Hotel Management System

Problem Statement

The problem that the hotel management system aims to solve is the efficient management of hotel operations, including reservation, guest check in/check out, room allocation, billing, inventory management and staff scheduling.

Software Requirement Specification

1. **Introduction:**

1.1 **Purpose of this document:**
To provide a comprehensive understanding of requirements and functionalities of the HMS.

1.2 **Scope of this document:**
Describe the intended users, features and benefits of the HMS along with development tool and this

1.3 **Overview:**
Gives a brief summary of the HMS outlining its primary function such as room booking, check in, check out and billing.

2. **General Description:**
To automate and streamline hotel expectations, improving efficiency and guest experience. Areas guest a book rooms online or through the front desk. Facilitates the check in/out process for guests.

3. Functional Requirements:
 - Room booking:
on criteria Users can search for available rooms based
such as date, room type & occupancy.
users can select rooms and proceed with the
booking process.

Check in/out
 front desk staff can assign rooms, check in
and issue room keys. Guest can check out
settle bills and receive invoices.

Inventory Management
 System automatically update room availability
on reservations and house keeping

4. Interface requirements
 Intuitive interfaces for staff and guests, accessible
via web browsers or mobile apps. Integration with
payment gateways for secure online payment.
Email notification for booking confirmation, reminder

5. Performance Requirements
 System should respond promptly to user request with
minimal latency. System should be available
24/7, with scheduled maintenance windows
communications.

6. Design Constraints

Compatibility with existing for heatwave and software infrastructure Support for multiple platforms.

Compliance with data protection

7. Non functional Attributes
- Encryption of sensitive data
- Role based access control to restrict unauthorized access
- System should be robust and resilient with failover mechanisms to prevent downtime.
- Intuitive user interface with clear navigation and footings.

8. Preliminary Schedule and budget.

Estimated timeline for development, testing and deployment phases.

Budget: cost estimates for development resource and infrastructure

1) **Hotel Management**

UML Class Diagram – Hotel Management System

Hotel Management System
- +hotelName
- +hotelLocation
- +addO
- +removeO
- +searchO
- +editO

Branch
- +branchName
- +branchLocation

Employees
- +name
- +age
- +gender
- +address
- +contact
- +bookRooms()
- +checkRooms()
- +manageDatabase()
- +generateBills()

+consists → Helper
+consists → Chef
+consists → Receptionist

Customer
- +custName
- +custAge
- +custPhoneNo
- +checkIn()
- +checkOut()
- +pay()

Order
- +orderQty
- +orderAmt
- +orderTime
- +calculateTax()
- +calculateTotal()

Payments
- +totalAmt
- +paymentType
- +paymentDate
- +makePayment()
- +printBills()

+CashPayments
+CardPayments

Restaurant
- +name
- +type
- +serve()
- +collectBill()
- +acceptPayment()

Menu
- +itemName
- +itemType
- +itemPrice
- +addItems()
- +removeItems()

+Veg
+Non-veg

«enumeration»
room-type
- singleAc
- singleNonAc
- doubleAc
- doubleNonAc

Rooms
- +roomNo
- +roomType
- +Attribute1
- +search()
- +reserve()

Relationships / multiplicities:
- Hotel Management System 1 — 1 branch_id Branch
- Branch 1 — 1.* +has Employees
- Branch 0.* — 0.* +has Customer
- Customer 1 — 0.* +makes Order
- Order 1 — 1.* +has Payments