A group of women with luggage in a hotel lobby

Description automatically generated

**Hotel Management System**

**Group 4**

**Apekshya , Kuwar , Suyang**

Contents

[Executive Summary 2](#_Toc127454286)

[Design Analysis Process 2](#_Toc127454287)

[Entity Relationship Diagram 9](#_Toc127454288)

[Table Designs – Data Dictionary 10](#_Toc127454289)

[Contributions 11](#_Toc127454290)

# **Executive Summary**

This Hotel Management System is a database-driven application designed to streamline daily hotel operations, such as room booking, customer information management, room type management, and payment processing. By organizing its functionality into well-defined tables (e.g., Customer, Room, Room Type, and Reservations) and automating workflows, the system provides an efficient and reliable solution for managing various hotel activities.

# **Design Analysis Process**

**User Stories for the System**

Booking Module

1. As a customer, I want to view real-time room availability so that I can choose a room that fits my budget and preferences.
2. As a customer, I want to book a room online so that I can secure my stay without visiting the hotel.
3. As a customer, I want to receive booking confirmation and check-in reminders so that I can plan my trip without forgetting important details.
4. As a hotel manager, I want to monitor current reservations so that I can manage room availability and avoid overbooking.
5. As a hotel manager, I want to update room availability dynamically so that the system reflects accurate data for customers.

Check-in Module

1. As a customer, I want to check in using my online booking details so that I can avoid delays at the reception.
2. As a customer, I want to receive a check-in guide so that I can easily navigate the process.
3. As a hotel staff member, I want to access a guest's booking details so that I can quickly complete the check-in process.
4. As a hotel manager, I want the system to notify staff of upcoming check-ins so that preparations can be made in advance.

**Functional and non-functional requirements**

1. User Management and Registration:   
Guests can create an account by registering, which includes payment information and personal data.   
Guest Login: A secure login for visitors that offers social networking or email/password login choices.   
User Profile Management: Visitors have the ability to edit contact details, save preferences, view booking history, and manage their profile.   
Role-based access control is the process of granting administrators, employees, and visitors distinct access to handle reservations, inventory, or visitor information.

2. Hotel Information Management:

Details about the hotel and rooms Store information on rooms (e.g., type, price, number of beds, description, availability) and hotels (e.g., location, star rating, amenities).   
Management of Room Availability: Monitor the availability of rooms in real time to prevent double booking.   
Exclusive Deals and Promotions: the capacity to store seasonal pricing data, promotions, and discount codes.

3. Reservation and Booking System:   
Search Functionality: Using parameters like location, cost, kind of room, dates of check-in and check-out, and amenities, guests can look for available rooms.   
Reservation Management: Visitors have the option of making instant reservations for particular dates or using a "hold" system, which places a reservation on hold until payment is received.   
Booking Confirmation: The system provides a confirmation email and/or text message with reservation details following a successful booking.   
Cancellation of Reservations: According to established guidelines, users ought should be able to cancel reservations (e.g., free cancellation within 24 hours).   
Booking Modification: Customers ought to have the option to alter their reservation, including the guest's information, dates, or room.

4. Payment Integration: Payment Gateway Integration: Secure payment processing by integration with third-party payment providers (such as PayPal and Stripe).   
Receipts & Payment History: Visitors can examine previous payments, obtain receipts, and verify the status of payments (e.g., reimbursed, pending, or finished).   
Deposits and Partial Payments: The system ought to allow for the collection of deposits or partial payments at the time of reservation, with the remaining amount being payable during check-in.   
5. Alerts & Notifications:   
Reminders for Bookings: Prior to check-in, visitors receive email or SMS messages (such as reminders about check-in time, room preparation, etc.).   
Cancellation Alerts: If a reservation is canceled, either by the user or the system, notifications ought to be provided.   
Payment Status Alerts: Notify recipients via email or SMS if a payment was successful or unsuccessful.   
Upselling and Special Offers: Inform visitors about exclusive offers, loyalty programs, or room upgrades available during this stay.

