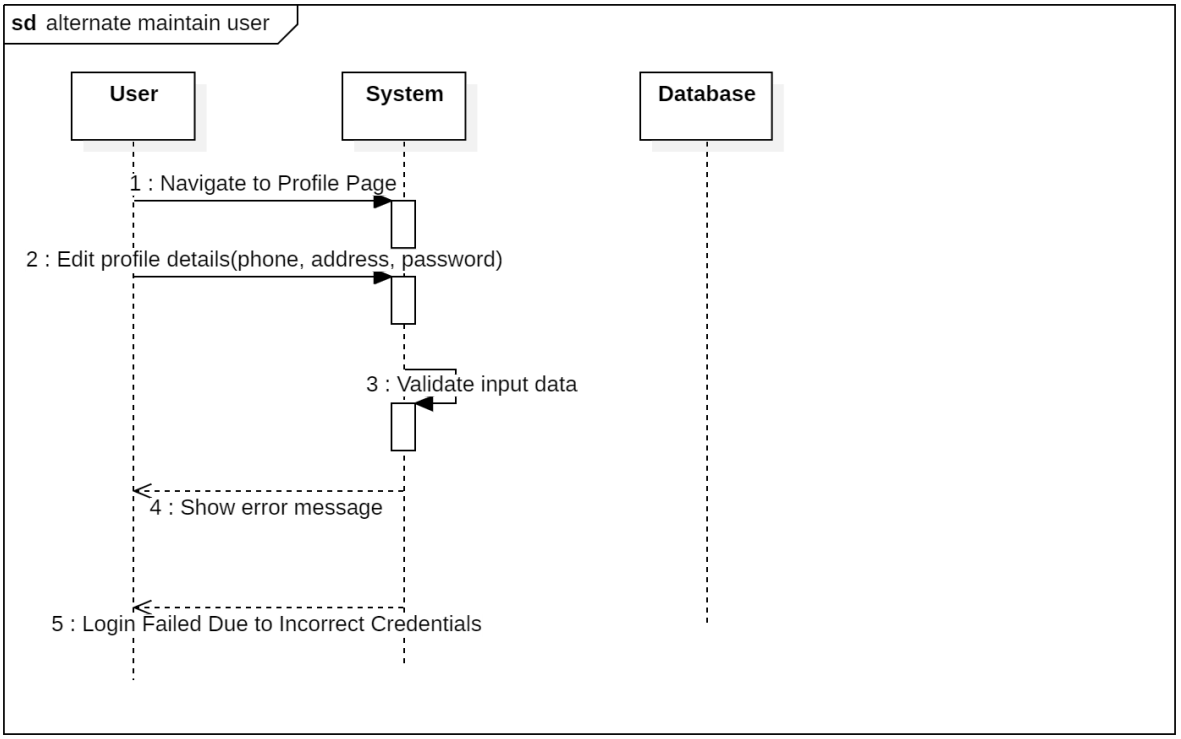


3.3 Manage User Information

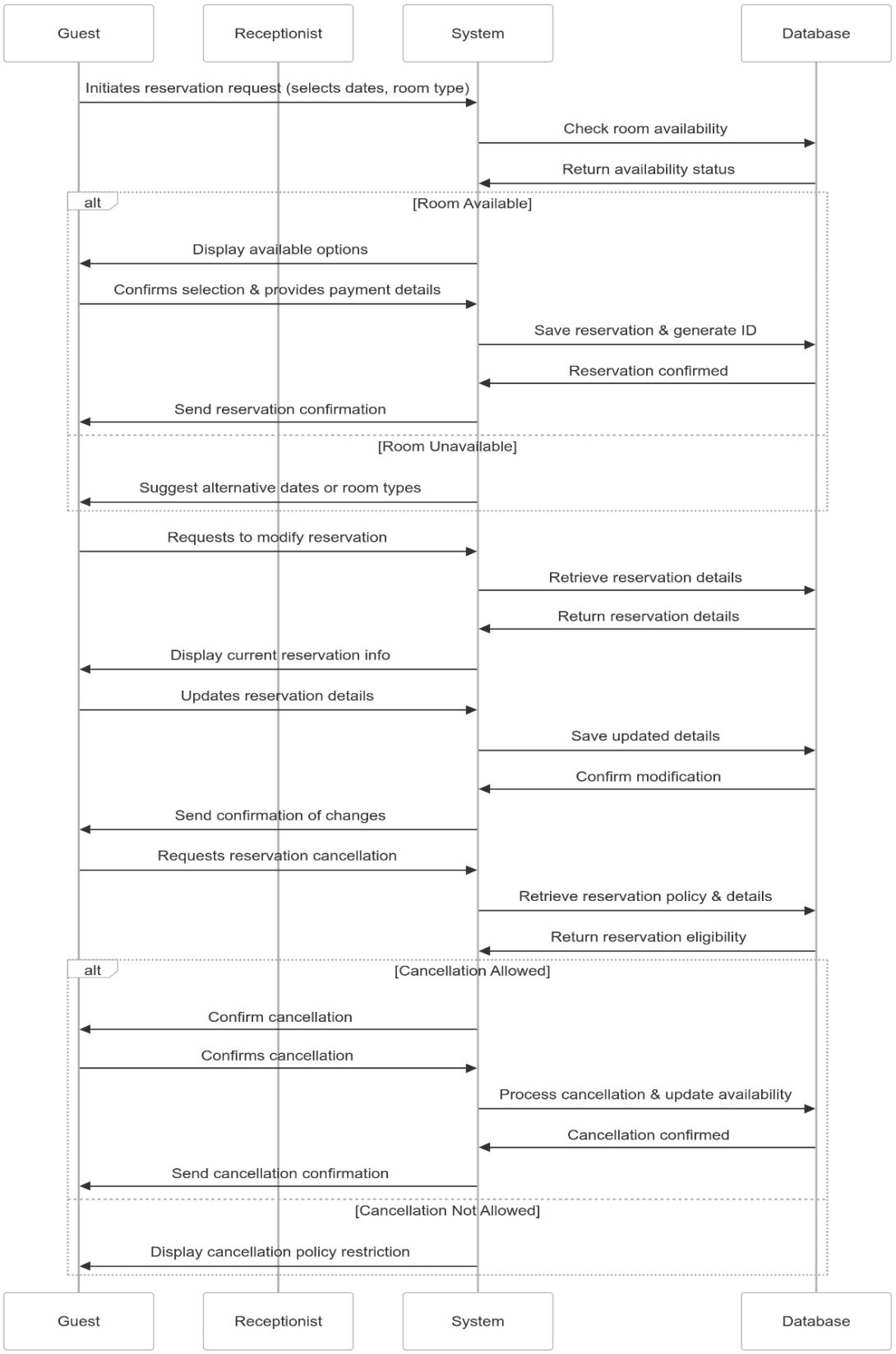
