**HOTEL MANAGEMENT SYSTEM**

**MINI PROJECT REPORT**

**BACHELOR OF TECHNOLOGY**

COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY : SUBMITTED TO :

Akanksha Raj - 07

Gorisha Babbar - 21 Mr. Jyotir Moy Chatterjee

Tejas Sethi – 69

Vaishnavi Mittal - 72



# Department of Computer Science & Engineering

# Graphic Era (Deemed to be University).

**TABLE OF CONTENT**

**S.NO CHAPTERS PAGE NO.**

1. Introduction 3
2. Problem Statement 4-5
3. Objective 6
4. Literature Survey 7
5. Software Requirement Specification 8-10
6. Screenshots 11-12
7. Result/Conclusion 13-14
8. References 15

**LIST OF TABLES:**

**S.NO PAGE NO.**

1. Table 1: Hotel Management User Interface Screens. 11
2. Table 2: Test Case Details and Results. 27-28

**LIST OF FIGURES:**

**S.NO PAGE NO.**

1. Figure 1: Use Case Diagram 15
2. Figure 2: Sequence Diagram 16
3. Figure 3: Class Diagram 17
4. Figure 4: ER Diagram 19

**INTRODUCTION-**

The hospitality industry thrives on efficiency and exceptional guest experiences. In today's digital age, a robust Hotel Management System (HMS) website is no longer a luxury, but a necessity. This document outlines the development of such a system, leveraging the power of PHP, JavaScript, HTML, CSS, and MySQL, facilitated by XAMPP.

**Understanding the Need: A Growing Industry Demands Smarter Solutions**

The travel and tourism industry is experiencing a surge, fuelled by a global desire for exploration and diverse experiences. This growth necessitates efficient management solutions for hotels to remain competitive. A well-designed HMS website tackles this challenge by streamlining various aspects of hotel operations:

* **Simplified Room Bookings and Reservations:** Guests can conveniently search for available rooms based on their preferences (dates, room type, occupancy) and seamlessly book their stay directly through the user-friendly interface. This eliminates the need for phone calls or emails, enhancing guest convenience.
* **Effortless Guest Management:** The system streamlines guest check-in and check-out processes, minimizing wait times and maximizing guest satisfaction. Additionally, it allows receptionists to efficiently manage guest requests, preferences, and communication, fostering a more personalized experience.
* **Centralized Inventory Management:** Maintain complete control over room availability and eliminate the risk of overbooking with a centralized inventory system. The HMS integrates with housekeeping operations, providing real-time room status updates and guest requests, ensuring a smooth flow of operations.
* **Integrated Payment Processing:** Offer guests a secure and convenient way to pay for their reservations directly through the HMS website. This eliminates the need for manual transactions and streamlines the payment process.
* **Food and Beverage Ordering:** Guests can conveniently order room service, meals at on-site restaurants, or pre-order special requests directly through the HMS website. This enhances guest convenience and increases revenue opportunities for the hotel.
* **Guest Communication and Engagement:** The HMS facilitates seamless communication between guests and the hotel staff. Guests can submit inquiries, make requests, and receive timely updates through integrated messaging systems or email notifications. This fosters a more connected and responsive guest experience.
* **Marketing and Promotions:** Utilize the HMS website as a powerful marketing tool. Showcase the hotel's unique offerings, highlight amenities, curate personalized promotions, and offer targeted discounts to attract new guests and encourage repeat bookings. Integrate social media platforms and online review sites to amplify your reach and build a strong online reputation.

**The Technological Powerhouse: Building Blocks of the HMS**

This project leverages a powerful combination of technologies to deliver a robust and user-friendly HMS website:

* **PHP (Hypertext Preprocessor):** As the server-side scripting language, PHP acts as the backbone of the system. It handles user interactions, processes data, interacts with the database, and dynamically generates web content based on user actions and database information.
* **JavaScript:** This dynamic scripting language adds an interactive layer to the user interface. JavaScript enables features like form validation, dynamic content updates on the website without complete page reloads, and asynchronous communication with the server, enhancing user experience and responsiveness.
* **HTML (HyperText Markup Language) and CSS (Cascading Style Sheets):** HTML forms the foundation of the website's content structure, defining the layout and elements displayed on the web pages. CSS styles the HTML content, ensuring an aesthetically pleasing and user-friendly interface that adapts to different devices.
* **MySQL:** This powerful relational database management system (RDBMS) serves as the central repository for all hotel data. Information on users, bookings, rooms, amenities, rates, and guest preferences are securely stored and efficiently managed by MySQL.
* **XAMPP:** This open-source solution offers a convenient local development environment by integrating Apache (web server), MySQL (database), PHP, and Perl. XAMPP streamlines the development process by eliminating the need for individual configuration and setup of these components, allowing developers to seamlessly test and debug the HMS website.

**This combination of technologies empowers the creation of a feature-rich HMS website, optimizing hotel operations, elevating guest experiences, and ultimately driving business growth in the competitive hospitality landscape.**

**Problem Statement:**

The hospitality industry is experiencing a boom, with travellers seeking diverse experiences worldwide. However, managing a hotel effectively involves complex tasks like reservations, guest communication, housekeeping, inventory control, and revenue management. Traditional methods, reliant on paper records and manual processes, create bottlenecks and hinder efficiency, leading to:

* **Time-consuming Operations:** Manual reservation processes, check-in/out procedures, and inventory management are laborious and prone to errors.
* **Limited Guest Accessibility:** Guests lack convenient 24/7 access to booking, managing reservations, or accessing hotel information.
* **Ineffective Communication:** Inefficient communication channels can lead to delays in addressing guest inquiries and requests.
* **Lack of Personalization:** Without guest data insights, hotels struggle to personalize services and amenities for a more fulfilling guest experience.
* **Data Silos and Limited Insights:** Scattered data across various sources hinders informed decision making regarding pricing strategies, marketing campaigns, and service optimization.
* **Inefficient Payment Processing:** Manual payment transactions are slow and inconvenient for guests.
* **Inventory Management Challenges:** Risk of overbooking and difficulty in managing room availability in real-time.

These issues culminate in subpar guest experiences, impacting customer satisfaction and loyalty.

**Proposed Solution:** A Hotel Management System (HMS) website

This project proposes the development of a user-friendly HMS website to streamline hotel operations, enhance guest experiences, and unlock new avenues for growth. The HMS website will leverage a combination of PHP, JavaScript,HTML, CSS, and MySQL facilitated by XAMPP.

