

# SRS Document for

## 1) Hotel Management system

### 1. Introduction

1.1 Purpose: The purpose of this document is to provide a detailed description of the hotel management system, which will handle the operation of the hotel, including room booking, guest management, billing and reporting.

1.2 Scope: The system will automate the daily tasks of the hotel management, such as guest registration, room booking, check-in/check-out, billing and reporting. It will be used by hotel staff, including receptionists, managers and housekeeping.

### 1.3. Definitions, Overview:

This document The Hotel Management system is a comprehensive software application designed for small to large hotels. It automates hotel operations and provides integrated solution for room reservation.

## 2. General Description.

### 2.1. Objectives of user:

- Hotel managers: ability to view real-time reports, manage room availability, oversee billing and ensure the overall smooth operation of a hotel
- Receptionists: facilitate guest check-ins & check-outs, manage room bookings, handle billing,
- Housekeeping staff: manage room cleaning and coordinate with staff.

### 2.2. User characteristics:

Manager: Experience with hotel operations

Receptionists: customer-service oriented, with basic knowledge of hotel systems.

Housekeeping staff: familiar with hotel operation and room management.

### Functional Requirements:

- Guest Registration & Authentication: Allow guest to register with their details.
- User accounts will be created with authentication for secure access.
- Room Reservation.
- Check-in and Check-out.
- Billing and payments.
- Generate reports.

## 4) Interface Requirements

Software interface: system uses RESTful APIs for communication with external systems such as payment gateways and email services.

### User Interface:

A web interface for receptionists & hotel managers to monitor bookings, payments and reports.

### Communication interfaces:

SMS/Email Integration for booking confirmation and reminders to guests.

## 5) Performance Requirements

- S.1 Response Time: The system should provide real-time updates for room availability.
- Response time for booking confirmation & check-in & check-out should not exceed 2 seconds.

## 6. Design constraints

- Software constraint: system use MySQL for databases.
- The Frontend will be developed using React.

Hardware constraints: The system will be deployed on a cloud based server to ensure scalability & reliability.

## NON-FUNCTIONAL ATTRIBUTES

- security
- portability
- Reliability
- scalability

## 7. Preliminary schedule & budget

~~Phase 1: Requirement gathering (2 weeks)~~

~~Phase 2: System design & architecture (4 weeks)~~

~~Phase 3: Development & implementation (8 weeks)~~

~~Phase 4: Testing & quality assurance (4 weeks)~~

~~Phase 5: Deployment & training (2 weeks)~~

## Budget:

~~Development: 50,000~~

~~Hardware/software: 10,000~~

~~Testing & deployment: 5,000~~

## 1) Library Management System.

1. Introduction: The purpose of this document is to give detailed overview of the library management system. To be different functional of the system.

Scope: The scope of this project is to implement different functionalities like maintaining catalog, keeping track of the books that are issued and non-membership plans to register members, the payment system, analysing state of books before and after issuing, keep record of users and their history of issuing.

## 2) General description:

### 2.1 Objectives of users

Manager: Manager should get the monthly report of library regarding stock and demand.

Staff: Staff should be able to manage members detail and their history of books issued and return. he should be able to route the members in direction of target books.

members: they should be able to track the record of the books they have borrowed and due date to return the book.

### 2.2 User characteristics → Manager

→ receptionist staff.

→ members.

→ security.

→ Management staff.

### 3) functional requirements

members registration & authentication: members should be able to register themselves to the application ~~and~~ & using their username & password, ~~membership plans~~: user must be made available with different plans with different number of access time and resource.

payment: users ~~can~~ can perform the payment through online with credit card, or other payment gateways.

catalog: users should be able to scroll through the different books available which are of different niche or genre and should be categorized based on age, like children, adult and elders.

### 4) interface Requirements

software interface: system uses restful API to communicate with external systems such as payment gateways and email address.

user interface: a web interface for library management system manages members data, history and section to find books.

communication interface:

sms / email integration for registration and payment confirmation.

### 5) Performance Requirements

→ Response times should provide real time update for available books

• response time for confirmation should be less,

6. Design constraints:
- front end uses reactjs
  - backend implemented using express, mongoDB for database

7) non-functional attributes.

- Reliability
- scalability
- availability

8) preliminary schedule & budget.

schedule:

phase 1: requirement gathering (3 weeks)

phase 2: system design & architecture (4 weeks)

phase 3: Development implementation (8 weeks)

phase 4: Testing (? weeks)

phase 5: Deployment (2 weeks)

## Credit Card System

1) Introduction: This document is intended to give an overview of different functionality of system. It is aimed at implementing functionalities like applying for credit card, payment tracking, scanning QR code, managing credit score.

### 2) General Description:

Manager: He should get control to handle all the communication to bank staff, and height management.

