

Hotel Management System

Problem Statement

- The hotel management wants to create a system to manage various customer actions like check in, check out, Price Calculations, etc. For their hotel, to make the system automated and easier to tabulate.

→ SRS Document

1. Introduction

1.1 Purpose of the document

This document is to highlight the requirements and features expected from a hotel Management System. It also describes the scope and assumptions of the system.

1.2 Scope of the document.

The Hotel Management system is expected to cater to a single hotel, and automate all its functions. It is expected that management intervention varies from less to none. The users interacting with the system are administrators, customers, staff and potential customers who are required to register.

1.3 Overview

The system is an end to end application that governs all processes which require interaction between the hotel and a customer. The actions include reservation, Check in, Check out, Price calculation.

and any other required.

2. General Description

2.1 Objective

The system aims to allow user to make a reservation, check in, check out, and make payment. All these activities are to be automated without human intervention.

2.2 Users

- Potential Customers (Un registered)
- Customers
- Staff
- Administrators

2.3 Benefits

Automation of the process leads to less human effort and hence applying more effort in areas that cannot be easily automated, like cleaning, Food management, etc.

3. Functional Requirements

3.1 User Authentication

- ~~Users~~^{Customers} are to be registered before a reservation is made. If no reservation is made, customers are to be registered at check in.
- Customers are to be authenticated for every other step.
- Admins are required to assist customers, and hence must be authenticated as a separate from customers.



3.2 Reservation

- Customers (or Admins) can make a reservation for a given valid time period from a start date to an end date.

3.3 Check in

- Customers (or Admins) can check in and hence block the room from being visible to others.

3.4 Check Out

- Customers (or admins) can check out and allow their room to be allocated to other customers.

3.5 Bill Payments

- After check out, the system should generate a bill depending on prices and duration of stay.
- Customer can pay a bill generated for them.

3.6 Reports

- Admins can generate occupancy and financial reports.

4 Interface Requirements

- An interface is required for customers to make reservations, check in/out, and complete payments.
- Admins need interface to view reports and assist customers.

5. Performance Requirements

- System should be able to handle 100 concurrent

users without decline in performance.

- It should be able to handle 500 transactions per minute.
- The reservation system should accommodate 200 concurrent users at a time.
- In case of failure, the system should be able to recover in no more than 30 minutes.

6. Design Constraints

- Technology: Being web based, the system uses a server based on Node.js and related technologies.
- Scalability: Addition of new subsystems must be supported in the future.

7. Non-Functional Attributes

- Security: Payment Gateway should be secure and isolated.
- Interface should be simple and intuitive.
- Uptime should be high.

8. Preliminary Schedule and Budget

The development of the system is estimated to be complete in about 15 weeks. The major milestones are requirements analysis (3 weeks), system design (2 weeks), Development (6 weeks), testing (2 weeks) and deployment (2 weeks).

✓ 9/8

Credit Card Automation System

→ Problem Statement.

- The Credit Card Management wants to make a system to automate transaction and bill processing for their product, to make company's human effort minimal.

→ SRS Document

1. Introduction

1.1. Purpose

This document specifies the requirements for the Credit Card Automation System. The system aims to provide customer with a secure and efficient platform to apply for credit cards, view and manage transactions, receive billing statements, and make payments. It will also allow administrators and staff to manage approvals, monitor fraud and generate reports.

1.2 Scope

The system will be a web and mobile based application integrated with the bank's existing infrastructure. It will support customer authentication, online applications, automated billing, fraud detection and reporting.

1.3 Overview

The system is made to replace the manual process in the existing system. Here, request processing isn't stalled by human hands, but is verified and completed entirely by technology. The system is designed to be implemented in steps, eventually replacing human effort entirely.

2. General Description

2.1 Objective

The system will integrate with the bank's core banking system, replacing the manual credit card management with a central automated platform.

2.2 Users

- Customers
- Bank Staff
- Admin

2.3 Benefits

The system intends to make processing seamless and quick, while reducing the human hours required.

3 Functional Requirements

- 3.1 The system shall allow customers to apply for credit cards with KYC verification.
- 3.2 The system shall allow staff to approve/reject applications based on eligibility.
- 3.3 The system shall track all credit card transactions in real time.
- 3.4 The system shall generate monthly billing statements automatically.
- 3.5 The system shall allow secure online payments with OTP verification.
- 3.6 The system shall allow admins to generate financial and fraud reports.