**OBJECTIVE-**

This Hotel Management System (HMS) aims to revolutionize hotel operations by focusing on two key objectives:

**1. Optimizing Hotel Efficiency:**

* **Effortless Reservation Management:** Automate reservation processes, including online booking, check-in/out,and room assignment. Free staff from administrative tasks, allowing them to focus on personalized guest service.
* **Centralized Inventory Control:** Maintain real-time visibility over room availability and eliminate the risk of overbooking with a centralized inventory system. Integrate with housekeeping for efficient room status updates and guest request management.
* **Integrated Payment Processing:** Offer secure and convenient online payment options for reservations and additional services, streamlining transactions and reducing manual workload.
* **Data-Driven Decision Making:** Collect and analyze guest data on booking patterns, preferences, and feedback.Utilize these insights to optimize pricing strategies, resource allocation, and marketing campaigns for maximum profitability.

**2. Elevating the Guest Experience:**

* **24/7 Guest Accessibility:** Empower guests with a user-friendly platform to book rooms, manage reservations, and access hotel information anytime, anywhere, on any device.
* **Seamless Communication:** Facilitate two-way communication through integrated messaging systems or email notifications. Address guest inquiries and requests promptly, fostering a more responsive and personalized experience.
* **Personalized Service Delivery:** Leverage guest data to personalize services and amenities. Offer pre-set room temperatures based on guest preferences or suggest in-house spa treatments based on booking history.
* **Enhanced Convenience:** Guests can conveniently order room service, meals, or pre-order special requests directly through the HMS website. This simplifies guest experience and increases revenue opportunities for the hotel.

By achieving these objectives, the HMS will not only streamline hotel operations but also create a more positive and memorable experience for guests, ultimately driving business growth and fostering long-term guest loyalty.

**LITERATURE SURVEY-**

(Louw, 2006) developed a hotel management system that can be used online. This system allows the guests to do their booking online by them self. Some of tasks that the system can do are providing a query for arriving date and the length of staying, providing the number of on rooms, view all available rooms and provides the user the ability to choose one or more of them, recording kind of guests and how many going to be in the single room, providing the cost of booking, asking the users if they want additional service; such as, diner or breakfast, storing the guests details; like, name, address and telephone, asking the user for confirmation, final confirmation views with the detail of booking and the guests can review or cancel the booking.

For more understanding to the system the author provided some figures with explanation. The first figure shows the relationship between the end user and the webserver and how the users interface start and the application is done step by step. The second figure shows the relationship between the user and the screen. The screen transfer HTML codes to interface and when the user interact with it some process are done then the screen shows another pages. While the third figure shows how each page is related to each other.

Fernandez & Yuan (1999). An Analysis Pattern for Reservation and Use of Reusable Entities is an article that focuses on the pattern for making a reservation for hotel. This article is written by Fernandez and Yuan. First it goes through the problem then solution, requirements and consequences. The article is supported with some diagrams that make the understanding more clear. The problem occurs when the user needs to reserve a hotel room or vehicle for that the system needs a table for hotel rooms and vehicle. In order to avoid duplicate reservation for same room or vehicle the system needs a table for available rooms and vehicle. Also, the user may needs to review the reservation so the system has a table for reservation.

In the solution part the article first discussed the use case for requirements. The first use case is making reservation, in this case the user make research for available demand by using date. When the demand which is room or vehicle is available the user can do reservation. The second use case is use a reserved entity. In this case when the usage of reservation is done, the room or vehicle is recorded as available again. The third use case is modifying the reservation. The fourth use case is cancelling the reservation.

Lauesen (2003) The author wrote a paper to describe the functional requirement for hotels and hospitals. Theses functional requirements indicate what the system shall do, data requirements indicate what it shall store, and quality requirements indicate how quickly or how easily it shall perform. This article focuses on functional requirements, which usually describe a system's input, output, and the relationship between the two.

The article has 3Unified Modelling Language use case diagram between the receptionist and the hotel system. The first UML deals only with the computer system's actions; the second a use case that specifies separate human and computer actions; and the last is a task descriptions, which do not separate human and computer actions. These UML diagram helps us to understand the functional requirements a hotel management system.

Also the article include figures, each figure describe subtasks. The first figure is a task descriptions for a hotel's reception work area. The second figure is a tasks and support description of hotel check-in. The last figure describe high-level tasks for innovation and business process redesign.

Ahmad (2012)Based on the results of statistical analysis, which show the impact of Customer Relations Management on the level of client's satisfaction, confidence and the retaining of the clients of the State Hotel of India, it is clear that there is no company in the world dropped from defects and imperfect. Without doubt, every company should scrutinize and upgrade its plans despite of its success to identify the new strategies, which appropriate to the developed management surroundings, which specified the needs and desires of old, or new consumers.

Popat (2013)Work in the area of Hotel Management involves ensuring that all operations, including accommodation, food and drink and other hotel services run smoothly. It includes instructions in hospitality administration, hotel or motel management, restaurant and food services management, facilities planning, leisure studies, recreation administration, marketing, recreation equipment and grounds operations and maintenance, business finance, insurance and taxation, event management and guest services, personnel management, travel and logistics management, safety and health services, professional standards and ethics, and applications to specific vacation types and locations.

**SOFTWARE REQUIREMENT SPECIFICATION (SRS)-**

The following subsections of the Software Requirements Specifications (SRS) document provide an overview of the entire SRS.

* 1. **Definitions, Acronyms, and Abbreviations.**

SRS – Software Requirements Specification

HMS – Hotel Management System

Subjective satisfaction – The overall satisfaction of the system

End users – The people who will be actually using the system

* 1. **Overview**

The SRS is organized into two main sections. The first is The Overall Description and the second is the Specific Requirements. The Overall Description will describe the requirements of the HMS from a general high level perspective. The Specific Requirements section will describe in detail the requirements of the system.

* 1. **The Overall Description**

Describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead it provides a background for those requirements, which are defined in section 3, and makes them easier to understand.

* 1. **Product Perspective**

The HMS is an independent stand–alone system. It is totally self contained.

**3.5 Hardware Interfaces**

The HMS will be placed on PC’s throughout the hotel.

**3.6Software Interfaces**

All databases for the HMS will be configured using Oracle 10g. These databases include hotel rooms and customers information. These can be modified by the end users. The room database will include the room numbers and if they are vacant or occupied. The customers' information database will contain all the information of the customer such as first name, last name, phone number, credit card number, confirmation number, check in date, check out date and time…etc.

* 1. **Product Functions**

Reservation and Booking System

* Allow the reception to modify, add or delete reservation.
* Allow the reception to modify, add or delete guest detail.
* Allow the reception to provide the guest with the payment detail.
* Display list of on and off rooms.
* Search about specific guest.
* Display detail about reservation.

General Management Services and Automated Tasks System

* Allow the administrator to modify, add or delete the room.
* View a daily, weekly and monthly report.
  1. **User Characteristics**

Educational level of HMS computer software – Low

Experience of HMS software – None

Technical Expertise – Little

* 1. **Apportioning of Requirements**

The audio and visual alerts will be deferred because of low importance at this time.