Basic Flow:



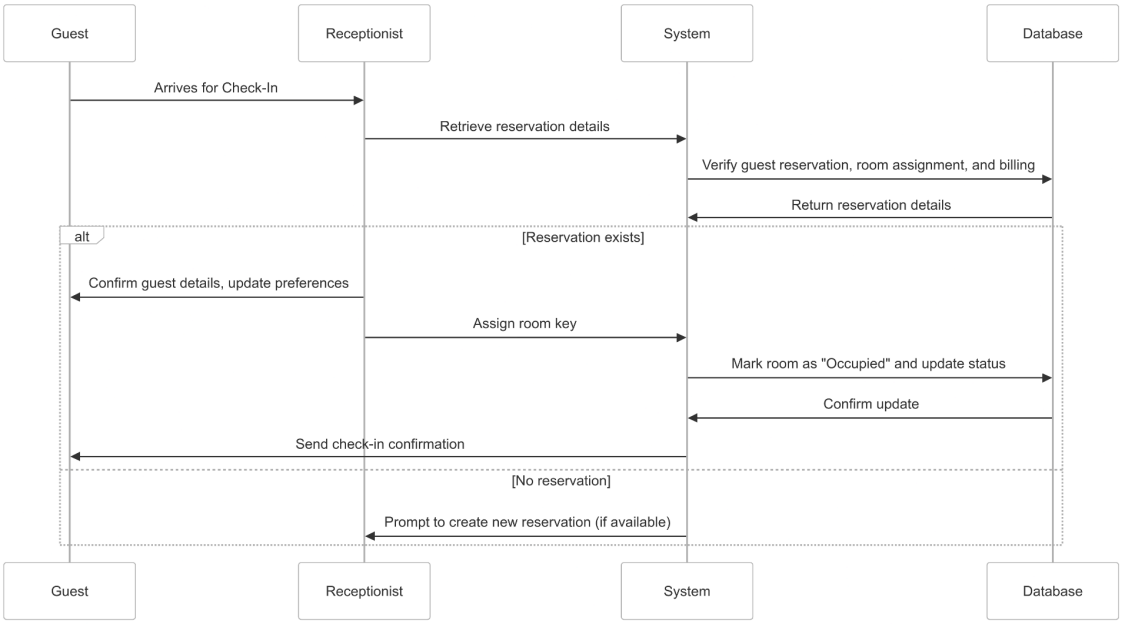
Alternate Flow:



