

19/8/25

1) HOTEL MANAGEMENT SYSTEM

Problem statement: \rightarrow (Title)

Manual hotel management results in inefficiencies like double bookings, delayed check-ins, billing errors and poor record management.

Traditional systems fail to provide real-time updates on room availability, integrated payment handling which impacts customer ~~sor~~ satisfaction and hotel staff productivity.

To address these challenges, a computerized hotel management system is needed to automate bookings, generate accurate billing and handle payments.

Such a system will enhance operational efficiency minimize human errors and improve customer experience.

Software Requirements Specification (SRS)

Title: Hotel Management System

1 Introduction

1.1 Purpose of the document -

This document specifies the requirements for a Hotel Management system. It will guide developers, testers, and stakeholders in designing and implementing a centralized system to handle hotel operations, including reservations, check in/check out, billing, staff management, and reporting.

1.2 Scope of the document -

The Hotel Management System is a web-based and desktop-compatible application that automates hotel operations. It supports:

- Room reservation and availability management
- Customer check in/check out
- Online and offline payments
- Automated invoice generation

The system aims to improve efficiency, reduce manual errors and enhance guest experience.

1.3 Overview

- System type: Web + Desktop application
- Users: Hotel receptionists, Managers, Guests
- Key functions: Reservation, Room Allocation, Billing, Staff Scheduling, Reports
- Benefits: Reduced workload, better resource utilization

improved customer satisfaction

2. General description

The HMS replaces manual/paper-based hotel operations with a fully automated digital solution. It integrates with payment gateways, SMS/email services for booking confirmations, and third-party travel sites.

3. Functional Requirements

FR1. The system shall allow users to search and book available rooms.

FR2. The system shall manage room check-in and check-out.

FR3. The system shall generate invoices automatically.

FR4. The system shall process payments via cash, card or online gateways.

FR5. The system shall maintain customer records and history.

4. Interface requirements

- User interface: Intuitive dashboard with login based on roles
- Hardware interface: PCs, tablets, POS machines, printers for receipts.
- Software interface: Database (MySQL), Payment APIs
- Communication interface: Internet (for online booking)

5. Performance Requirements

- The system must be able to handle 500 concurrent guest sessions
- The system must provide search results for room availability in minimum 2 seconds.
- The system shall ensure 99.5% uptime

6. Design constraints

- Developed using Java / Python with SQL backend
- Must comply with hotel data privacy
- Cross - browser compatibility

7. Non functional attributes

- Reliability : 24/7 availability with redundancy
- Security : Role-based access control, encrypted payments
- Usability : Simple UI, Multilingual support
- Scalability : Capable of supporting multiple hotel branches.

8. Preliminary schedule and budget

Schedule for 6 months: weeks

- i) Month 1 : Requirement gathering and analysis
- ii) Month 2 : System design (UI)
- iii) Month 3 & 4 : Development of core modules
- iv) Month 5 : Integration and testing
- v) Month 6 : Deployment.

Job No

Budget :

- Development team : \$60,000
 - Hardware and infrastructure : \$15,000
 - Licensing and APIs : \$5,000
 - Training and maintenance (first year) : \$10,000
- Total estimated budget : \$90,000

Table

An

2. CREDIT CARD PROCESSING

Problem Statement:

Manual and outdated credit card processing methods cause delays, transaction errors, fraud risks. Customers demand quick, secure and reliable payment options, while businesses require accurate transaction tracking and fraud prevention.

A credit card processing system is needed to securely handle authorization, authentication, transaction recording, and settlement in real-time. The system should integrate with banking networks, support multiple payment gateways, detect fraudulent activity and provide detailed transaction reports. This will ensure faster and safer transactions.

Software Requirements Specification (SRS)

Title: Credit Card Processing System

1. Introduction

1.1 Purpose of the document

The purpose of this document is to define the requirements for the credit card processing system. It provides a detailed description of system functionalities, interfaces, performance requirements and constraints.

1.2 Scope

The CCPS is a secure payment solution designed to handle real-time credit card transactions for businesses and financial institutions. It supports card authorizations, authentication, transaction recording, fraud detection.

Key objectives:

- Secure and efficient transaction processing
- Integration with multiple banks and gateways.
- Fraud detection and prevention mechanisms

1.3 Overview

- System Type: secure web + backend server application
- Users: Merchants, customers, banks
- Key functions: Authorization, authentication, transaction recording
- Benefits: Reduced fraud risk, faster transactions, better record mgmt, enhanced customer experience

2. General description

The CCPS will operate as middleware between merchants, cardholders, and banking/payment networks. It will integrate with merchant POS systems, online shopping portals and financial institutions.

3. Functional requirements

FR1. The system shall accept and validate credit card numbers, expiry dates and CVV codes.

FR2. The system shall authenticate transactions using OTP, PIN or biometric verification.

FR3. The system shall authorize transactions through bank/payment gateways in real-time.

FR4. The system shall record all transactions in a secure database.

FR5. The system shall detect and flag suspicious transactions.