**3.10Specific Requirements**

This section contains all the software requirements at a level of detail, that when combined with the system context diagram, use cases, and use case descriptions, is sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements.

* + 1. **External Interfaces**

The Hotel Management System will use the standard input/output devices for a personal computer. This includes the following:

* Keyboard
* Mouse
  + 1. **User Interfaces**

The User Interface Screens are described in table 1.

|  |  |
| --- | --- |
| Screen Name | Description |
| Login | Log into the system as Administration |
| Reservation | Retrieve button, update/save reservation, cancel reservation, modify reservation, , accept payment type/credit card |
| Hotel Payment | Accept payment for room. |
| Customer Record | Delete or update customer records |
| Administer Rooms | Add, modify and delete rooms. |
| Reports | Select, view |

**Table 1: Hotel Management User Interface Screens**

* + 1. **Software Interfaces**

The system shall interface with an Oracle.

* + 1. **Hardware Interfaces**

The system shall run on a Microsoft Visual Basic (ASP.net).

* + 1. **Communication Interfaces**

The system shall be a standalone product that does not require any communication interfaces.

* 1. **Functional Requirements**

Functional requirements define the fundamental actions that system must perform.

The functional requirements for the system are divided into two main categories, Reservation/Booking, and Management. For further details, refer to the use cases.

1. Reservation/Booking
   1. The system shall record reservations.
   2. The system shall record the customer’s first name.
   3. The system shall record the customer’s last name.
   4. The system shall record the mobile number.
   5. The system shall record the room number.
   6. The system shall display the default room rate.
   7. The system shall record the customer’s email.
   8. The system shall generate a unique confirmation number for each reservation.
   9. The system shall automatically show the customers that have reservation after two days and send a reminder email to them
   10. The system shall record the expected check-in date.
   11. The system shall record the expected checkout date.
   12. The system shall allow reservations to be modified without having to reenter all the customer information.
2. Management
   1. The system shall allow for the addition of rooms.
   2. The system shall allow for the deletion of rooms.
   3. The system shall allow for the modification of rooms.
   4. **Nonfunctional Requirements**

Functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability, and portability.

* + 1. **Performance Requirements**

Performance requirements define acceptable response times for system functionality.

* The load time for user interface screens shall take no longer than two seconds.
* The log in information shall be verified within five seconds.
* Queries shall return results within five seconds.
  + 1. **Logical Database Requirements**

The logical database requirements include the retention of the following data elements.

**Booking/Reservation System**

* Customer first name
* Customer last name
* Customer address
* Customer phone number
* Assigned room
* Default room rate
* Credit card number
* Confirmation number
* Expected check-in date
* Expected check-out date
* Payment type
  + 1. **Design Constraints**

The Hotel Management System shall be a stand-alone system running in a Windows environment. The system shall be developed using Oracle and ASP.net.

* + 1. **Standards Compliance**

There shall be consistency in variable names within the system. The graphical user interface shall have a consistent look and feel.

* + 1. **Reliability**

Specify the factors required to establish the required reliability of the software system at time of delivery.

* + 1. **Availability**

The system shall be available during normal hotel operating hours.

* + 1. **Security**

Customer Service Representative will have access to the Reservation/Booking. Managers will be able to log in to the Hotel Management System and will have access to the Management subsystem.

* + 1. **Maintainability**

The Hotel Management System is being developed in ASP.net.Visual studio is one of the products that are used for building the system. This product is chose because it provides the user developers a compelling development environment for Windows and. NET platforms. This product helps in building Windows application, console application, Windows service, Windows mobile applications, ASP.NET application and ASP.NET web services with C++, C#, VB or J#. Also, you can add additional tools that help in building interface in easy way. For the end user, the ASP and VB is used in developing Hotel Management System. Also, it helps in creating Crystal report and connection with the database by using ODB. The other product that has been chosen is Oracle Database 10g Express Edition. It has been chosen for the background of the system. As it can build the tables and quires in easy way and connected with the end user in effective way.

* + 1. **Portability**

The Hotel Management System shall run in any Microsoft Windows environment.