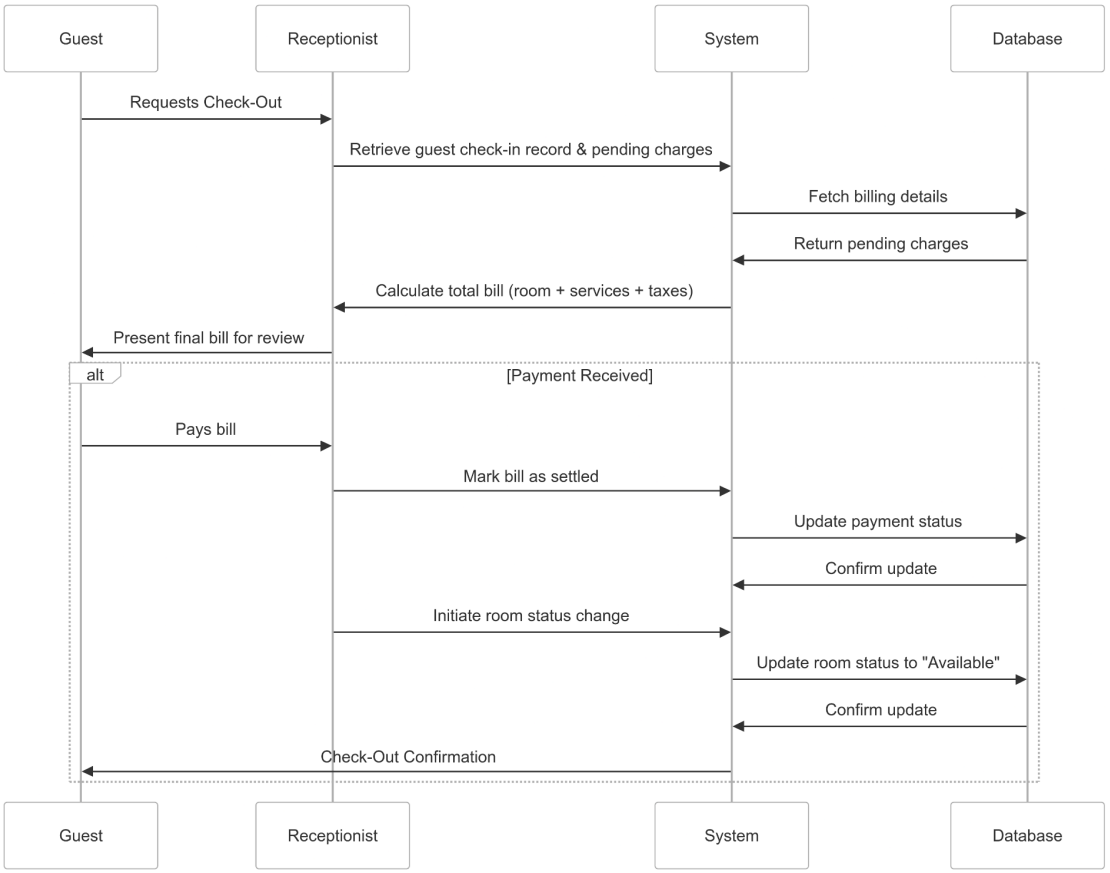
3.4 Manage Hotel Reservation



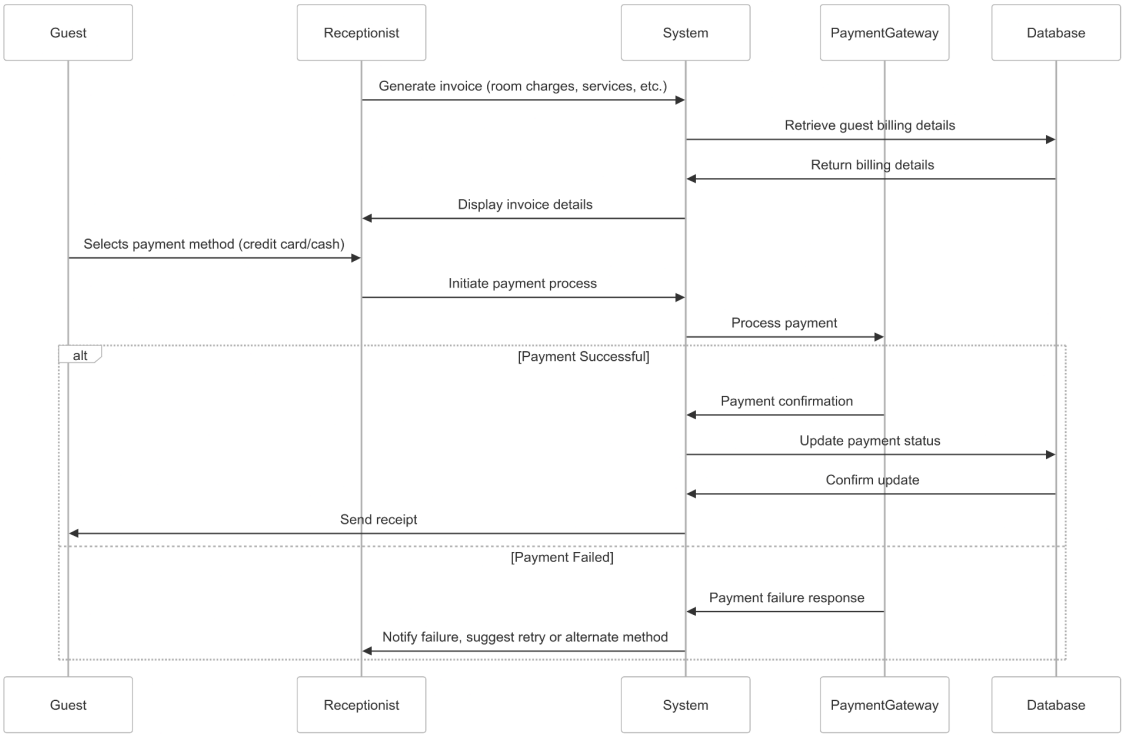
