Hotel Management System

Problem Statement:
Design and implement a Hotel Management System (HMS) that is required to automate and streamline the operations like delayed check-ins/check-outs, billing errors, tracking room availability. ~~billin~~ The System should manage customer details, reservations, room availability, staff & billing efficiently while reducing manual work & ensuring accuracy.

1. Introduction:

1.1 Purpose of the Document:

The Purpose of this SRS Document is to define the requirements for the (HMS) Hotel management System. It will minimize the manual errors improving staff productivity, and allows for better resource allocation.

1.2 Scope of the Document:

The HMS will automate tasks such as room booking, check-in/out, billing, staff managment financial management, and generating reports. It will improve efficiency, minimize errors, and enhance the customer experience. The System is accessible via both web & dustop.

1.3 Overview:

The HMS provides an all-in-solution to manage hotel operations. It tracks room availability, automates reservations, handles payment, guest data billing, housekeeping, managing staff records, and generate reports for analysis.

2. General Description:

- Users: Receptionists, Managers, Customers, Staff.

- Features: Room booking, availability check, check-in / check-out, billing & invoices, staff management, report generation.
- Benefits: Reduces human errors, saves time, ensures data security, improves customer satisfaction
- User community: Customers (for booking), hotel staffs (for managing rooms & billing), and managers (for administration and reporting).

3. Functional Requirements :-

FR1: The System shall allow customer registration and login.

FR2: The System shall provide room availability checks.

FR3: The System shall support both online & offline booking of rooms.

FR4: The System shall manage check-in and check-out process

FR5: The System shall generate bills & handle payment processing.

4. Interface Requirements :

- User Interface: Web-based dashboard with easy navigation, mobile app for customers.
- Software Interfaces: Integration with payment gateways, email & / sms notification APIs
- Hardware Interface: Barcode / QR scanner for room keys, printers for bills.

5. Performance Requirements :

- The system should handle at least 1000 concurrent users.
- Response time should be < 3 seconds for booking and billing operations.
- Database should support quick queries for room availability.

6. Design Constraints:

- Must support windows, Linux, & web environments.
- Database: MySQL / PostgreSQL
- Frontend: HTML, CSS, Javascript / React.
- Backend: Node.js / Java / Python.
- Mobile app: Flutter

7. Non-Functional Attributes:

- Security: Data encryption for payments & user details.
- Reliability: 24/7 uptime with backup.
- Scalability: Support for multiple hotels/ branches.
- Portability: Compatible across devices
- Maintainability: Easy updates & bug fixes.

8. Preliminary Schedule & Budget:-

- Time Estimate: 3-4 months for development.
- Cost estimate: Depends on our team size, approx: $15000 - $25,000.
- Phases: Requirement Analysis → Design → Development → Testing → Deployment → maintenance.

Credit Card Processing System

Problem Statement:

Design and implement a Credit Card Processing System (CCPS) that enables secure, fast, and reliable handling of credit card transactions. The system should address issues of delayed payments, fraud risks and lack of transparency by providing real-time authorization, settlement, and reporting while ensuring compliance with PCI DSS standards.

1. Introduction:

1.1 Purpose of this Document

The purpose of this document is to describe the requirements for developing a Credit Card Processing System (CCPS). The system will handle secure and fast processing of credit card transactions including authorization, settlement, and fraud detections. This document is mainly for developers, testers, and stakeholders to understand the system's objectives & functions clearly.

1.2 Scope of this Document

The system will allow businesses (merchants) to accept payments from customers through credit cards. It will securely capture card details, verify the transaction with the issuing bank, and maintain compliance with financial security standards such as PCI DSS (Payment Card Industry Data Security Standard).

1.3 Overview

The System works as a bridge blw the customer's bank (issuer) and the merchant's bank (acquirer). Once a customer swipes or entries their credit card details, the transaction will be authorized within 2-5 seconds. The system also helps in detecting fraudulent transactions by using algorithms and real-time monitoring.

2. General Description:

- Users: Merchants, customers, Bank Administrators
- Main functions: Card authorization, transaction processing, fraud detection, settlement, and report generation.
- Benefits: Secure and fast payments, reduced fraud risks, 24/7 availability, improved customer satisfaction.
- Community: customers using credit cards, merchants accepting payments, and banks managing accounts.

3. Functional Requirements:

FR1: The System shall allow merchants to register & log in securely.
FR2: The System shall accept credit card details from customer in a secure way.
FR3: The System shall validate credit card details before processing.
FR4: The System shall communicate with the issuing bank for transaction authorization.
FR5: The System shall complete the transaction within 2-5 secords under normal conditions.
FR6: The System shall provide fraud detection and alert mechanisms.

FR7: The system shall support settlement and transfer of funds to the merchant's account.

FR8: The system should provide multi-currency support for international transactions.

FR9: The system should allow integration with e-commerce websites & mobile apps.

4. Interface Requirements:

- User Interface: A web-based dashboard for merchants and admins, mobile app support for customers.
- Software Interfaces: Integration with payment gateways, banks' APIs, & fraud detection services.
- Hardware Interfaces: POS terminals, card readers, NFC-enabled devices.