**1.13 Use Case Diagram:**

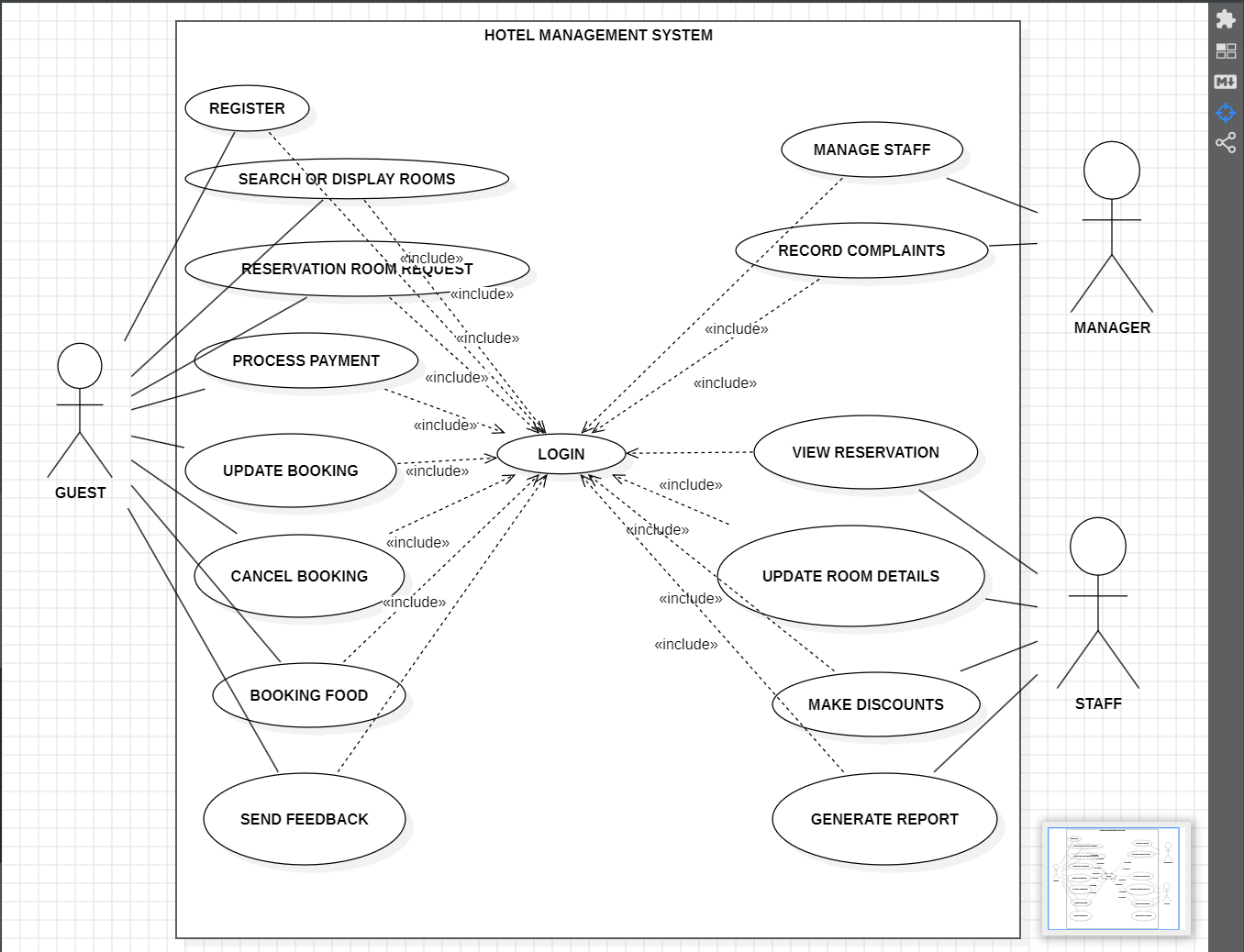


Fig 1. Use case diagram

 **Actor:** Guest

* Use cases:
  + Search or display rooms
  + Book a room
  + Update booking
  + Cancel booking
  + Make a payment
  + Send feedback

 **Actor:** Manager

* Use cases:
  + Register staff
  + Manage staff
  + Update room details
  + Process payment
  + View reservation
  + Generate report
  + Login

 **Included use cases:**

* Process payment use case includes:
  + Update booking use case
* Update booking use case includes:
  + Record complaints use case
* Login use case might include:
  + View reservation use case

**1.14 Sequence Diagram:**

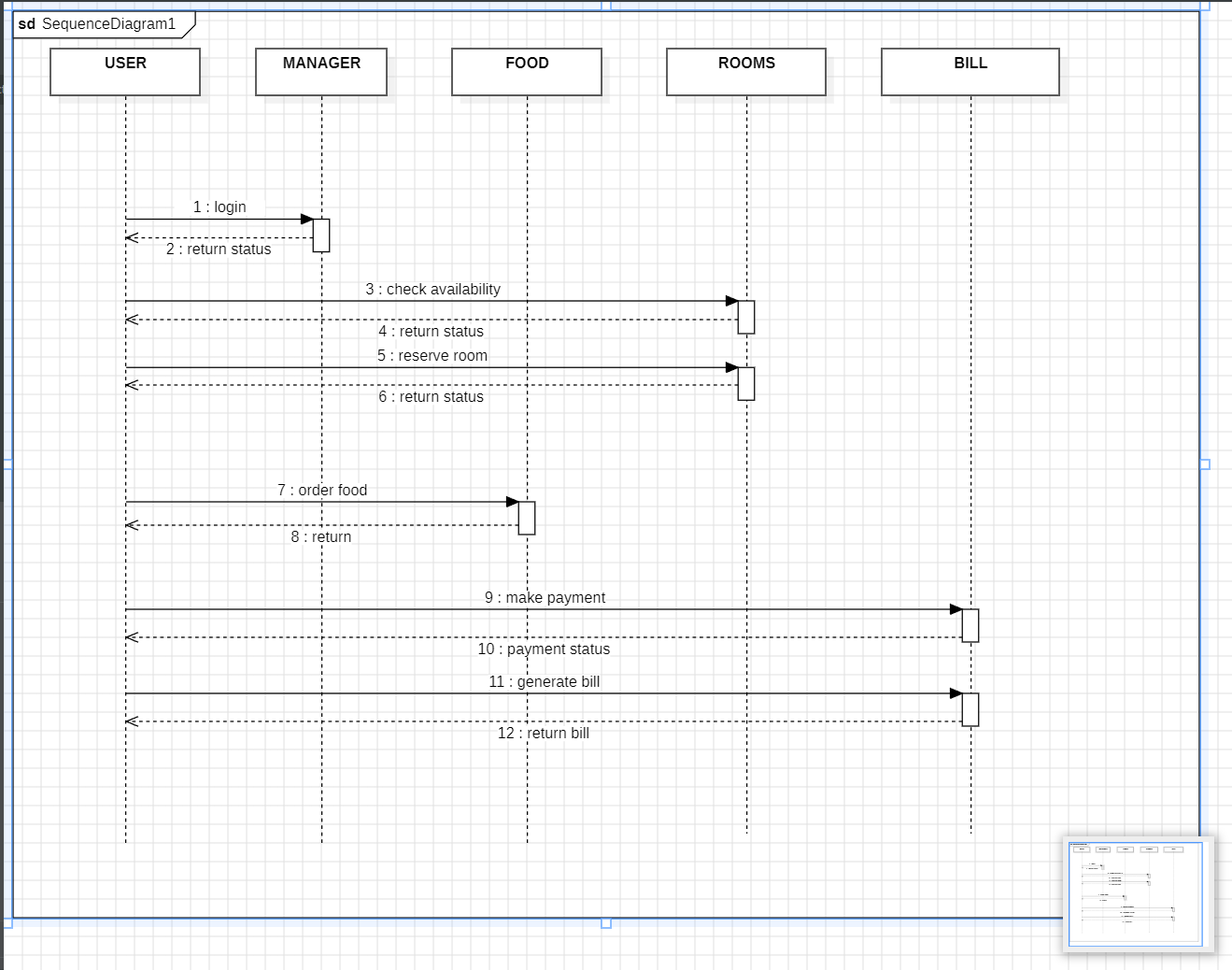


Fig 2. Sequence diagram

**Actors:**

* Guest
* System
* Manager (Partially shown)

**Use Case:** Booking a Room (This may be part of a larger use case like reserving a room)

**Sequence of Interactions:**

1. **Guest**: The guest initiates the interaction by searching for rooms through the HMS. This could involve specifying dates, room type, number of guests, and other preferences.
2. **System**: The HMS retrieves room availability information based on the guest's search criteria and presents it to the guest. This may include details like room rates, descriptions, and images.
3. **Guest**: The guest selects a room and proceeds to book it. This might involve entering additional information like contact details and payment method.
4. **System**: The HMS validates the guest's information and checks room availability for the selected dates.
5. **System** (conditional): If the room is available and the guest's information is valid, the system processes the reservation. This may involve:
   * Updating the room inventory to reflect the booked room.
   * Recording the guest's reservation details in the system database.
   * Sending a confirmation email or notification to the guest with their booking details.
6. **System** (conditional): If the room is unavailable or there are errors in the guest's information, the system informs the guest and allows them to modify their search or contact the hotel for assistance.