Staff: Staffs should able to get the <sup>customer</sup> detail based on user specification of detail and keep track of user credit scores.

User: He should be able scan credit card and get all details, monitor spending, make payment.

### 3) Functionality requirement:

Application: User will be able to apply for a credit card by providing necessary personal & details.

Validation: Validate user based on predefined criteria.

Account Management: Card holder can view their balance, credit limit & availability, user can update their information.

Transaction status: Cardholders can view status of transaction.

### 4) Interface Requirements:

Software interfaces: System uses restful API's to communicate with external system.

A web interface for credit card management system which keeps track of user details & payment status.

Performance requirements: Response time for payment confirmation and user registration. It should provide real-time transaction updates of user.

#### 6) Design constraints:

- front end using React JS
- backend implemented using express, and MongoDB for database.

#### 7) Non-functional attributes:

- Reliability
- scalability
- availability.

#### 8) Preliminary schedule & budget

Schedule:

Phase 1: requirement gathering (2 weeks)

Phase 2: system design architecture (4 weeks)

Phase 3: Development (8 weeks)

Phase 4: testing (2 weeks)

Phase 5: Deployment (1 week)

✓  
Sep  
Oct  
Nov  
30/10/20

# Stock Maintenance System

## 1. Introduction

1.1 Purpose of this document: The purpose of this SRS document is to outline the requirements for Stock Maintenance System. This document serves as a communication tool between stakeholders, including customers, project managers, and development teams.

1.2: The Stock Maintenance System is designed to streamline inventory management for business, enhancing efficiency and accuracy. This document describes the overall working of SMS, highlighting its objectives to minimize stock discrepancies and optimize stock levels. The anticipated development time is approx six months, with an estimated budget of 50,000. This system will provide value to customers.

## 1.3 Overview

The SMS is a web-based application that allows the businesses to manage their inventory seamlessly. It includes features such as stock tracking, automated alerts for low inventory, and comprehensive reporting capability.

## 2) General description:

The SMS is designed for inventory management & warehouse staff who require efficient tools for managing stock levels. Key features include:

Stock tracking: Realtime visibility into stock levels

Automated alerts: Notifications for low stock reordering.

Reporting tools: detailed analysis for informed decision.

## 3) Functional Requirements

1. User authentication.

2. Inventory Management.

3. Low stock alerts.

4) Reporting:

5) Search functionality

9) interface Requirements  
User interface: web based dashboard to interact  
Database: a relational database for storing stock info  
APIs: Integration with supplier for order management

### 3) performance requirements:

- Response time.
- concurrency.
- data integrity

### 4) Design constraints:

Technology stack: use TS for frontend & node.js for backend

Database: MySQL

Compatibility: compatible with major web browsers

### 7) Non-functional:

• Security.

• reliability.

• scalability.

• portability.

### 8) preliminary schedule & budget

phase 1: req gathering (2 weeks)

phase 2: system design architecture (3 weeks)

phase 3: development

(4 weeks)

phase 4: testing

(2 weeks)

phase 5: deployment

(1 week)

# Passport Management system

## 1. Introduction

### 1.1 purpose :

The purpose of this SRS document is to outline the requirement for passport management system (PMS). This document is intended for stakeholders, including government officials, project managers & developers, to provide a comprehensive understanding of system's functionalities, goals & implemented strategy.

### 1.2 scope

The PMS aims to streamline the application, issuance & renewal of passport for citizens. It includes functionalities such as online submissions, status tracking, & appointment scheduling. The anticipated development time is approx month.

### 1.3 Overview:

The PMS is a web-based application that enables users to apply for, renew, & track their passport status online. The system provides a user-friendly interface, auto mailed notifications for application status update, & secure data handling to ensure the confidentiality of user information.

## 2. General Description:

The passport management system is designed for citizens applying for passports, govt officials processing applications & administrative staff managing records. Key features include:

- Online application submission
- Status tracking
- Appointment scheduling

### 3. Functional Req:

- user Registration & Authentication
- application form.
- Document upload.
- status notification.
- appointment Management.

### 4. Interface Requirements

- user interface: website.
- Database: a secure database to store user data & application records, & document upload.
- govt system: Integration with existing govt database.

### 5. Performance Req:

- Response time
- Concurrency
- Data integrity

### 6. Design constraints.

Technology stack: HTML, React, CSS, Node.js.

Database limitation: use MySQL to storage.

Compliance: Must comply with govt regulations regarding data security & privacy

### 7. Non-functional

- Security
- Reliability
- Scalability
- Usability

### 8. Preliminary schedule

Phase 1: req gathering (2 weeks)

Phase 2: System design Architecture (2 weeks)

Phase 3: development (3 weeks)

Phase 4: testing (2 weeks)

Phase 5: deployment (2 weeks)