4. Interface Requirements

- An interface is required for customer to complete online transactions, and observing their reports/statements.
- An interface is required for the Bank staff to generate or view applications.

PAGE :
DATE :

- An interface is required for admins to generate and view financial and fraud reports.

5. Performance Requirements

- The system should be able to handle 10000 concurrent transactions without decline in performance.
- It should be able to complete 1 million transactions in 5 minutes.
- The system shall maintain 99% uptime with recovery time of maximum 15 minutes.

6. Design Constraints

- The system must follow RBI regulations and other International standards.

7. Non-Functional Requirements

- Security: Must comply with encryption standards for sensitive data.
- Usability: Intuitive User experience.
- Reliability: 99% uptime for unimpeded financial transactions.
- Scalability: Should be able to expand concurrent users limit in the future.
- Performance: Transactions should be processed in under 5 seconds.

8. Preliminary Schedule and Budget

The system is expected to be developed over 16 weeks which includes requirements and analysis (2 weeks), system design (2 weeks), development (8 weeks), testing and quality assurance (2 weeks) and deployment (2 weeks).

Library Management System

→ Problem Statement

The library management System aims to replace inefficient manual processes of tracking books, members and transactions by providing functionalities like searching books, issuing/returning, tracking due dates and managing fines, to reduce paperwork and improving efficiency.

→ SRS

1. Introduction

1.1 Purpose of the document

This document defines the requirements of the Library Management system, which aims to automate various common library tasks.

1.2 Scope

The library management system is a desktop based application to be accessed by librarians, members and admin. Its key functionalities are book keeping record, tracking of due dates, fine calculation, etc.

1.3 Overview

The system is intended to automate repetitive manual tasks like checkin and out of books by members, and mathematical calculations like due date and fine calculations. The system is to be implemented in local computers only, and is operated on location. So, the hardware is limited to single network.



2. General Description

2.1 Objective

The system will act as a centralised system replacing manual registers and records. It will store details of books, members, and transactions in a database accessible through simple interface.

2.2 Users

- Members: Customers that borrow and return books
- Librarians
- Admins.

2.3 Benefits

The system reduces the human hours spent on repetitive tasks of scanning books for issuing and returning of books. Furthermore, it improves efficiency and decreases error in arithmetic calculations.

3 Functional Requirements

3.1 User Management

- The system shall allow librarians and admins to register new members with new IDs.
- The system shall allow login access for members to view issued books and due dates.
- The system shall allow admins to manage librarian accounts

3.2 Catalog Management

- ✓ • The system shall allow librarians/admins to add new books with details (title, author, ISBN, category).
- The system shall allow updates to book availability.

PAGE :
DATE :

- The system shall allow search by title, author, ISBN or category.

3.3 Transaction Management

- The system shall allow librarians to issue books to members with a due date.
- The system shall allow members to return books and automatically update availability.
- The system shall calculate late fees for overdue books.

3.4 Reports and Analytics

- The system shall generate reports on book inventory.
- The system shall generate reports on member activity.
- The system shall allow admins to generate monthly / annual usage and financial reports.

4 Interface Requirements

- An interface is required for members to return books and view / pay late fine.
- An interface is required for librarians to issue books, view member activity and set due dates.
- An interface is required for admins to generate reports.

5. Performance Requirements

- The system should be able to handle 500 users without decline in performance.
- The system must process issuing / returning within 2 seconds.
- The system should be able to generate reports from 10000 records in 20 - 30 seconds.

6. Design Constraints

The system must comply with institutional policies on data

protection. It must be compatible with standard hardware like barcode scanners.

7. Non-Functional Requirements

- Security: Password-protected access and encrypted member data.
- Usability: Simple, intuitive interface for both staff and members.
- Performance: Ability to handle at least 500 concurrent users.
- Reliability: 99% uptime and quick recovery.

8. Preliminary Schedule

The system is expected to be completed in about 12 weeks. The phases include requirements gathering, system design, development, testing and deployment.

~~✓ 20/08~~

Stock Maintenance System

→ Problem Statement

The Stock maintenance System aims to replace manual and error prone methods of tracking with an automated solution that manages stock levels, purchases, sales, and reporting to ensure accuracy, efficiency and real time visibility.