5. Performance Requirements:

- The System shall process each transaction within a maximum of 5 seconds.
- The system shall handle at least 500 transactions per second (TPS) during peak times.
- The system shall have an uptime of 99.9%, to ensure continuous availability.
- The System shall support high-volume transaction for international merchants.

6. Design Constraints:

- The system shall comply with PCI DSS security standards.
- All sensitive data shall be encrypted before storage or transmission.
- The System shall use SSL/TLS for secure communication.

- The system shall support both cloud-based and on-premises deployment.

7. Non-Functional Attributes :

- Security : End-to-End encryption, tokenization, and fraud monitoring.
- Reliability : 24/7 service availability with backup servers.
- Scalability : should handle growing no. of merchants and users.
- Portability : Should work across platforms
- Maintainability : Easy to update compliance features as per banking regulations.

8. Preliminary Schedule and Budget :

- Timeline :
  - Requirement Analysis - 2 weeks
  - System Design - 3 weeks.
  - Development - 8 weeks
  - Testing - 4 weeks.
  - Deployment - 2 weeks.
- Estimated Duration : Around 4-5 months
- Estimated Budget : $50,000 - $80,000 depending on integration with multiple banks & fraud detection modules.

Library management System

Problem Statement:

Design & implement a Library management System (LMS) that automates the process of managing books, members, and transactions in a library. Manual systems often face problems like misplaced records, difficulty in tracking issued/returned books, and time-consuming searches. The proposed system should provide efficient book cataloging, member management, issue/return tracking, & fine calculation. It should also generate reports & ensure quick access to information for both librarians & students.

1. Introduction

1.1 Purpose of this Document

The purpose of this SRS is to define the requirements for the Library management System. It will help developers, librarians, & administrators to understand the system's features, constraints, and objectives clearly.

1.2 Scope of this Document

The LMS will handle book records, member registration, book issue/return, and fine calculation. It will reduce manual work, improve efficiency, and make searching and managing library resources much faster.

1.3 Overview

The System will maintain a database of books, tracks issued and available copies, and allow librarians to generate reports. Members will be

able to search for books, view availability, and borrow or return books through the system.

2. General Description:

- Users: Librarians, Students/Members, Administrators.
- Main Functions: Book catalog management, member registration, issue/return tracking, fine calculation & report generation.
- Benefits: Saves time, reduces errors, improves accessibility, and enhances resource utilization.

3. Functional Requirements:

FR1: The System shall allow member registration and login.
FR2: The System shall allow librarians to add, update & delete book records.
FR3: The System shall allow searching of books by title, author or category.
FR4: The System shall manage issue & return of books with proper tracking.
FR5: The System shall calculate fines for late returns automatically.
FR6: The System shall maintain records of all transactions.
FR7: The System shall generate reports.
FR8: The System should provide notifications/reminders for due dates.
FR9: The System should support barcode scanning for faster transactions.

4. Interface Requirements:
- User Interface: A simple & user-friendly web/mobile interfaces.
- Software Interfaces: Database system for storing records, notification APIs (SMS/email).
- Hardware Interface: Barcode scanners, printers for receipts/reports.

5. Performance Requirements:
1. The System shall process book searches in less than 2 seconds.
2. The System shall support at least 5000 books and 1000 active members.
3. The System shall be available 99% of the time during library hours.

6. Design Constraints:
- The System shall use a relational database (MySQL / PostgreSQL).
- The System shall support both desktop & web access.
- The System shall ensure data security & regular backups.

7. Non-Functional Attributes:
- Security: Authentication for users & librarians, data backup
- Reliability: Accurate tracking of issued and returned books.
- Scalability: Ability to handle large numbers of books and users.

- Maintainability: Easy to update & upgrade

8. Preliminary Schedule & Budget:-
   - Timeline: 3 months (Requirement analysis - 2 weeks, Design - 3 weeks, Development - 6 weeks, Testing - 3 weeks, Deployment - 2 weeks).
   - Budget: Estimated $8,000 - $15,000

Stock Maintenance System

Problem Statement:

Design and implement a stock maintenance system (SMS) to automate the management of product's, stock levels, and transactions in an organization. Manual stock handling often causes issues like inaccurate inventory counts, delays in updating stock records, and difficulties in tracking sales or purchases. The proposed system will provide real-time stock updates, low-stock alerts, transaction history, and reporting features, ensuring efficiency, accuracy and better decision-making.

1. Introduction:

1.1 Purpose of this Document:

This document defines the requirements for the stock maintenance system. The system will help organizations track inventory levels, update stock records, manage suppliers/customers, and generate useful reports for planning and analysis.

1.2 Scope of this Document:

The SMS will allow users to add, update & remove stock items, track sales and purchases, monitor low-stock items, and generate periodic reports. It will reduce manual errors & provide a clear view of stock availability at any time.

1.3 Overview:

The System will act as a central database for products and inventory. It will maintain details such as item name, category, quantity, price, and supplier information. Automated alerts will notify users when items fall below minimum stock levels.