4. Interface requirements

i) User interface :

i) Customer - Simple form to enter card details and receive OTP confirmation.

ii) Merchant : Dashboard for transaction reports and refunds.

iii) Admin : Secure dashboard for monitoring fraud and logs.

- Hardware interface - POS terminals, card readers, merchant computers and mobile devices
- Software ~~repositor~~ interface - Banking APIs, payment gateways, SMS/ Email services
- Communication interface - Secure internet ~~sec~~ Encrypted communication with banks

5. Performance requirements :

- The system shall process at least 1000 transactions / second.
- The system shall complete a transaction within 3 seconds
- The system shall provide 99.9% uptime.

6. Design constraints

- Must comply with PCI DSS, EMV
- Must use TLS/SSL encryption for data transfer
- Should support multi currency transactions.

7. Non functional attributes

- Security : End to end encryption
- Reliability : Transaction logs with recovery options.
- Usability : Intuitive interface

8. Preliminary schedule and budget:

Schedule (8 months)

Month 1-2 : Requirement gathering

Month 3-4 : System design

Month 5 : Integration with payment gateways

Month 6: Fraud detection module implementation

Month 7: System testing

Month 8: Deployment

Budget

- Development team: \$120,000
 - Hardware and infrastructure: \$40000
 - Licensing and APIs: \$30000
 - Security and maintenance: \$25000
- Total estimated budget: \$215,000

3. LIBRARY MANAGEMENT SYSTEM

Problem Statement:

A Library Management System is required to automate book cataloguing, borrowing, returning and record management to reduce manual errors and improve efficiency.

Software requirement system (SRS)-

Title: Library Management System

1. Introduction

1.1 Purpose of the document

The purpose of the document is to define the requirements of the Library Management System. It serves as a guide for developers, testers, librarians, and administrators to design, implement and maintain as a system that automates library operations such as cataloging, borrowing, returning and fine management.

1.2 Scope

The LMS is a digital solution that manages a library's resources efficiently. It will handle book cataloging, member registrations, borrowing/return transactions, fine calculations and generate reports for administrators. The system will provide role-based access for students, librarians and administrators.

2. General description

The library management system automates library operations like cataloging, member registration, borrowing, return, fine calculation. It provides real-time book availability, search functionality, and reporting tools, ensuring efficient and secure access for students, librarians and administrators.

3. Functional Requirements

FR1. The system shall allow librarians to add, update and delete book records.

FR2. The system shall allow students to search for books by title, author and or subject.

FR3. The system shall allow members to borrow and return books.

FR4. The system shall automatically calculate fines for late returns.

FR5. The system shall maintain a borrowing history for each member

FR6. The system shall support role-based access.

4. Interface Requirements

• User interface -

Student - Search catalog, view borrowing history

Librarian - Issue/ return books, manage inventory

Admin - Generate reports, configure system.

• Hardware interface - Barcode / RFID scanners.

• Software interface - Database (MySQL), library

catalog APIs.

- Communication Interface - LAN (local operations), Internet (remote access)

5. Performance Requirements

The system shall support at least 200 concurrent users.

The system shall process borrowing/return transactions within 2 seconds.

The system shall support a database of at least 100000 book records.

The system shall provide 99% uptime.

6. Design constraints

- Must support common OS
- Developed using Java with SQL backend.
- Comply with institutional IT policies for data security.

7. Non functional attributes

member

- Security - Role based access control, encrypted data ~~data~~
- Usability - Simple and intuitive interface for all users.
- Scalability - Support expansion for multi-branch libraries.

8. Preliminary Schedule and Budget

Phase	Duration	Activities
• Req. analysis	2 weeks	Gather requirements, feasibility study
• System design	2 weeks	Database design, UI/UX design
• Development - Core modules	4 weeks	Cataloging, member registration, borrowing/return
• Development - Add ons	2 weeks	Fine calculation, reporting tools
• Integration & testing	2 weeks	Module integration, unit/system testing
• Deployment & training	2 weeks	Installation, user training, final adjustments

Total duration: 14 weeks

Budget

Category	Estimated cost (USD)
• Development team (3 people x 14 weeks)	\$45,000
• Hardware & Infrastructure (servers, scanners)	\$10,000
• Software licensing & APIs	\$5000
• Training & Maintenance (first year)	\$7000

Total budget: \$67,000

4. STOCK MAINTENANCE SYSTEM

Problem Statement -

Manual stock management often results in errors like in tracking, overstocking and stockouts, so a Stock Maintenance System is needed to automate inventory recording, monitoring and reporting.

Software Requirements Specification
Title : Stock maintenance system

1. Introduction

1.1 Purpose

This document defines the requirements for the Stock Maintenance system. It provides details for developers, testers, administrators and business stakeholders to design, implement and maintain a reliable system that manages inventory efficiently.

1.2 Scope

The SMS is a web and desktop-based system designed to automate inventory management for businesses. It will track stock levels, handle purchases and sales records, send alerts for stock shortages or surpluses and generate reports. The system will improve efficiency, reduce errors and support decision making for managers.