3.5 Check In



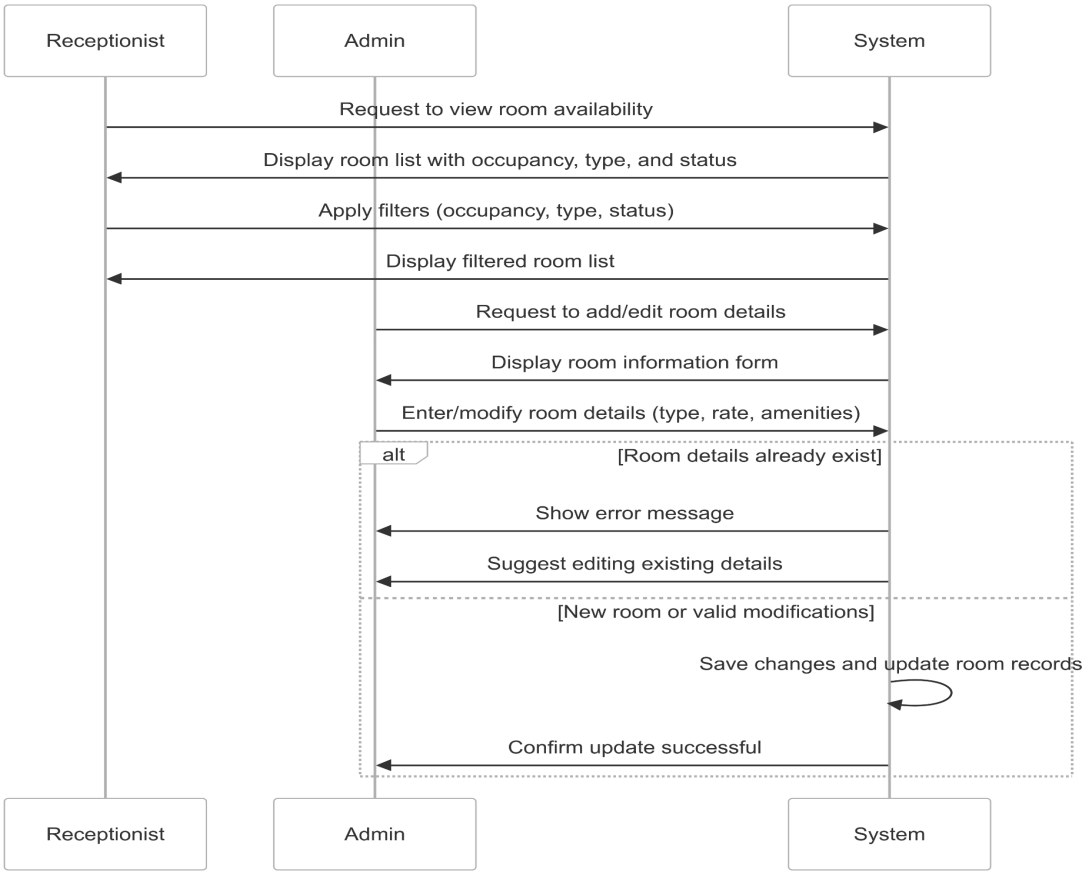
3.6 Check-Out



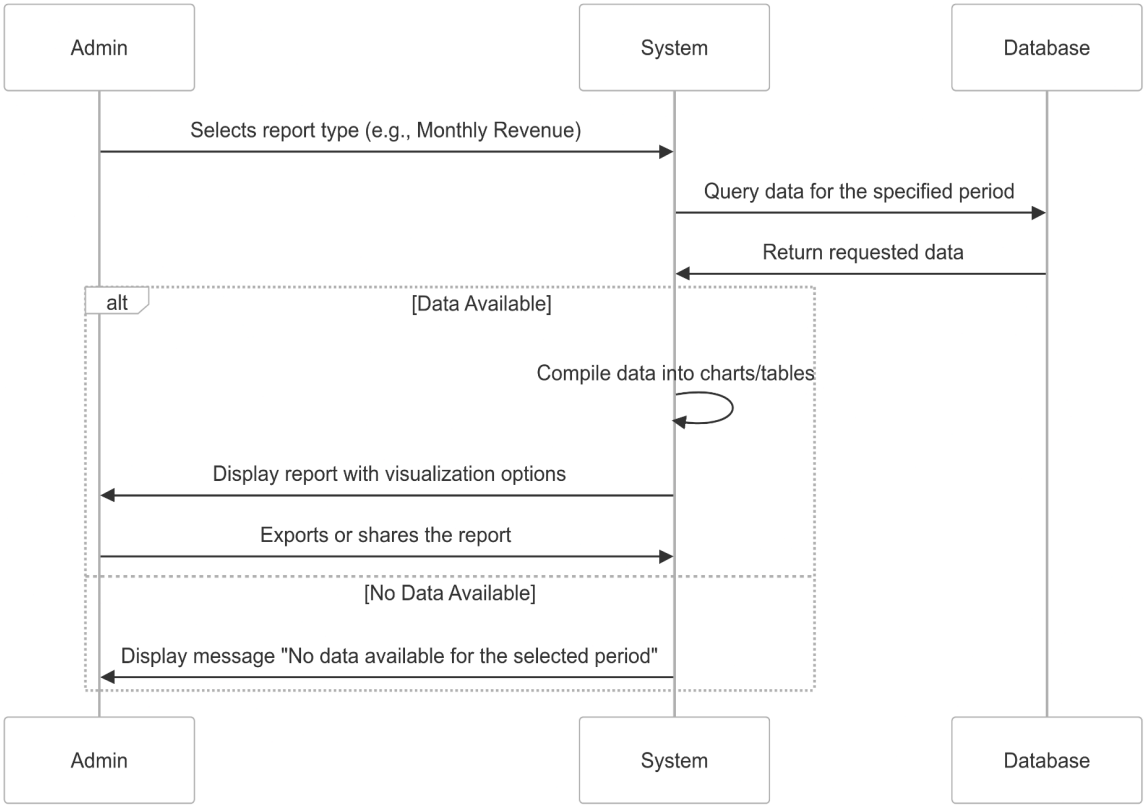
3.7 Billing Transaction



3.8 Maintain Hotel Details



3.9 Generate Reports



AI/Ml Feature in the Hotel Website Case Study:

