1. Introduction

- Purpose
- Scope
- Definitions, Acronyms, and Abbreviations
- References

2. Overall Description

- 2.1 Product Perspective
- 2.2 Product Functions
  - Room Availability Management
  - Booking Management
  - Guest Management
  - Billing
  - User Authentication and Authorization
- 2.3 Assumptions and Dependencies

3. Specific Requirements

- **3.1 Functional Requirements**

- Room Availability Management
  - Add Room/Room Type
  - Update Room Details
  - Search Room Availability
  - Manage Room Blocking
  - Generate Room Occupancy Reports
- Booking Management
  - Create Booking
  - View Booking Details
  - Modify Booking
  - Cancel Booking
  - Manage Booking Confirmations
  - Handle Group Bookings
- Guest Management
  - Guest Registration
  - Manage Guest Profile
  - Handle Guest Check-in/Check-out
  - Track Guest History
  - Manage Guest Requests
- Billing
  - Generate Guest Folio
  - Record Payment

        - Calculate Room Charges
        - Manage Taxes and Fees
        - Generate Billing Reports
    - User Authentication and Authorization
        - User Login
        - User Roles
        - Password Management
- **3.2 Non-Functional Requirements**
- Usability
- Reliability
- Performance
- Security
- Maintainability
- **3.3 External Interface Requirements**
- User Interface (UI)
- Software Interfaces
- Hardware Interfaces
- **3.4 Constraints**

4. Appendices

- 4.1 Glossary
- 4.2 Supporting Documents
    - Use Case Diagram
    - Entity-Relationship Diagram
    - UI Mockups

“Software Requirements Specification (SRS) for Hotel Reservation and Management System”

1. Introduction

- **Purpose**

The purpose of this document is to specify the requirements for a web-based Hotel Reservation and Management System. This system aims to streamline and automate hotel operations, including room availability management, booking management, guest management, and billing processes. It will provide a user-friendly platform for both hotel staff and customers to efficiently manage hotel reservations and related activities.

- **Scope**

This SRS document covers the requirements for the following aspects of the Hotel Reservation and Management System:

  - **Room Availability Management:** Managing details of available rooms, including room types, capacity, amenities, and availability status.
  - **Booking Management:** Handling room reservations, including booking creation, modification, cancellation, and tracking.
  - **Guest Management:** Managing guest information, including registration, check-in, check-out, and guest history.
  - **Billing:** Generating guest folios, processing payments, and managing payment records.

This document does not cover:

  - Integration with external point-of-sale (POS) systems for restaurant or other hotel services.
  - Advanced event management or conference booking features.
  - Detailed employee scheduling and payroll.

- **Definitions, Acronyms, Abbreviations**
  - SRS: Software Requirements Specification
  - UI: User Interface
  - DBMS: Database Management System
  - API: Application Programming Interface
  - OTA: Online Travel Agency
  - PMS: Property Management System

- **References**
  - IEEE/ANSI 830-1998: Recommended Practice for Software Requirements Specifications (General SRS standard)
  - Project proposal document (assumed document that initiated this SRS)

2. Overall Description:

- ### 2.1 Product Perspective:

The Hotel Reservation and Management System will be a web-based application. It will be a standalone system, but it will interact with a database system to store and retrieve data. The system will provide different user interfaces for hotel staff (administrators, receptionists) and customers. It may also have the potential to integrate with external systems via APIs.

- ### 2.2 Product Functions:

The system will provide the following functions:

1. **Room Availability Management:**
   - Add new rooms and room types to the system.
   - Update room details (e.g., amenities, status).
   - Search for available rooms based on various criteria (e.g., room type, dates).
   - Manage room blocking (e.g., for maintenance).
   - Generate room occupancy reports.
2. **Booking Management:**
   - Create new room reservations.
   - View booking details.
   - Modify existing bookings.
   - Cancel bookings.
   - Manage booking confirmations.
   - Handle group bookings.
3. **Guest Management:**
   - Register new guests.
   - Manage guest profiles (e.g., contact information, preferences).
   - Handle guest check-in and check-out.
   - Track guest history.
   - Manage guest requests (e.g., extra towels).
4. **Billing:**
   - Generate guest folios.
   - Record payments.
   - Calculate room charges.
   - Manage taxes and fees.
   - Generate billing reports.
5. **User Authentication and Authorization:**
   - Secure login for staff and customers.
   - Different access levels for different user roles.

- **2.3 Assumptions and Dependencies:**
    1. It is assumed that the system will be deployed on a reliable web server.
    2. The system's functionality depends on the availability of a functional database management system (DBMS).
    3. It is assumed that the network connection is stable for uninterrupted access.
    4. Integration with any third-party services (e.g., payment gateways) will depend on the availability and compatibility of their APIs.