→ SRS

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements for the stock maintenance system. The system will automate stock trading, inventory updates, supplier management, and supplier sales monitoring to improve operational efficiency.

1.2 Scope

The system will be a desktop or web-based application designed for small to medium businesses. It will support functionalities like stock entry, stock updates, sales processing, purchase tracking, supplier management, and report generation.

1.3 Overview

The system is aimed to replace human effort with an efficient system that is less error prone. Being a fast and often mathematically intensive, the chances of events like overstocking and stock outs are very much possible. The system will be able to maintain the inventory such that such events are minimized.

2. General Description

2.1 Objective

The system will act as a centralized system replacing many registers or spreadsheets. It will store all stock data in a secure database and provide access through an easy-to-use interface.

2.2 Users

- Admin: Manages the system, users and overall inventory.
- Staff/User: Enter Stock details, record sales and update inventory.
- Supplier: Provides stock replenishment system.

3. Functional Requirements

3.1 User Management

- The system shall allow admins to create, update and delete staff accounts.
- The system shall enforce secure login and role-based access.

3.2 Stock Management

- The system shall allow staff to add new items with details.
- The system shall automatically update stock levels after each purchase or sale.
- The system shall allow adjustments for expired, damaged, or returned goods.

3.3 Purchase and Supplier Management.

- The system shall allow recording of purchase orders with supplier details.
- The system shall maintain supplier records.
- The system shall track outstanding purchase orders and expected delivery dates.



3.4

Sales Management

- The system shall record each sale and automatically deduct quantities from stock.
- The system shall generate invoices/reciepts for each sale.
- The system shall maintain daily, weekly and monthly sales reports.

3.5

Reporting and Analytics

- The system shall provide stock reports.
- The system shall generate sales reports by item; category and ~~prod~~ period.
- The system shall provide supplier performance reports.

4.

Interface Requirements

- An interface is required for staff to edit the stock reserve database
- An interface is required for admin for report generation.

5.

Performance Requirements

The stock system must process stock updates in under 2 seconds, generate reports for 100000 records within 8 seconds, and support 200 concurrent users.

6.

Design Constraints

The system must run on commonly used operating systems and must be compatible with existing hardware like barcode/RFID scanners.

7.

Non-Functional Requirements

- Security: Encrypted data storage.
- Performance: Handle at least 1000 stock updates per minute.

- Usability: Simple UI for non-technical staff
- Scalability: Should support growing stock sizes and more warehouses.

8. Preliminary Schedule and Budget.

The system is expected to be developed in 12 weeks including requirement analysis, system design, development, testing, deployment and staff training. The cost of development estimated at ₹500000.

Passport Authentication System

→ Problem Statement

The passport Authentication System aims to replace manual verification processes with an automated platform that securely manages passport applications, identity verification and approval.

→ SRS

1. Introduction

1.1 Purpose

The document defines the requirements for the passport application processing, identity verification and document validation.

1.2 Scope

The system will be a secure web-based application used by government staff, verification officers and applicants.

2. General Description.

2.1 Objective

The system will serve as a centralized solution integrating identity documents, biometric data and verification workflows, replacing paper based procedures.

2.2 Users

- Applicant
- Officer
- Admin

3. Functional Requirements

- 3.1 Applicant Registration and Profile Management.
- 3.2 Passport Application Submission.
- 3.3 Identity verification and authentication
- 3.4 Approval
- 3.5 Security and Compliance to norms.
- 3.6 Reporting.

4. Interface Requirements.

- An interface is required for applicants to submit and view documents.
- An interface is required for officers to view applicant documents.
- An interface is required for admins for report generation

5. Performance Requirements

The system must process application submissions within 5 seconds, identity verification within 10 seconds, and support 1000 concurrent users.

6. Design Constraints.

The system must comply with government identity laws and data protection standards.

7. Non-Functional Requirements.

- Security: Enforce encryption and multi-factor authentication.
- Performance: Process 1000 applications per hour.
- Reliability: Should have 99% uptime.
- Usability: Interactive and user friendly UI.
- Scalability: Support large-scale deployment.

8. Preliminary Schedule and Budget.

The system is expected to be developed in 20 weeks, with estimated budget of ₦2000000.