4. Payment Integration: Payment Gateway Integration: Secure payment processing by integration with third-party payment providers (such as PayPal and Stripe).   
Receipts & Payment History: Visitors can examine previous payments, obtain receipts, and verify the status of payments (e.g., reimbursed, pending, or finished).   
Deposits and Partial Payments: The system ought to allow for the collection of deposits or partial payments at the time of reservation, with the remaining amount being payable during check-in.   
5. Alerts & Notifications:   
Reminders for Bookings: Prior to check-in, visitors receive email or SMS messages (such as reminders about check-in time, room preparation, etc.).   
Cancellation Alerts: If a reservation is canceled, either by the user or the system, notifications ought to be provided.   
Payment Status Alerts: Notify recipients via email or SMS if a payment was successful or unsuccessful.   
Upselling and Special Offers: Inform visitors about exclusive offers, loyalty programs, or room upgrades available during this stay.

**Non-Functional Requirements**

1.Performance and Scalability: Quick Search and Response Time: Even with a high load, the system should react to room searches and booking activities in two to three seconds.   
Scalable Infrastructure: Especially during busy times (holidays, promotions), the database and server architecture should be able to accommodate growing user and transaction volumes.   
High Availability: 99.9% uptime is required for the system to guarantee that visitors can make or change reservations at any time.

2. Privacy and Security of Data:   
Secure Data Storage: Encryption technologies like AES-256 and secure password hashing (like bcrypt) should be used to safely store sensitive data, such as payment information and personal information.   
Legal Regulation Compliance: When managing personal and payment data, the system must abide by the CCPA, GDPR, and PCI-DSS.   
Use two-factor authentication (2FA) for admin logins and visitor accounts.

3. Reliability and Fault Tolerance: Recovery and Backup: In the case of a system failure, daily database backups and disaster recovery protocols should guarantee little data loss.   
Fault Detection: With auto-recovery and administrator notifications, the system should be able to identify and manage failures with grace.   
Using automatic session timeouts for security, session management makes ensuring that user sessions are appropriately preserved across interactions.   
4. Accessibility and Usability:   
Intuitive UI/UX: To cut down on time spent on tasks (such as making a reservation or browsing reservations), the system should offer a user-friendly interface for both visitors and hotel employees.   
Mobile Accessibility: To ensure a smooth experience on smartphones and tablets, the system should be responsive and mobile device optimized.   
Compliance with Accessibility: WCAG (Web Content Accessibility) should be followed by the system.

5.Extensibility and Maintainability: Modular Design: Simple updates, feature additions (like a loyalty program), and connections with external tools are made possible by a modular system architecture.   
Monitoring and Logging: Detailed logs should document transactions, problems, and system behavior. Real-time system performance monitoring should be possible for administrators.   
Automated Testing: For essential elements (such as the booking flow and payment processing), the system should have automated testing processes.

6. Localization and Internationalization:   
Support for Multiple Currencies: Depending on the user's location, the system ought to enable payments in various currencies.   
Support for Multiple Languages: For visitors from abroad, the system should be able to communicate in English, Spanish, and French, among other languages.   
Timezone Handling: The system should appropriately handle reservations made in different time zones, converting check-in and check-out hours according to the user's location.

7.Data Integrity and Consistency: ACID Compliance: To guarantee transaction integrity, especially for booking, payment, and cancellation procedures, the database should adhere to ACID (Atomicity, Consistency, Isolation, Durability) standards.   
Uniformity Among Systems: Make sure that the database, backend processes, and front-end user interface are all consistent to avoid inconsistencies like double booking or inaccurate billing.

8. Resource Optimization and Cost Efficiency:   
Effective Resource Usage: To cut down on resource usage (such as CPU and memory) and hosting expenses, optimize database queries and backend operations.   
Cloud Scalability: If the system is hosted in the cloud, elastic scaling should be used to modify resources in response to demand, guaranteeing cost effectiveness during off-peak hours.

**User cases**

**1.Login to System**

* **Actor:** Customer
* **Description:** A customer logs into the system using their credentials.
* **Precondition:** Customer has a registered account.
* **Postcondition:** Successful login redirects to the booking dashboard; unsuccessful login shows an error message.

**2.Verify Password**

* **Actor:** System
* **Description:** The system validates the entered password against the stored credentials.
* **Precondition:** Customer enters their username and password.
* **Postcondition:** Grant or deny access based on validation.

**3.Display Calendar**

* **Actor:** Customer
* **Description:** After logging in, the system displays a calendar for selecting booking dates.
* **Precondition:** User is logged in.
* **Postcondition:** Calendar interface is shown.