3. Specific Requirements:

- **3.1 Functional Requirements:**

1. **Room Availability Management:**
    - **Add Room/Room Type:** The system shall allow administrators to add new rooms and room types to the system. Room details shall include room number, room type (e.g., single, double, suite), capacity, amenities (e.g., Wi-Fi, TV, balcony), and standard rate.
    - **Update Room Details:** The system shall allow administrators to update room details, such as amenities, status (e.g., available, occupied, out of service), and rates.
    - **Search Room Availability:** The system shall allow staff and customers to search for available rooms based on criteria such as:
        i. Room type
        ii. Check-in date
        iii. Check-out date
        iv. Number of guests
        v. Price range
    - **Manage Room Blocking:** The system shall allow administrators to block rooms from being booked for maintenance, renovations, or other reasons.
    - **Generate Room Occupancy Reports:** The system shall generate reports on room occupancy, including:
        i. Occupancy rate for a given period
        ii. Number of rooms occupied/available
        iii. Average length of stay
2. **Booking Management:**
    - **Create Booking:** The system shall allow staff and customers to create new room reservations. Booking details shall include:
        i. Guest information (name, contact details)
        ii. Room type
        iii. Check-in date and time
        iv. Check-out date and time
        v. Number of guests

vi. Special requests (if any)

- **View Booking Details:** The system shall allow staff and customers to view booking details, including all information entered during booking creation.
- **Modify Booking:** The system shall allow staff and customers to modify bookings, such as:
  i. Changing check-in/check-out dates
  ii. Changing room type (subject to availability)
  iii. Adding or removing guests
  iv. Updating contact information
- **Cancel Booking:** The system shall allow staff and customers to cancel bookings. The system shall enforce cancellation policies and calculate any applicable cancellation fees.
- **Manage Booking Confirmations:** The system shall automatically generate and send booking confirmations to customers via email.
- **Handle Group Bookings:** The system shall facilitate the management of bookings for multiple rooms under a single reservation.

3. **Guest Management**

i. **Guest Registration:** The system shall allow staff to register new guests during check-in, collecting information such as:
   1. Name
   2. Contact details (phone, email)
   3. Address
   4. Identification details (e.g., ID card number, passport number)

ii. **Manage Guest Profile:** The system shall allow staff to update guest information and record guest preferences (e.g., room preferences, allergies).

iii. **Handle Guest Check-in/Check-out:** The system shall manage the check-in and check-out process, including:
   1. Assigning rooms to arriving guests
   2. Recording check-in and check-out times
   3. Generating welcome messages
   4. Facilitating late check-out requests

iv. **Track Guest History:** The system shall maintain a history of guest stays, including:
   1. Dates of stay
   2. Room numbers
   3. Charges incurred
   4. Feedback or reviews

v. **Manage Guest Requests:** The system shall allow staff to record and track guest requests (e.g., extra towels, room service).

4. **Billing:**

i. **Generate Guest Folio:** The system shall generate a guest folio, which is an itemized bill of charges incurred by the guest during their stay. Charges may include:
   - Room charges
   - Taxes and fees
   - Additional services (e.g., room service)

ii. **Record Payment:** The system shall allow staff to record payments made by guests, including:
   - Payment method (e.g., cash, credit card)
   - Payment amount
   - Date and time of payment

iii. **Calculate Room Charges:** The system shall calculate room charges based on:
   - Room type
   - Duration of stay
   - Seasonal rates or discounts

iv. **Manage Taxes and Fees:** The system shall automatically calculate and apply applicable taxes and fees to guest bills.

v. **Generate Billing Reports:** The system shall generate various billing reports, such as:
   - Daily revenue reports
   - Occupancy revenue reports
   - Outstanding payments reports

- **3.2 Non-Functional Requirements:**

1. **Usability:**
   - The system shall have a user-friendly and intuitive interface.
   - The system shall provide clear and concise error messages.
   - The system shall be easy to navigate.
2. **Reliability:**
   - The system shall be available 99% of the time.
   - The system shall handle errors gracefully and prevent data loss.
   - The system shall have a backup and recovery mechanism.
3. **Performance:**
   - The system shall respond to user requests within 3 seconds.
   - The system shall be able to handle a large number of concurrent users.
   - The system shall be optimized for fast data retrieval and storage.
4. **Security:**
   - The system shall protect user data from unauthorized access.

- The system shall use secure authentication and authorization mechanisms.
- The system shall validate user input to prevent security vulnerabilities (e.g., SQL injection).

5. **Maintainability:**
   - The system shall be designed in a modular way to facilitate future modifications and enhancements.
   - The system shall be well-documented.

- **3.3 External Interface Requirements**

1) **User Interface (UI):**
   i. The system shall have a web-based UI accessible through standard web browsers.
   ii. The UI shall be responsive and adaptable to different screen sizes.
   iii. The UI shall be designed according to modern web design principles.
2) **Software Interfaces:**
   i. The system shall interact with a database management system (DBMS) (e.g., MySQL, PostgreSQL).
   ii. The system may optionally integrate with a payment gateway for online payment processing.
3) **Hardware Interfaces:**
   i. The system shall be compatible with standard web server hardware.
   ii. The system shall be compatible with standard client hardware (e.g., desktop computers, laptops, mobile devices).

- **3.4 Constraints:**
  1) The system must be developed using specific technologies (e.g., programming language, framework, database). (Specify the technologies if required)
  2) The development must be completed within a specific timeframe.
  3) The development cost must be within a specific budget.

4. Appendices

- **4.1 Glossary:**
  - **Guest Folio:** An itemized bill of charges incurred by a guest during their stay.
  - **PMS (Property Management System):** A software system used by hotels to manage various operations.
  - **OTA (Online Travel Agency):** A website that allows users to book hotels online.

- **4.2 Supporting Documents:**
  - Use Case Diagram (UML)
  - Entity-Relationship Diagram (ERD)
  - UI Mockups