**1.15 Class Diagram:**

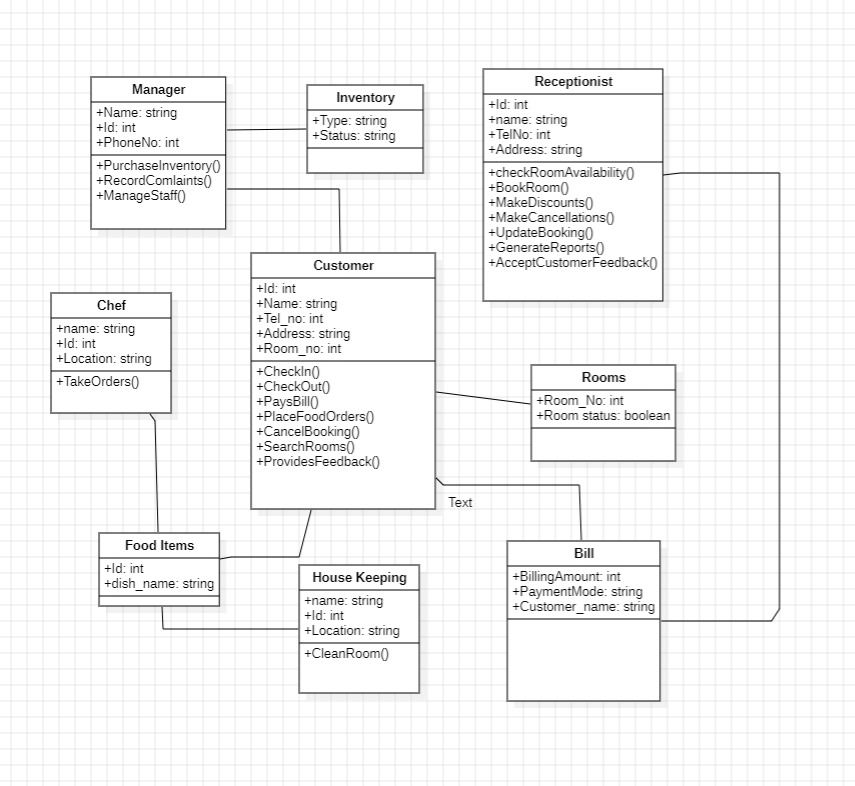


Fig 3. Class diagram

**Classes:**

* **Guest**
  + Attributes:
    - Guest ID
    - Name
    - Contact Information (e.g., email, phone number)
    - Address
    - Preferences (optional)
  + Methods:
    - Book a room
    - Update booking information
    - Cancel booking
    - Make a payment
    - Send feedback (optional)
* **Room**
  + Attributes:
    - Room ID
    - Room Type (e.g., standard, suite)
    - Capacity (number of guests)
    - Amenities (list of amenities offered in the room)
    - Rate (price per night)
    - Availability (available, booked, undergoing maintenance, etc.)
  + Methods:
    - Check availability
    - Book room (might be linked to the Guest class booking method)
    - Update room details (e.g., status, rate)
* **Reservation**
  + Attributes:
    - Reservation ID
    - Guest (linked to the Guest class)
    - Room (linked to the Room class)
    - Check-in date
    - Check-out date
    - Number of guests
    - Total cost
    - Status (e.g., confirmed, pending, cancelled)
  + Methods:
    - Update reservation details (e.g., dates, guests)
    - Cancel reservation

**Relationships:**

* **Guest** has a one-to-many relationship with **Reservation**. A guest can have multiple reservations, but a reservation is for one guest.
* **Room** has a one-to-many relationship with **Reservation**. A room can be booked in multiple reservations, but a reservation is for one room.
* **Reservation** is a combination of **Guest** and **Room**. It links a specific guest to a specific room for a particular stay.

**1.16 ER diagram:**

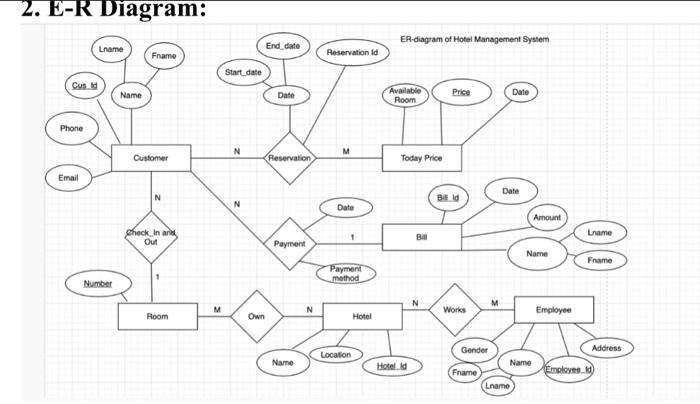


Fig 4. ER diagram

**Entities:**

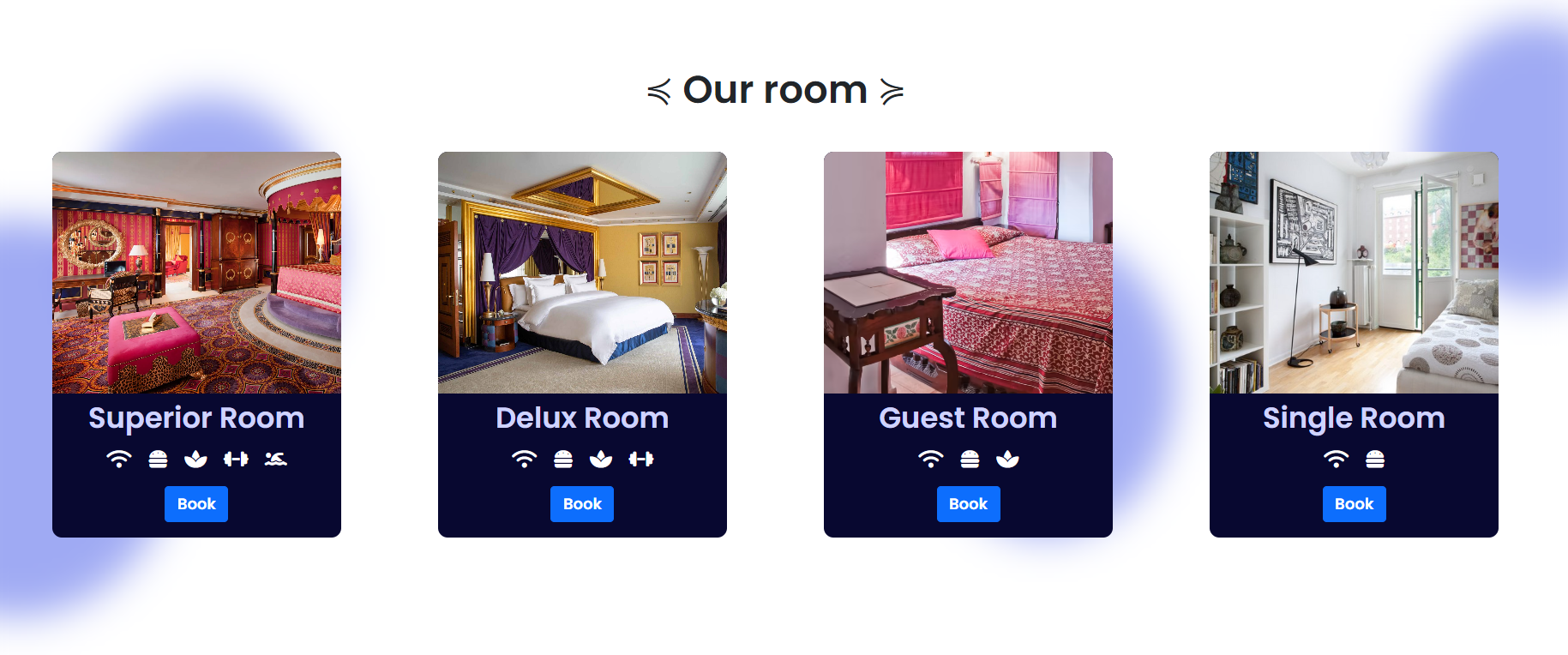
* **Guest:**
  + Attributes:
    - Guest\_ID (Primary Key)
    - Name
    - Contact\_Info (e.g., email, phone number)
    - Address
    - Preferences (optional)
* **Room:**
  + Attributes:
    - Room\_ID (Primary Key)
    - Room\_Type (e.g., standard, suite)
    - Capacity (number of guests)
    - Amenities (list of amenities offered in the room)
    - Price (rate per night)
* **Reservation:**
  + Attributes:
    - Reservation\_ID (Primary Key)
    - Check\_in\_Date
    - Check\_out\_Date
    - Number\_of\_Guests
    - Total\_Cost
    - Status (e.g., confirmed, pending, cancelled)
* **Payment:**
  + Attributes:
    - Payment\_ID (Primary Key)
    - Payment\_Method (e.g., credit card, cash)
    - Amount
    - Date

