Date __/__/__
Page _____

SRS for a Hotel Management System

1 Introduction

1.1 Purpose

The purpose of this document is to specify the software requirements for the Hotel Management System (HMS), a web-based application designed to streamline hotel operations. It serves as a foundational blueprint for developers, project managers, and stakeholders, ensuring a shared understanding of the system's intended functions and features.

1.2 Scope

This SRS covers the core functionality required for managing hotel operations, including guest check-in / check-out, reservations, room inventory, and basic reporting. It does not cover advanced features like point-of-sale integrations for restaurants or advanced marketing and loyal programs.

1.3 Overview

The HMS will be a secure, centralized system accessible via a browser. It will be designed to simplify daily tasks for front-desk staff and provide management with real-time insights into hotel performance.

2. General Description

The HMS is intended for use by hotel staff, including receptionists, housekeeping and management. It will be a cloud-based solution, allowing for access from any location with an internet connection. The system's primary goal is to replace manual or disparate processes with a single, integrated platform. The user interface will be sim

simple and intuitive, minimizing the need for extensive training. The system will handle multiple user roles with different access permissions.

3. FUNCTIONAL REQUIREMENTS

The system shall perform the following core functions:

- Reservations Management: Allow staff to create, modify, and cancel reservations, including group bookings and walk-ins.
- Guest Management: Store and retrieve guest profiles, including contact information, stay history, and special requests.
- Room Inventory: Display real-time room status (eg. occupied, vacant, dirty, out of order) and enable staff to update it.
- Check-in / check-out: Facilitate quick and easy guest check-in and check-out processes, including room assignment and key card generation.
- Billing and payments: Automatically generate invoices, apply discounts, and process payments via credit card, cash, or other supported methods.
- Reporting: Generate daily, weekly, and monthly reports on key metrics such as occupancy rates, revenue per available room (RevPAR), and payment summaries.

4. INTERFACE REQUIREMENTS

- User Interface: The system shall have a responsive, web-based UI accessible on desktop, tablet and mobile devices

Date __/__/__
Page _____

- Hardware Interface: The system shall be able to connect with and print receipts and invoices from standard thermal printers at the front desk.
- Software Interface: The system shall integrate with a third-party payment gateway to process secure online transactions.

5. PERFORMANCE REQUIREMENTS

- Response Time: All search queries and data retrieval operations shall complete within 2 seconds.
- Concurrency: The system shall support up to 25 concurrent users without significant performance degradation.
- Availability: The system shall maintain a minimum uptime of 99.9% per month.

6. DESIGN CONSTRAINTS

- Technology Stack: The system must be developed using a modern web framework like React and a scalable backend such as Node.js or Python. The database must be PostgreSQL for data integrity and performance.
- Compilance: The system must comply with all relevant data privacy laws, including the secure handling of guest personal and payment information.

- Scalability: The architecture must be designed to support the addition of new features and accomodate an increase in the number of concurrent users and properties.

7. NON-FUNCTIONAL REQUIREMENTS

- Security: The system shall implement role-based access control. All data, especially guest information and payment details, shall be encrypted both in transit and at rest.

- Usability: The UI shall be intuitive, requiring minimal training for new staff members. Key workflows, such as check-in shall be simple and efficient.

- Reliability: The system shall be robust and handle unexpected errors gracefully, with an automatic backup and recovery plan in place.

- Maintainability: The codebase shall be well-documented and modular, making it easy for future developers to maintain and update.

8. PRELIMINARY SCHEDULE AND BUDGET

- Sche

- Schedule: The estimated development timeline for the core features is 6-8 months, followed by a one-month testing and deployment phase.

Date __/__/__
Page _____

- Budget: The preliminary budget for development, based on a team of 3-4 developers, is estimated to be in the range of $50,000 to $80,000. This includes ongoing maintainance and hosting costs.