2.3 Overview

- System type: Web + desktop application
- Users: Storekeepers, Managers, Administrators
- Functions: Stock entry, update tracking, alerts, reporting.
- Benefits: Reduced stock errors, improved visibility, better resource allocation utilization

2. General description

The SMS automates inventory operations by recording incoming and outgoing stock, updating stock levels in real-time, and generating alerts for low or excess stock. It supports role-based access for storekeepers and managers, provides detailed reporting on sales and stock levels, and ensures efficient and error-free management of inventory.

3. Functional requirements

FR1. The system shall allow entry and update of stock items

FR2. The system shall record incoming and outgoing stock

FR3. The system shall display real-time stock levels.

FR4. The system shall send alerts when stock is low or exceeds defined thresholds

FR5. The system shall support role based access.

4. Interface requirements.

- User requirement interface

Storekeeper: Stock entry/update

Manager: Reporting, alerts

Admin: User and system configuration

- Hardware interface : Barcode/RFID scanners, label printers
- Software interface : Database (MySQL), reporting tools
- Communication interface : LAN for internal use, Internet for remote access.

5. Performance Requirements

Supports at least 500 concurrent stock transactions per day.

Update stock levels in real-time

Generate reports for upto 5 years of stock data within 5 seconds.

Maintain 99% uptime for daily operations.

6. Design Constraints

Must run on common OS

Developed with Java and SQL backend

Should integrate with barcode/RFID hardware

7. Non functional attributes

Security - Role based access, encrypted records

Usability - Easy to use UI for non-technical staff.

Scalability - Support expansion for multiple warehouses

8. Preliminary schedule and budget

Schedule	Duration	Activities
Phase		
Requirement analysis	2 weeks	Reqn gathering, feasibility study
System design	2 weeks	Database schema, design
Development - core modules	4 weeks	Stock entry, update tracking
Development - Alerts / Reports	2 weeks	Low stock alerts, reports
Integration & Testing	2 weeks	Integration, unit/system testing
Deployment & Training	2 weeks	Installation, user training

Total duration: 14 weeks

Budget	Estd cost (USD)
Category	
Development team (3×14)	\$48000
Hardware & infrastructure	\$12000
Software licensing and APIs	\$5000
Training & Maintenance	\$8000

Total estimation: \$73000

5. Passport Automation System

Problem Statement :

A Passport Automation system is required to digitize application submission, verification, status tracking and delivery to improve efficiency, reduce errors and enhance citizen convenience.

Software Requirements Specification

Title : Passport Automation System

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements of the Passport Automation System. It describes the functionalities, performances, constraints, and design needs for developers, testers and government authorities responsible for implementing and maintaining the system.

1.2 Scope

The PAS is an online system that automates the entire passport lifecycle, including new applications, renewals, ~~and~~ verification, appointment scheduling, payment processing and status tracking. The system will reduce manual workload, minimize delays, prevent errors and ensure secure data handling.

1.3 Overview

- System type: Web + Mobile application
- Users: ~~to~~ Citizens, Passport officials, Administrators
- Functions: Application submission, document uploads, verification, fee payment, appointment booking, status tracking

2. General description

The passport automation system digitizes passport application and renewal processes by allowing citizens to apply online, upload documents, pay fees and track status in real time.

3. Functional Requirements

FR 1. The system shall allow users to register and apply for a new passport or renewal.

FR 2. The system shall allow uploading and validation of reqd. documents

FR 3. The system shall enable online fee payment via secure gateways.

FR 4. The system shall allow users to book and manage appointments

FR 5. The system shall update and display application status in real-time.

4. Interface requirements

• User interface:

Citizen: Application form, doc upload, status tracking

Passport official : Verification dashboard, document management

Admin : Report generation, user management

- Hardware interface : Computers, biometric scanners, printers
- Software interface : Payment gateways, national ID databases, verification APIs.

5. Performance Requirements

- The system shall support 5000+ concurrent applications
- It shall process application submissions within 3s.
- It shall generate reports for up to 1 million records in under 10 seconds
- It shall maintain 99.9% uptime.

6. Design constraints

- Must comply with govt data privacy and security regulations.
- Developed using scalable technology.
- Must support biometric integration

7. Non functional attributes

- Security : Data encryption, role based access, audit logging
- Reliability : Failover servers and backup mechanism
- Usability : Simple, multilingual and accessible interface

8. Preliminary Schedule and Budget

Schedule

Phase	Duration	Activities
• Requirement Analysis	2 weeks	Collect requirements from stakeholders
• System design	3 weeks	Database schema, workflow diagrams, UI mockups
• Development - Core modules	5 weeks	Application, document upload, payment
• Development -	3 weeks	Official dashboard, biometric integration
• Verification		
• Integration & Testing	3 weeks	Security, load and user acceptance testing
• Deployment & Training	2 weeks	Production rollout, staff training
Total duration: 18 weeks		

Budget

Category	Estd cost (USD)
• Development team (5x18)	\$ 150000
• Hardware & infrastructure	\$ 50000
• Software licensing & APIs	\$ 25000
• Training and maintenance	\$ 20000

Total estimation: \$245000