Hotel Management System. (HMS)

Problem Statement:

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

Software Requirement Specifications (SRS)

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 cost and time estimates.

1.3 Overview:

Gives a brief summary of the HMS, outlining its primary function such as room booking check-in/out and billing.

2. General Description:

To ~~operate~~ automate and streamline hotel operations, improving efficiency and guest experience. Allows guest to book rooms online or through the front desk. Facilitates the check-in/out process for guests. Tracks room availability, housekeeping status and amenities. Enhance guest satisfaction, increase revenue and optimizes resource utilization.

3. Functional Requirements:

- Room Booking:

Users can search for available rooms based on criteria such as date, room type and occupancy. Users can select rooms and proceed with the booking process.

- Check-in/out:

Front desk staff can assign rooms, check-in guests and issue room keys. Guests can check-out, settle bills and receive invoices.

- Inventory Management:

System automatically updates room availability based on reservations and housekeeping status.

4. Interface Requirements.

Intuitive interfaces for staff and guests, accessible via web browsers or mobile apps.

Integration with payment gateways (eg. Paypal, Gpay) for secure online payments.

Email notifications for booking confirmations, reminder and

feed-back requests.

5. Performance Requirements:
System should respond promptly to user requests, with minimal latency. System should be available 24/7, with scheduled maintenance windows communicated in advance. Ability to handle peak loads during high-demand periods (eg holidays, events).

6. Design Constraints
Compatibility with existing hardware and software infrastructure
Support for multiple platforms (windows, macOS, iOS, etc.)
Compliance with data protection regulations and industry standards.

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 interfaces, with clear navigation and helpful tooltips.
Architecture should support horizontal scaling to accommodate growth in user base and transaction volume.

8. Preliminary Schedule and Budget:
- Schedule:
- Estimated timeline for development, testing and deployment phases.

- Budget: Cost estimates for development, ~~data~~ resources, software licenses and infrastructure.