

19/8/25

Date ___/___/___

Page _____

SRS for Hotel Management System

1. Introduction

1.1 Purpose

This document specifies the requirements for the Hotel Management System (HMS) which automates the booking, check-in, check-out, billing & managing processes of a hotel. It is intended for developers, testers and stakeholders to understand the system functionalities & constraints.

1.2 Scope

HMS will manage reservations, customer profiles, room assignments, billing & generate reports to improve operational efficiency in hotel environment. The system supports both online & on-site booking.

1.3 Overview

The Hotel Management System will cater the needs of hotel staff & management, providing features such as room booking, guest profiles, inventory management and financial reporting. It will be accessible to users with varying levels of technical expertise.

2. General Description:

The Hotel Management System will cater to the needs of hotel staff and management, providing features such as room booking, guest profiles, inventory management and financial reporting. It will be accessible to users with varying levels of technical expertise.

3. Functional Requirements

1.1 Purpose of this Document: The purpose of this document is to outline the requirements and specifications for the development of a Hotel Management System. It will provide a clear understanding of the project objectives, scope, & deliverables.

1.2 Scope of this Document: This document defines the overall working and main objectives.

3. Functional Requirements

3.1 Reservation Management:

- Allow users to make room reservations online or through the front desk.
- Generate reservation confirmations & send notification to guest.

3.2 Room Management

- Assign rooms to guests based on availability & preferences.
- Track room status (clean, occupied, vacant) in real-time.

3.3 Guest Management

- Maintain guest profiles with personal information, preferences and booking history.
- Facilitate guest check-in & check-out processes.

3.4 Billing and Invoicing:

- Generate account bills for room charges, additional services, and taxes.
- Accept various payment methods & generate invoices for corporate clients.

4. Interface Requirements:

4.1 User Interface:

- Intuitive and user-friendly interface for hotel staffs & guests.
- Accessible via web browsers, mobile devices & desktop applications.

4.2 Integration Interface:

- Integration with payment gateways for secure transactions.
- Integration with third-party booking platforms for seamless reservation management.

5. Performance Requirements:

5.1 Response Time: The system should respond to user actions within 2 seconds.

5.2 Scalability: Handle a minimum of 1000 concurrent users during peak hours.

5.3 Data Integrity: Ensure data consistency & accuracy across all modules.

6. Design Constraints:

6.1 Hardware Limitations: The system should be compatible with standard hotel hardware (computer, printer, POS terminals).

6.2 Software Dependencies

- Utilise a relational database management system for data storage.
- Use programming languages & frameworks conducive to modeling (Java, Spring Boot).

7. Non-Functional Attributes:

- 7.1 Security: Implement robust authentication & authorisation mechanisms to protect sensitive data.
- 7.2 Reliability: Ensure high availability & fault tolerance to minimise system downtime.
- 7.3 Scalability: Design the system to accommodate future growth & expansion.
- 7.4 Portability: Support multiple platforms & devices for user accessibility.
- 7.5 Usability: The system shall have a user-friendly interface with clear navigation.
- 7.6 Reusability: The system shall use modular code design to facilitate future enhancements & maintenance.
- 7.7 Compatibility: The system shall be compatible with common web browsers (Chrome, Firefox).
- 7.8 Data Integrity: The system shall ensure accurate

& consistent data storage & retrieval.

8. Preliminary Schedule & Budget: The development of the HMS is estimated to take 6 months with a budget of \$100,000. This includes project planning, development, testing & deployment phases.