2. General Description:

- Users: Store Managers, Staff, Administrators
- Main Functions: Stock entry and update, sales and purchase tracking, low-stock alerts, and report generation.
- Benefits: Real-time stock visibility, reduced human error, improved planning and timely restocking

3. Functional Requirements:

FR1: The system shall allow users to add, update, and delete stock items.

FR2: The System shall record stock transactions (sales, purchases, returns)

FR3: The system shall update stock levels automatically after each transaction.

FR4: The system shall generate alerts for items running below minimum stock levels.
FR5: The system shall generate reports.
FR6: The system shall allow search & filter of products by category, supplier or name.
FR7: The system should support barcode / QR scanning for faster data entry.
FR8: The system should allow exporting reports to excel / PDF.

4. Interface Requirements:

- User Interface: Web-based dashboard with search & filter options, simple forms for stock entry.
- Software Interfaces: Database system, optional integration with POS systems.
- Hardware Interfaces: Barcode scanners, printers for invoices & reports.

5. Performance Requirements:

- The system shall process stock updates within 2 seconds.
- The system shall handle at least 50,000 stock items efficiently.
- The system shall support at least 200 concurrent users in large organizations.

6. Design Constraints:

- The system shall use a relational database for stock data.
- The system shall ensure login and role-based access control.

- The system shall provide backup and recovery options for stock data.

7. Non-Functional Attributes:

- Security: Authentication, authorization and data encryption.
- Reliability: consistent stock updates without data loss.
- Scalability: Should handle increased stock and users as business grows.
- Maintainability: Easy to modify stock categories or add new modules.

8. Preliminary Schedule and Budget:

- Timeline: Around 3-4 months (Requirement analysis - 2 weeks, Design - 3 weeks, Dwelopment - 6 weeks, Testing - 3 weeks, Deployment - 2 weeks)
- Budget: Estimated $10,000 - $20,000 depending on features and integrations.

Passport Automation System

Problem Statement:

Design and implement a Passport Automation System (PAS) that simplifies and digitizes the process of applying for, verifying and issuing passports. The traditional manual system involves long queues, delays in verification, missing documents, and lack of transparency, leading to inconvenience for both applicants and officals. The proposed system will provide online application submission, document verification, appointment scheduling, status tracking, and report generation, ensuring faster processing, reduced manual errors, and better service to citizens.

1. Introduction:

1.1 Purpose of this Document:

The purpose of this document is to define the requirements for the Passport Automation System.. The system will help applicants apply for passports online, track the status of their applications, and enable government officials to process requests efficiently and securely.

1.2 Scope of this Document:

The PAS will allow users to submit applications online, upload necessary documents, pay fees digitally, and schedule appointments for verification. It will also help passport officials verify documents,

update status, and generate reports. The system will ensure faster, transparent, and error-free passport issuance

1.3 Overview:

The system will replace manual processing with an automated workflow, covering steps from application submission → verification → police check → passport approval/issuance. Users will get updates at every stage via SMS/email notifications.

2. General Description:

- Users: Applicants (citizens), passport office staff, Police Department, Administrators.
- Main Functions: Application submission, document verification, appointment scheduling, fee payment, and report generation.
- Benefits: Saves time, improves accuracy, increases transparency, and enhances user satisfication.

3. Functional Requirements:

FR1: The System shall allow applicants to register and log in securely.

FR2: The system shall allow applicants to fill out and submit passport applications online.

FR3: The system shall allow applicants to upload the required documents in digital format.

FR4: The system shall allow online fee payment through secure gateways.
FR5: The system shall notify applicants about application status via email/sms.
FR6: The system shall allow integration with police databases for background checks.
FR7: The system shall generate reports.
FR8: The system should allow biometric verification for added security.

4. Interface Requirements:

- User Interface: web-based portal and mobile app for applicants, admin dashboard for officials.
- Software Interfaces: Integration with payment gateways, police databases, and notification APIs.
- Hardware Interface: Biometric devices, printers for passports/receipts.

5. Performance Requirements:

- The system shall process each application submission within 5 seconds.
- The system shall handle at least 1,000 concurrent users.
- The system shall ensure 24/7 availability with 99.9% uptime.

6. Design constraints:

- The system shall comply with government data protection laws.
- The system shall encrypt all personal and financial information.

- The system shall support both cloud-based and on-premises deployment.

7. Non-Functional Attributes:

- Security: Strong authentication, encrypted, & access control.
- Reliability: Continuous availability & error-free data handling.
- Scalability: Support for increasing number of users nationwide.
- Maintainability: Easy upgrades to meet future policy changes.
- Transparency: Applicants should be able to track progress in real-time.

8. Preliminary Schedule & Budget:

- Timeline: Around 5-6 months (Requirement analysis - 3 weeks, Design - 4 weeks, Development - 10 weeks, Testing - 4 weeks, Diployment - 2 weeks)
- Budget: Estimated $80,000 - $1,20,000 (depending on biometric integration & nationwide

28/8/25