**Relationships:**

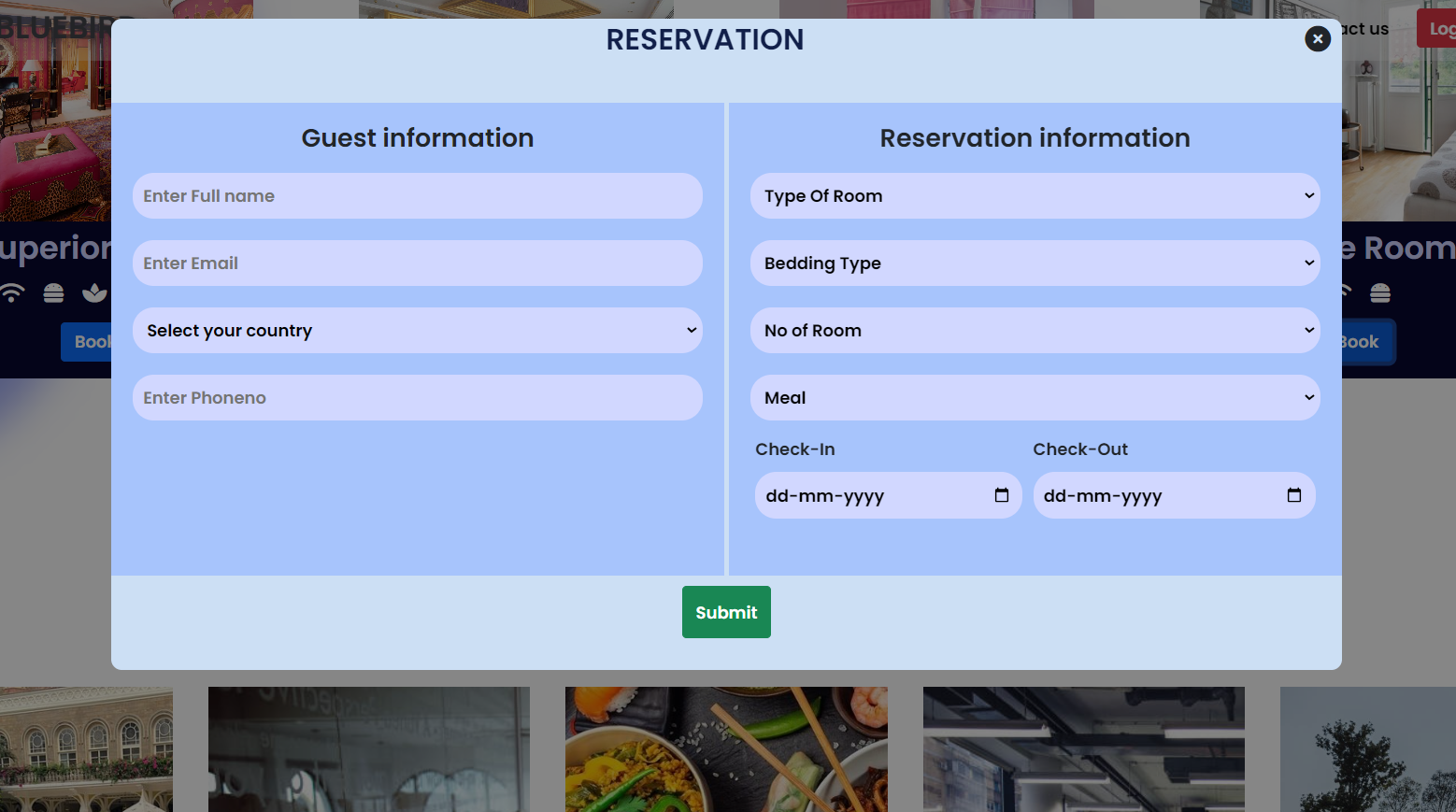
* **Guest** can have many **Reservations** (one-to-many). A guest can make multiple reservations over time. Each reservation, however, is for one guest.
* **Room** can have many **Reservations** (one-to-many). A room can be booked in multiple reservations on different dates. Each reservation, however, is for one room.
* **Reservation** is associated with one **Guest** (one-to-one). A reservation is for a single guest. Each reservation details who made the booking.
* **Reservation** is associated with one **Room** (one-to-one). A reservation is for a single room. Each reservation specifies which room is booked.
* **Reservation** can have one **Payment** (one-to-one). A reservation can have a single payment associated with it. Each payment is linked to a specific reservation.

