Hotel Reservation System

- Requirements & Documentation

1. Overview:

The Hotel Reservation System is an application that allows users to search, view, and book rooms at a hotel. The system should also provide functionality for hotel administrators to manage room availability, view bookings, and process payments. The system should be intuitive, secure, and scalable for both small hotels and larger hotel chains.

---

2. Functional Requirements:

2.1. User Features:

1. User Registration and Authentication:

Users can register an account with their personal details (name, email, phone number).

Users can log in with their registered email and password.

Password recovery via email in case users forget their credentials.

2. Room Search:

Users can search for rooms based on the check-in and check-out dates, room type, and the number of guests.

Search results should display room availability, prices, and amenities.

3. Room Details:

Users can view detailed information about a specific room, including images, amenities, price per night, and room type (e.g., single, double, suite).

The room page will show available dates and pricing.

4. Room Booking:

Users can book a room by selecting the check-in/check-out dates, number of guests, and any special requests.

The system should show an overview of the selected room, check-in and check-out dates, and total cost.

Users can proceed to payment after booking the room.

5. Payment Gateway:

Integration with a payment gateway (e.g., Stripe, PayPal) to process payments securely.

Users will enter their payment details (credit card, debit card, or other online payment methods) for booking confirmation.

Users receive booking confirmation via email and/or SMS.

6. Booking History:

Registered users can view a history of past bookings.

Users can cancel or modify bookings within a defined period before the check-in date (as per the hotel’s policy).

7. User Profile Management:

Users can update their personal information such as name, phone number, and email address.

Users can change their password.

2.2. Admin Features:

1. Admin Authentication:

Hotel administrators can log in to the system using an admin-specific account with higher privileges.

2. Room Management:

Admins can add, update, and delete room types, pricing, availability, and descriptions.

Admins can upload images and amenities for each room.

Admins can set room availability (e.g., blocking dates for maintenance).

3. Booking Management:

Admins can view all customer bookings (upcoming and past).

Admins can modify booking details, such as changing dates or room assignments.

Admins can cancel a booking (with applicable fees).

4. Payment and Invoice Management:

Admins can view payments made for bookings and generate invoices.

Admins can set payment policies, such as advance payments or cancellations.

5. Reporting:

Admins can generate reports on bookings, revenue, occupancy rates, and customer demographics.

Generate reports on cancellation rates, booking trends, and popular room types.

6. Customer Management:

Admins can view customer details and communicate with customers if necessary.

Admins can manage customer feedback and complaints.

2.3. Additional Features:

1. Multi-language and Currency Support:

The system should support multiple languages (e.g., English, Spanish, French) and multiple currencies (USD, EUR, etc.).

2. Notifications:

Email or SMS notifications for booking confirmation, cancellations, reminders, and special offers.

Admins should be able to send promotions or special offers to users via email.

3. Room Review System:

After a stay, users can rate the room, service, and overall hotel experience.

Admins can moderate reviews to ensure they are appropriate.

4. Discounts and Promotions:

The system should allow admins to create discount codes for promotions.

Discounts could be applied during booking checkout or as part of special seasonal offers.

---

3. Non-Functional Requirements:

1. Performance:

The system must handle high traffic, especially during peak times (holidays, weekends).

Response time for the user interface should be under 3 seconds for an optimal user experience.

2. Scalability:

The system should scale with increasing users and bookings, allowing for more hotels and users to join the platform.

It should support both small and large hotels.

3. Security:

Secure authentication (password encryption, two-factor authentication for admins).

Payment data should be securely handled via HTTPS and PCI-DSS compliance.

Sensitive user data (personal details) should be encrypted in the database.

4. Reliability:

The system must be highly available with minimal downtime.

Data backups should be taken regularly to prevent data loss.

5. User Experience (UX):

The system should have a responsive design, working on desktops, tablets, and smartphones.

The UI should be user-friendly, easy to navigate, and visually appealing.

6. Maintenance:

The system should be easy to maintain, with clear documentation for any future updates or fixes.

Admins should have an intuitive dashboard for managing the system.

---

4. Database Design:

Entities:

1. Users:

Attributes: UserID, Name, Email, Phone, Password (hashed), Role (admin/user), Address, Profile Picture.

2. Rooms:

Attributes: RoomID, RoomNumber, RoomType, Price, Description, Image, Amenities, MaxGuests, Availability (booked/free), HotelID.

3. Bookings:

Attributes: BookingID, UserID, RoomID, CheckInDate, CheckOutDate, Status (confirmed, cancelled), TotalPrice, PaymentStatus (paid/unpaid).

4. Payments:

Attributes: PaymentID, BookingID, Amount, PaymentMethod, PaymentStatus, PaymentDate.

5. Reviews:

Attributes: ReviewID, BookingID, Rating, Comment, Date.

6. Discounts:

Attributes: DiscountID, Code, DiscountPercentage, ExpiryDate, Status (active/inactive).

---

5. Technology Stack:

1. Frontend:

HTML5, CSS3, JavaScript (React.js or Angular)

Responsive design using frameworks like Bootstrap.

2. Backend:

ASP.NET Core, Django, or Spring Boot (Java) for handling business logic and database communication.

RESTful API for frontend-backend interaction.

3. Database:

Relational database like MySQL, PostgreSQL, or SQL Server.

4. Payment Gateway Integration:

Stripe or PayPal API for handling payments securely.

5. Authentication:

JWT (JSON Web Tokens) or OAuth for secure login and session management.

6. Hosting/Deployment:

Cloud platforms like AWS, Azure, or Heroku for deployment.

Continuous integration/continuous deployment (CI/CD) pipelines for updates and maintenance.

--

6. Future Enhancements:

1. Mobile Application:

Develop mobile apps for iOS and Android for better user engagement and management.

2. Integration with Other Travel Services:

Integration with flight booking or car rental services to offer complete travel packages.

3. AI-Powered Recommendations:

Use machine learning to recommend rooms based on user preferences, past bookings, or trending destinations.

4. Multi-Hotel Platform:

Allow users to book rooms across different hotel chains or locations in one platform, creating a network of hotels.

---

This documentation outlines the key requirements and functional specifications for developing a Hotel Reservation System that can handle user and admin operations, ensure security, provide excellent user experience, and be scalable for growth.