1. Introduction

Project Overview: The Hotel Management System (HMS) website is a comprehensive platform designed to manage hotel operations, including room bookings, customer management, staff management, payments, and reports. This system aims to streamline hotel operations, improve customer satisfaction, and enhance the overall efficiency of hotel management.

Objectives:

- Simplify the booking process for customers.
- Provide efficient management tools for hotel staff.
- Maintain records of all hotel activities.
- Generate reports for analysis and decision-making.

2. System Requirements

2.1 Hardware Requirements:

- Web Server: Apache, Nginx, or any equivalent
- Database Server: MySQL, PostgreSQL, or equivalent
- Client Devices: Desktop, Laptop, Tablet, or Mobile with a modern web browser

2.2 Software Requirements:

- Programming Language: PHP, Java, Python, or any web-based language
- Database: MySQL, PostgreSQL, or equivalent
- Frontend: HTML5, CSS3, JavaScript, Bootstrap, React, or Angular
- Backend: Django, Laravel, Spring Boot, or Node.js
- Version Control: Git

2.3 Security Requirements:

- SSL for data encryption
- User authentication (OAuth, JWT)
- Role-based access control

3. System Design

3.1 System Architecture: The system follows a three-tier architecture:

- **Presentation Layer:** User Interface

- **Business Logic Layer:** Application services, user authentication, session management
- **Data Layer:** Database management, CRUD operations

3.2 User Roles:

- **Admin:** Full access to all features and settings.
- **Manager:** Access to daily operations, staff management, and reporting.
- **Staff:** Access to specific operational tasks (e.g., room service, check-ins).
- **Customer:** Access to booking rooms, viewing reservations, and making payments.

4. Entities and Screens

4.1 Entities:

1. **User**
   - Attributes: UserID, Name, Email, Password, Role, ContactInfo
   - Description: Stores user information for authentication and role management.
2. **Customer**
   - Attributes: CustomerID, Name, Email, Phone, Address, BookingHistory
   - Description: Contains details about the customers who book rooms.
3. **Room**
   - Attributes: RoomID, RoomNumber, RoomType, Status (Available/Occupied), Price
   - Description: Defines the rooms in the hotel, their type, status, and pricing.
4. **Booking**
   - Attributes: BookingID, CustomerID, RoomID, BookingDate, CheckInDate, CheckOutDate, Status
   - Description: Captures all details of a room booking made by a customer.
5. **Payment**
   - Attributes: PaymentID, BookingID, Amount, PaymentDate, PaymentMethod, Status
   - Description: Tracks payments made by customers for their bookings.
6. **Staff**
   - Attributes: StaffID, Name, Role, Department, ContactInfo, ShiftTiming
   - Description: Stores information about hotel staff members.
7. **Service**
   - Attributes: ServiceID, ServiceType, Description, Price
   - Description: Details additional services offered by the hotel (e.g., room service, laundry).

8. **ServiceRequest**
   - Attributes: RequestID, ServiceID, CustomerID, RoomID, RequestDate, Status
   - Description: Records requests made by customers for services during their stay.
9. **Inventory**
   - Attributes: InventoryID, ItemName, Quantity, Description
   - Description: Manages hotel inventory items (e.g., toiletries, linens).
10. **Report**
    - Attributes: ReportID, ReportType, GeneratedDate, Content
    - Description: Represents various reports generated for analysis.

4.2 Screens:

1. **Login Screen**
   - Entities: User
   - Description: A screen for users to authenticate and log into the system.
2. **Dashboard Screen**
   - Entities: User, Booking, Room, Payment
   - Description: Overview of the hotel's key metrics, accessible by admin and manager.
3. **Customer Management Screen**
   - Entities: Customer, Booking
   - Description: Allows management of customer details and viewing of their booking history.
4. **Room Management Screen**
   - Entities: Room, Booking
   - Description: Enables management of room details, including availability and pricing.
5. **Booking Management Screen**
   - Entities: Booking, Room, Customer, Payment
   - Description: Interface for handling new bookings, check-ins, and check-outs.
6. **Payment Management Screen**
   - Entities: Payment, Booking
   - Description: Tracks and manages customer payments, including pending and completed transactions.
7. **Staff Management Screen**
   - Entities: Staff
   - Description: For managing hotel staff information, roles, and schedules.
8. **Service Management Screen**
   - Entities: Service, ServiceRequest

   - Description: Manages hotel services and tracks service requests from customers.

9. **Inventory Management Screen**
   - Entities: Inventory
   - Description: Tracks and manages hotel inventory, including adding, updating, and removing items.
10. **Report Generation Screen**
    - Entities: Report
    - Description: Generates and displays various reports (e.g., occupancy rate, revenue) for admin and managers.
11. **Profile Management Screen**
    - Entities: User
    - Description: Allows users to manage their profiles, update contact info, and change passwords.

5. System Workflows

5.1 Booking Workflow:

1. Customer searches for rooms.
2. Selects a room and fills in booking details.
3. System checks availability and confirms booking.
4. Customer makes payment.
5. Booking confirmation is sent to the customer.

5.2 Check-In/Check-Out Workflow:

1. Staff checks customer booking details.
2. Confirms identity and room assignment.
3. Updates room status to 'Occupied.'
4. At check-out, updates room status to 'Available' and generates final bill.

5.3 Payment Workflow:

1. Customer selects a payment method.
2. Payment is processed through an integrated gateway.
3. System updates booking and payment status.
4. Receipt is generated and sent to the customer.

5.4 Service Request Workflow:

1. Customer places a service request through their account.
2. Staff receives the request and assigns it.
3. Service is provided and the request status is updated.
4. Service charge is added to the customer's bill, if applicable.

6. Security Considerations

- **Authentication:** Implement secure login mechanisms (e.g., OAuth, JWT).
- **Authorization:** Role-based access control to ensure that users can only access permitted features.
- **Data Encryption:** All sensitive data should be encrypted, both in transit (SSL/TLS) and at rest.
- **Audit Logging:** Log all user actions for monitoring and troubleshooting.

7. Testing and Deployment

7.1 Testing:

- Unit Testing: Verify individual components.
- Integration Testing: Ensure modules work together seamlessly.
- User Acceptance Testing (UAT): Validate the system against business requirements.

7.2 Deployment:

- Deploy the application to a production environment.
- Set up continuous integration/continuous deployment (CI/CD) pipelines.
- Monitor the system for any issues post-deployment.

8. Maintenance and Support

8.1 Maintenance:

- Regular updates for security patches and feature enhancements.
- Monitor system performance and fix bugs promptly.

8.2 Support:

- Provide user support through a helpdesk or ticketing system.
- Documentation and tutorials for end-users.

9. Future Enhancements

- Integration with third-party booking platforms (e.g., Booking.com, Expedia).
- Mobile app development for both Android and iOS.
- Advanced analytics and AI-driven recommendations.