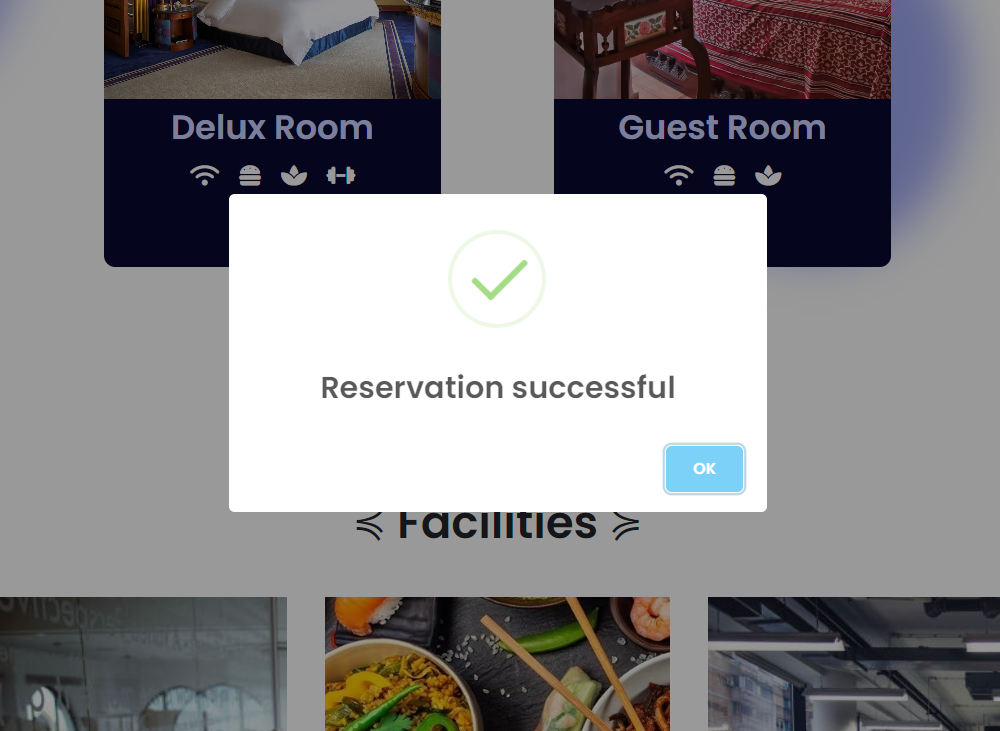
**Booking Management**

The booking management module is a critical component of the hotel management system. It encompasses the processes by which users make reservations and how these reservations are handled by the admin.



**Reservation**

****

****

**User Booking Process**

1. **User Registration and Login**
   * Users need to register by providing personal details such as name, email, and contact information.
   * Post-registration, users can log in using their credentials to access the booking system.
2. **Search and Select Rooms**
   * Users can search for available rooms based on various criteria such as dates, room type, and number of guests**.**
   * The system displays a list of available rooms that match the user’s search criteria.
   * Users can view detailed information about each room, including photos, amenities, and price per night.
3. **Make a Reservation**
   * After selecting a room, users proceed to the booking form where they provide additional details such as the number of guests, check-in and check-out dates, and any special requests.
   * Users confirm their booking details and proceed to the payment gateway to complete the reservation process.
   * Upon successful payment, a booking confirmation is generated and sent to the user via email and displayed on their account dashboard.

**Admin Booking Management:**

1. **Admin Login**
   * Admins log in using secure credentials to access the backend of the hotel management system.
2. **View and Manage Bookings**
   * Admins can view all incoming bookings through an intuitive dashboard that displays booking details such as user information, room details, booking dates, and payment status.
   * Each booking is marked with a status (e.g., pending, confirmed, cancelled).
3. **Accept or Reject Bookings**
   * the booking based on availability and hotel policies**.** Admins review the details of each booking and have the authority to accept or reject
   * If a booking is accepted, the status is updated to confirmed, and the user is notified via email and their account dashboard**.**
   * If a booking is rejected, the user is notified with a reason for the rejection, and any payment made is processed for refund.
4. **Manage Room Inventory**
   * Admins can update room availability, pricing, and details to reflect current inventory status.
   * This ensures that the booking system provides up-to-date information to users.
5. **Staff Registration and Roles**
   * Admins can register new staff members by inputting their personal information, contact details, and job roles.
   * Assign specific roles and permissions to staff members based on their responsibilities (e.g., housekeeping, front desk, maintenance).

**Actors in the Use Case Diagram**

In a hotel management system, various actors interact with the system to perform specific functions. These actors include users, receptionist, staff, manager, and software developer. Each actor has distinct roles and responsibilities that contribute to the overall functioning of the system.

**1. Users**

**Role:** Users are the guests or customers who interact with the hotel management system primarily for booking rooms and availing other services offered by the hotel.

**Responsibilities:**

* **Registration and Login:** Users create an account and log in to access the system.
* **Search and Book Rooms:** Users search for available rooms based on their preferences and make reservations.
* **Manage Reservations:** Users can view, modify, or cancel their bookings.
* **Make Payments:** Users complete the payment process for their bookings.
* **Provide Feedback:** Users can leave reviews and feedback about their stay and the services received.

**2. Receptionist**

**Role:** The receptionist is responsible for front desk operations and acts as an intermediary between the users and the hotel management system.

**Responsibilities:**

* **Check-in/Check-out Process:** Assist users during the check-in and check-out processes.
* **Manage Reservations:** View and manage room reservations, including modifications and cancellations.
* **Handle Inquiries:** Respond to user inquiries and provide information about hotel services and facilities.
* **Billing and Payments:** Manage billing processes and handle user payments at the front desk.
* **Customer Support:** Address user complaints and provide assistance as needed.

**3. Staff**

**Role:** Hotel staff includes various employees such as housekeeping, maintenance, and room service personnel who ensure smooth hotel operations and guest satisfaction.

**Responsibilities:**

* **Housekeeping:** Clean and maintain guest rooms and common areas.
* **Maintenance:** Perform routine maintenance and address repair requests to ensure the hotel is in good condition.
* **Room Service:** Provide food and beverage services to guests in their rooms.
* **Task Management:** Complete tasks assigned through the hotel management system and update their status upon completion.
* **Inventory Management:** Manage and report inventory levels of housekeeping and maintenance supplies.

**4. Manager**

**Role:** The manager oversees the overall operations of the hotel, ensuring that all departments function efficiently and guests receive high-quality service.

**Responsibilities:**

* **Staff Management:** Oversee staff performance, schedule shifts, and handle hiring and training processes.
* **Monitor Operations:** Ensure that all hotel operations are running smoothly and efficiently.
* **Handle Complaints:** Address and resolve guest complaints and issues.
* **Financial Management:** Monitor hotel finances, including budgeting, billing, and revenue management.
* **Reporting:** Generate and review reports on hotel performance, occupancy rates, and financial status.