**4.Select Booking Dates**

* **Actor:** Customer
* **Description:** The customer selects check-in and check-out dates for their reservation.
* **Precondition:** Customer accesses the calendar.
* **Postcondition:** Selected dates are recorded.

**5.Display Available Rooms**

* **Actor:** System
* **Description:** The system displays a list of available rooms based on the selected dates.
* **Precondition:** Customer has selected dates.
* **Postcondition:** Available room types and their prices are shown.

**6.Select Room Type**

* **Actor:** Customer
* **Description:** The customer selects a room type from the list of available options.
* **Precondition:** System displays available room types.
* **Postcondition:** The selected room type is recorded for the reservation.

**7.Confirm Reservation Information**

* **Actor:** Customer
* **Description:** Customer reviews and confirms their reservation details, including dates, room type, and price.
* **Precondition:** Customer has selected a room type.
* **Postcondition:** Reservation is marked as confirmed.

**8.Input Contact Information**

* **Actor:** Customer
* **Description:** Customer inputs their contact details, including name, email, and phone number.
* **Precondition:** Customer is confirming reservation details.
* **Postcondition:** Contact information is saved in the system.

**9.Make Payment**

* **Actor:** Customer
* **Description:** The customer makes a payment for their reservation.
* **Precondition:** Reservation details are confirmed.
* **Postcondition:** Payment status is updated as successful or failed.

**10.Handle Payment Failure**

* **Actor:** System
* **Description:** If a payment fails, the system shows an error message and suggests retrying the payment process.
* **Precondition:** Payment attempt failed.
* **Postcondition:** Customer is prompted to retry or use a different payment method.

