19/08/25

1. IEEE SRS Hotel Management System

1. Introduction

1.1 Purpose

This document specifies the requirements for the Hotel Management System (HMS). The system will allow hotel staff and administrators to manage room bookings, customer ~~care~~ check-in/check-out, billing and staff allocation in an efficient and automated manner.

1.2 Document Conventions

- UML diagrams used for ~~modeling~~ modeling
- The term "shall" indicates mandatory requirements
- The term "should" indicates desirable but optional features
- Technical terms are defined in a glossary
- Numbering follows the IEEE recommended practice for SRS

1.3 Intended Audience and Reading Suggestions

- Developers → to design and implement HMS
- Testers → to validate system requirements
- Hotel Staff / Admin → to understand system functionality
- Instructors / Reviewers → to evaluate completeness of the requirements

1.4 Product Scope

The HMS will provide an easy-to-use and efficient solution for hotel operations. It will reduce manual work, prevent booking conflicts, maintain accurate billing records and ensure faster service to customers. By

digitizing core hotel processes, the HMS will improve staff productivity, customer satisfaction, and financial management.

1.5. References

- IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
- College lab guidelines for project submissions
- Hotel industry guidelines or regulatory documents.

2. Overall Description:

2.1 Product Perspective

The HMS is a standalone system that can later be integrated with online platforms. It consists of:

- Customer Module: Booking rooms, checking availability
- Staff Module: Managing guest checkin/check out and billing
- Admin Module: Adding rooms, managing staff, generating reports.

2.2 Product Features

- Room booking and reservation management
- Real-time room availability updates
- Customer check-in and check-out records
- Automated billing and invoice generation
- Staff record management
- Reporting (occupancy, revenue, staff performance)

2.3 User Classes and Characteristics

- Administrator: Full system privileges (rooms, staff, reports).
- Staff: Limited access (guest check-in/out, billing)
- Customer: Limited self service (search rooms, booking)

2.4 Operating Environment:

- Server OS: Windows / Linux
- Database: ~~MySQL~~ MySQL
- Programming Languages: Java / PHP / Python
- User ~~details~~ devices: PC or laptop with at least 4GB RAM and internet browser.

2.5 Design and Implementation Constraints

- Limited lab infrastructure and time constraints
- Database capacity restricted to small-scale demo
- Must follow the technologies mandated by the instructor.

2.6 User Documentation

- User Manual for hotel staff and admin
- Quick Start Guide for customers.

2.7 Assumptions and Dependencies

- Internet access available for customer portal
- Payment gateway API operational
- Hotel staff trained in basic computer usage.

3. System Features

3.1. Room Booking

- The system shall allow customers to search rooms by type, price and date.
- The system shall ensure no two customers can book the same room simultaneously.

3.2. Customer Check-in / Check-out

- Staff shall record check-in details (name, ID, room number, date)
- System shall automatically update room availability after checkout.

3.3. Billing and Payment

- System shall calculate charges based on room rate and duration of stay.
- System shall generate digital/printed invoices.
- Multiple payment methods (cash, card, online) shall be supported.

3.4 Staff Management

- Admin shall add, update, or remove staff accounts.
- System shall assign roles and privileges to staff accounts.

3.5 Report Generation

- System shall generate daily, weekly, and monthly occupancy reports.
- System shall generate revenue and staff activity reports.

4. External Interface Requirements

4.1 User Interfaces
- Login screen (username, password, role selection)
- Admin dashboard (rooms, staff, reports)
- Staff dashboard (bookings, check-ins, check-outs, billing)
- Customer booking page.

4.2 Hardware Interfaces
- Standard PC/Laptop with minimum configuration: Intel i3 or equivalent, 4GB RAM, 250GB HDD.

4.3 Software Interfaces
- Database: MySQL
- Backend: Java/PHP/Python
- Frontend: HTML, CSS, Javascript.

4.4 Communication Interfaces
- Secure HTTPS protocol for web interactions.
- Local network access in lab environment.

5. Nonfunctional Requirements

5.1 Performance Requirements
- The system shall handle up to 50 simultaneous booking requests.
- Average response time shall be less than 2 seconds

5.2 Safety Requirements
- System shall regularly back up booking and billing records.
- Data loss prevention measures shall be implemented.

5.3 Security Requirements

- System shall regularly back up booking and billing records.
- Data loss prevention measures shall be implemented
- All passwords shall be stored in encrypted form
- Role-based access control shall prevent unauthorized access
- Customer data shall not be accessible by other customers.

5.4. Software Quality Attributes

- Usability: Simple, intuitive interface.
- Reliability: System uptime of 99% in lab tests.
- Portability: Compatible with Windows and Linux OS

6. Other Requirements.

- System must comply with the academic project evaluation criteria
- Source code must be documented for maintainability.