**5. Software Developer**

**Role:** The software developer is responsible for designing, developing, and maintaining the hotel management system.

**Responsibilities:**

* **System Development:** Design and code the hotel management system based on the specified requirements.
* **Database Management:** Develop and manage the database that stores all hotel-related information.
* **Feature Implementation:** Add new features and functionalities to enhance the system.
* **System Maintenance:** Perform regular maintenance and updates to ensure the system runs smoothly and securely.
* **Technical Support:** Provide technical support to resolve any system issues and assist users in troubleshooting problems.
* **Documentation:** Create and maintain comprehensive documentation for the system, including user manuals and technical guides.

**Conclusion**

1. The successful implementation of the hotel management system marks a significant milestone in enhancing the operational efficiency and guest experience within the hotel industry. The system effectively integrates various core functionalities such as booking management, payment processing, and staff management, ensuring a seamless and user-friendly experience for all stakeholders involved.
2. Users benefit from an intuitive and streamlined booking process, from room selection to payment confirmation. The ability to manage reservations and receive timely notifications enhances their overall experience and satisfaction. For the receptionist and hotel staff, the system simplifies daily operations by automating routine tasks, managing room inventory, and facilitating effective communication and task assignments.
3. The manager, overseeing the entire operation, now has access to comprehensive tools for staff management, financial oversight, and performance reporting. This allows for more informed decision-making and ensures that all aspects of hotel management are conducted efficiently. The integration of secure payment gateways and compliance with industry standards also assures guests of a safe and reliable transaction process.
4. Moreover, the software developer's role in maintaining and enhancing the system ensures that it remains up-to-date with the latest technological advancements and user needs. The continuous improvement cycle helps in addressing any potential issues promptly and adding new features that further improve system functionality.
5. Overall, the hotel management system has proven to be a valuable asset in modernizing hotel operations. It bridges the gap between traditional hospitality practices and contemporary technological solutions, fostering a more productive environment for staff and a more enjoyable stay for guests. With this robust system in place, the hotel is well-equipped to handle current demands and future growth, maintaining a high standard of service and operational excellence.

| **Test Case ID** | **Test Case Description** | **Priority** | **Preconditions** | **Input Test Data** | **Steps to be Executed** | **Expected Results** | **Actual Results** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Test if user is able to login successfully. | A | User must be registered already | correct username, correct password | 1) Enter correct username and password in the respective fields. 2) Click submit/login. | User must successfully login to the HMS dashboard. | User is logged in. |
| 2 | Test if unregistered users are not able to login to the HMS. | A | - | incorrect username, incorrect password | 1) Enter incorrect username and password in the respective fields. 2) Click submit/login. | Proper error must be displayed and prompt to enter login again. | Displays error and prompts to enter login credentials again. |
| 3 | Test with valid username and empty password such that login must fail. | B | User must be registered already | valid username and empty password | 1) Enter valid username in the username field and leave the password field empty. | Proper error must be displayed and prompt to enter login again. | Displays error and prompts to enter login credentials again. |
| 4 | Test with empty username and valid password such that login must fail. | B | Registered user's password | empty username and valid password | 1) Leave the username field empty and enter a valid user's password in the password field. | Proper error must be displayed and prompt to enter login again. | Displays error and prompts to enter login credentials again. |
| 5 | Test with empty username and empty password and check if login fails. | A | - | - | 1) Leave both the username and password fields empty. 2) Click submit. | Proper error must be displayed and prompt to enter login again. | Displays error and prompts to enter login credentials again. |
| 6 | Check if the password is masked on the screen (e.g., displayed as bullets or asterisks). | B | - | some passwords (registered or unregistered) | 1) Enter the password field with some characters. | The password field should display the characters as asterisks or bullets. | Displays the password in masked form. |
| 7 | Check if the login function handles case sensitivity. | B | Registered user's password which is originally in lower case | case-changed username/password | 1) Enter the case-changed username/password in the respective field. 2) Click login button. | Login must fail with an incorrect username/password error. | Displays error and prompts to enter login credentials again. |
| 8 | After logging in, try to copy/cut the password and paste it on another screen (passwords are usually masked). | B | - | Registered user's login ID and password | 1) Enter username and password in the respective fields. 2) Copy the password field's content. 3) Paste the content on another screen. | The password shouldn’t get pasted or should not be visible. | Displays blank password field. |
| 9 | Verify account lock after multiple failed login attempts. | B | Registered user's login ID and incorrect password | - | 1) Try to login with a registered username and incorrect password more than 3 times. | Account should be locked, and access should be granted only after certain verification. | Account gets locked. |
| 10 | Check if selecting the back button (after logging out) does not sign the user back in. | B | Registered username and password | - | 1) Login with registered username and password. 2) Sign out. 3) Press back button. | User shouldn’t be signed in, and a general webpage must be visible. | Displays a general page. |
| 11 | Verify the URL access without logging into the HMS. | B | Registered username and password | - | 1) Login using registered username and password. 2) Copy and save the URL of the logged-in page. 3) Logout. 4) Paste the copied URL in the browser. | The URL should redirect to a logged-out page. | Displays the logged-out page. |
| 12 | Automatic logout when pressing backspace button. | B | User must be registered already | Registered username and password | 1) Login using registered username and password. 2) Press backspace. | User must logout of the site properly. | User is logged out of the site. |

**Testing**

The key purpose of a test case is to ensure that different features within the hotel management system (HMS) are working as expected. Test cases help validate that the software is free of defects and meets the expectations of the end users. The benefits of test cases include:

* Ensuring good test coverage.
* Improving the quality of software.
* Decreasing maintenance and software support costs.

Verifying that the software meets the end user requirements.

* Encouraging testers to think thoroughly and approach the tests from various angles.
* Reusability for future reference and execution.

Above is a detailed table of test cases designed to validate the core functionalities of the HMS project. Each test case includes a unique identifier, description, priority, preconditions, input test data, steps to be executed, expected results, and actual results.

**References**

1. **Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education.**
2. **Sommerville, I. (2015). Software Engineering (10th ed.). Pearson.**
3. **Kaner, C., Falk, J., & Nguyen, H. Q. (1999). Testing Computer Software (2nd ed.). Wiley.**
4. **Hunt, A., & Thomas, D. (1999). The Pragmatic Programmer: Your Journey to Mastery. Addison-Wesley Professional.**
5. **IEEE. (2008). IEEE Standard for Software and System Test Documentation. IEEE Std 829-2008.**
6. **Davis, G. B., & Olson, M. H. (1985). Management Information Systems: Conceptual Foundations, Structure, and Development (2nd ed.). McGraw-Hill.**
7. **Turban, E., Sharda, R., & Delen, D. (2010). Decision Support and Business Intelligence Systems (9th ed.). Pearson.**
8. **Kumar, S. A., & Suresh, N. (2009). Operations Management. New Age International.**
9. **Garg, A. (2014). Hotel Management and Operations. Oxford University Press.**
10. **Bhasin, H. (2018). Business Analysis: Techniques for Hotel Management Systems. Business Analysis Publications.**