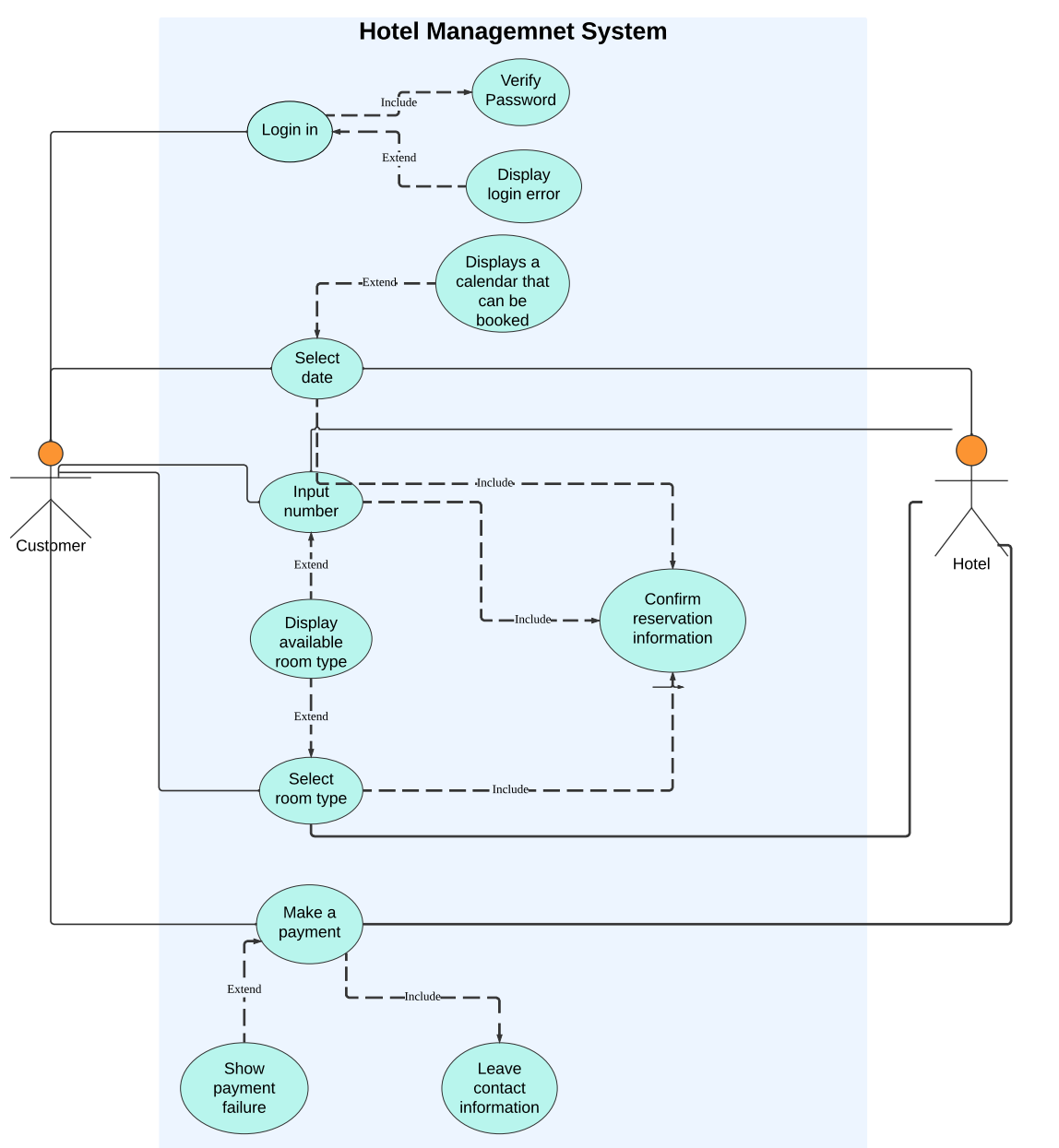
**11.Cancel Reservation**

* **Actor:** Customer
* **Description:** Customer cancels their reservation before the check-in date.
* **Precondition:** Reservation exists.
* **Postcondition:** Reservation is marked as canceled, and the room is made available.

**12.Update Customer Information**

* **Actor:** Customer
* **Description:** Customer updates their contact information, such as phone number or email.
* **Precondition:** Customer has an existing account.
* **Postcondition:** Updated information is saved in the system.

**use case diagrams**

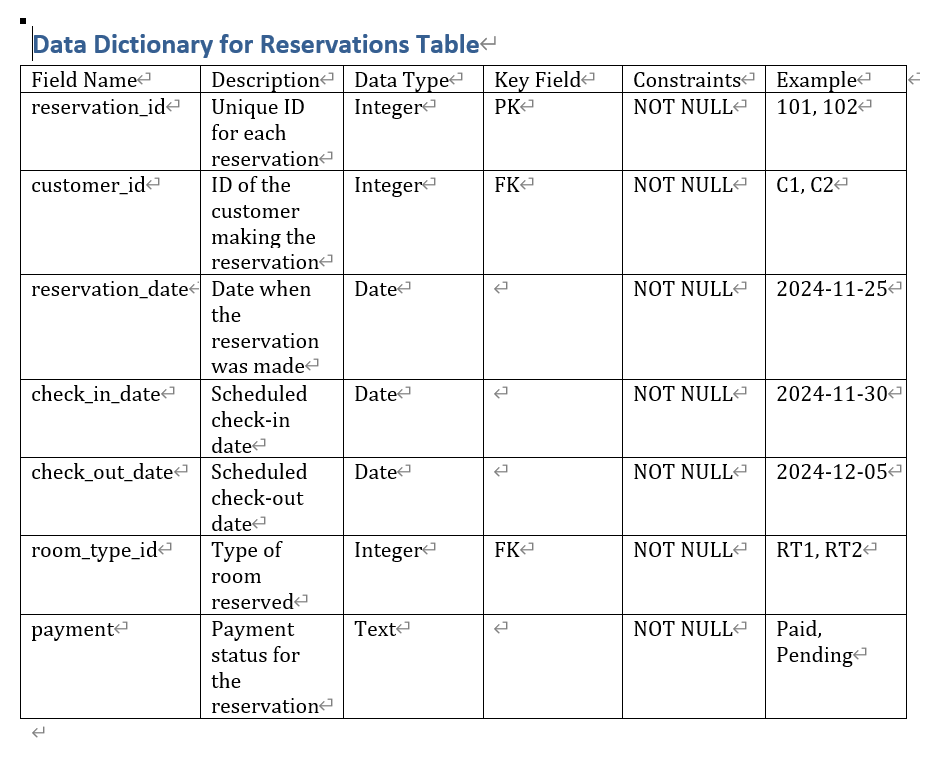


# **Entity Relationship Diagram**

A diagram of a room type

Description automatically generated

# **Table Designs – Data Dictionary**



A white sheet with black text

Description automatically generated

A screenshot of a data dictionary

Description automatically generated

# **Contributions**

**Group number：**

Apekshya , Kuwar , Suyang