

Software Requirement Specifications. [SRS]

19/08/25.

1. HOTEL MANAGEMENT SYSTEM:

1.1. Introduction:

1.1.1. Purpose:

The purpose of this SRS is to define the functional & non-functional requirements of the HIMS. The system aims to automate & streamline hotel operations, including room reservations, check-in/check-out, customer record management, billing & staff administration.

This document serves as a formal agreement between stakeholders, developers and end users. It will guide the development, testing & maintenance of the system. The primary objective of the HIMS is to replace manual hotel operations with a centralized digital solution that improves accuracy, reduces workload, enhances customer satisfaction and ensures secure transactions.

1.2. Document Conventions:

* This document follows the IEEE 830-1998 standard for SRS.

* Section headings represent major components of the SRS. The sections are organized into Introduction, Overall System Feature, External Interface Requirements & other non-functional requirements. Major section titles (like introduction, overall description) are double underlined while subsections are single underlined.

1.3 Scope

The Hotel Management System is a software application that allows hotel staff & customers to manage all hotel related operations efficiently. The system provides functionalities such as online room booking, cancellation, check in, check out, payment, automated billing. The HMS also offers administrative functionalities like staff management, room inventory management & report generation. By eliminating the manual dependency of paper-based systems, the HMS ensures higher accuracy, reduced operational time & improved customer satisfaction. The scope of this project extends to medium & large scale hotels that require computerized assistance in day-to-day operations.

1.4 Intended Audience & Reading suggestions

The intended users of this document include

1. Developers & Designers - who will use this specifications to implement the software.
2. Testers - who will design & execute test cases based on functional & non-functional requirements.
3. End Users (Hotel Staff & customers) - who will refer the user documentation & manuals for system operation.
4. Hotel Management & Administration - who will evaluate how the system aligns with their operational requirements.
5. Project Evaluators / Faculty - who will review this document.

Reading suggestion: Readers with a technical background should review the requirements & design constraints while non technical stakeholders should read the scope & product functions to understand the application.

1.5. References

- * IEEE Std 830-1998: IEEE Recommended Practice for SRS
- * OOMD Syllabus Guidelines
- * Existing hotel reservation software systems as make my trip.

Overall Description:

2.1. Product Perspective

The HMS is designed as a standalone application that operates with a centralized database. It is intended to be deployed on both local hotel servers & cloud based platforms to support remote access. The system interacts with three main entities: the customer, the receptionist, & hotel administrator. Each of these users has a dedicated interface suited to their role. The HMS can be integrated with third party payment gateways to support secure online transactions.

2.2 Product Functions

The HMS supports the following primary functions:

1. Reservation & Booking Management: Customers can search available rooms, make reservations & cancel bookings.
2. Check in & check out management: Receptionists can assign rooms at check in & update room availability & check out.

- ↳ Hotel management system : <http://www.hotelmanagement.com>
- 3. Billing & Payment: Automatic invoice generation & secure payment through integrated gateway.
 - 4. Staff & room inventory management: Administrators can add, remove rooms, update facilities & assign staff duties.
 - 5. Reports & Analytics: Automated daily, weekly & monthly reports for financial tracking & occupancy rates.

3. User Types & Characteristics:

- * Customer: Typically an occasional or frequent hotel guest. Customers need a simple & intuitive interface to search, reserve & pay for rooms.
 - * Receptionist: Hotel staff who manage front desk operations. They require a moderately complex interface for handling check ins & check outs & bookings.
 - * Administrator / Manager: Hotel staff who require advanced functionality like staff allocation, inventory updates & financial reporting.
- ### 4. Operating Environment
- * Front End: Java (JavaFX) or Web (HTML, CSS, JS, React)
 - * Back End: Java / PHP / Node.js with REST APIs
 - * Database: MySQL, Oracle, PostgreSQL, MongoDB, etc.
 - * Operating Systems: Windows 10/11, Linux or cloud servers
 - * Device: Desktops, Laptops, Tablets, mobile with browser

2.5 Clear Documentation

- The HMS will be supported by the following documentation.
1. Customer Clear Manual : Step by step user guide for registration, booking, cancellation, payment.
 2. Receptionist Guide : Instructions for managing rooms, updating staff schedules, handling customer complaints, general tasks.
 3. Administrator Guide : Procedure for managing rooms, staff scheduling, generate reports.

2.6 Design & Implementation Constraints

- * Must use a relational database system (MySQL / Oracle)
- * Payment integration must comply with security standards (PCI DSS encryption)
- * The system must support concurrency for multiple customers booking at the same time.
- * UI design must follow responsive & accessible design guidelines.
- * All passwords must be encrypted before storage.

2.7 Assumptions & Dependencies

- * Customers will provide valid ID & payment information.
- * Internet connectivity will be available for online booking.
- * The payment gateway is assumed to be always functional.
- * Hotel staff will be trained to use the system.

2.8 Appendix :

HMS: Hotel management system

SRS: System requirement software

3. Specific Requirements

3.1 Functional Requirements

- * Customer registration & login.
- * Search available rooms based on criteria (date, price type).
- * Room booking & cancellation.
- * Check-in & out management.
- * Payment processing & invoice generation.
- * Staff management by admin.
- * Report generation.

3.2. Non Functional Requirements

- (should support) major websites to make it user friendly.
- * Performance: Booking confirmation in 2s.
- * Security: End-to-end encryption, hashed passwords.
- * Availability: 99.9% uptime.
- * Usability: User-friendly interface with responsive design.
- * Reliability: Data backup every 24 hrs.
- * Portability: Should work on web, Android & iOS platforms.

3.3 External Interface Requirements

- * Client Interface: Web & mobile dashboards.
- * Database Interface: MySQL with SQL queries for CRUD operations.
- * Payments Gateway: Integration with UPI/Gpay/Credit Card API.

4. System Features (Modules)

- Reservation module : Allows customers to search & book rooms.
- Billing module : Handles, processes & payment confirmation.
- Staff Module : Admin manages employee details.
- Reporting Module : Provides reports on revenue, bookings & staff performance.