The hotel booking website incorporates an AI-powered chatbot built using **Rasa** (version 3.5.17), a machine learning-based framework for conversational agents. Deployed in the hotel-chatbot directory and running on port 5001 with commands rasa run --enable-api --cors "*" --port 5001 and rasa run actions, the chatbot enhances user interaction by providing instant responses to queries directly on the website, particularly within the homepage and

Functionality

- **Natural Language Understanding (NLU)**: The chatbot leverages Rasa's NLU pipeline to interpret user intents and extract entities from text inputs. For example:
    - User input: "Hi" → Recognizes greeting intent, responds with "Hello! How can I help you with your hotel needs?"
    - User input: "Price of Hotel?" → Extracts hotel name (Test Hotel) and intent (price_query), responds with "The price for Test Hotel is $1000/night."
- **Interactive Assistance**: Integrated via Chatbot.jsx, it features a minimize/maximize UI, allowing users to toggle visibility while browsing hotels.
- **Predefined Responses**: Handles basic queries about hotel details (e.g., price, availability) using custom actions defined in Rasa's actions.py, connecting to the backend API (http://localhost:8800/api).
- **Contextual Dialogue**: Maintains conversation context to answer follow-up questions, enhancing user engagement.

Technical Implementation

- **Rasa Framework**: Utilizes Rasa's machine learning models for intent classification and entity recognition, trained on data in data/nlu.yml and data/stories.yml.
- **Frontend Integration**: Embedded in React components (Chatbot.jsx) with WebSocket or REST API (--enable-api) for real-time communication.
- **Training and Deployment**: Trained with rasa train and deployed with CORS enabled, ensuring seamless interaction across frontend and backend.

Benefits

- **User Convenience**: Provides 24/7 support, answering common questions without human intervention, reducing reliance on customer service.
- **Engagement**: Enhances user experience with quick, conversational responses, encouraging exploration of hotels.
- **Scalability**: ML-based NLU allows easy expansion to new intents (e.g., booking requests) by updating training data.

Limitations

- **Limited Scope**: Currently handles basic queries (greetings, prices) and lacks advanced features like booking or complex filters (e.g., "Find hotels in Delhi with a pool").

Limitations and Future Scope of the Hotel Website Case Study:

Limitations:

1. **Limited Data Availability**: The system currently has a small dataset, with only a few hotels listed, which restricts user choices and testing scalability.
2. **Basic Search Capabilities**: Search functionality is limited to city and basic price filters, lacking options like amenities, star ratings, or sorting preferences.
3. **Integrate Booking Payment**: While hotel browsing is supported, the full booking flow, including room selection and payment integration, is not fully implemented.
4. **Basic User Interface**: The interface lacks dynamic visuals (e.g., limited hotel images) and advanced features like interactive maps or detailed hotel galleries.
5. **Scalability Constraints**: The system lacks optimizations like caching or pagination, which could impact performance with a larger number of hotels or users.

Future Scope:

1. **Expanded Hotel Database**: Incorporate a larger, standardized dataset with diverse hotels across multiple regions to enhance user options and realism.

2. **Enhanced Search and Filters**: Introduce advanced filters (e.g., amenities, ratings, proximity) and sorting options to improve user search experience.

3. **Complete Booking System**: Implement a full booking pipeline, including room availability checks, secure payment gateways, and booking confirmations.

4. **Improved User Experience**: Add dynamic hotel images, interactive maps, and responsive design to make the platform visually appealing and mobile-friendly.

5. **Performance and Scalability**: Integrate caching, pagination, and database indexing to ensure the system handles increased data and user traffic efficiently.