

LAB PROGRAM - 1

SOFTWARE REQUIREMENT SPECIFICATION (SRS) FOR HOTEL MANAGEMENT SYSTEM

1 Introduction

1.1 Purpose of the Document :- The purpose of the document is to define the requirements for Hotel Management System (HMS). It will serve as a reference for developers, testers, project managers and stakeholders to understand the scope, features & constraints of the system. It automates operations like booking, check-in/out, billing, staff management & reporting.

1.2 Scope :- HMS enables

- Guests : online/offline room booking & billing
- Receptionists : check-in/out, reservations
- Admin : room/staff management and reports
- Staff : update room/housekeeping status

It improves efficiency, reduces manual errors & enhances service quality.

1.3 Overview :- A centralized desktop/web-based application with modules for booking, billing, customers & staff management & reports. Secure database and role-based access are ensured.

② General Description :- The Hotel Management System will cater to the needs of hotel staff & management, providing features such as room booking, guest profiles, inventory management & financial reporting. It will be accessible to users with varying levels of technical expertise.

3. Functional Requirements :-

3.1 Reservation Management :

- Allow users to make room reservations online or through the front desk.
- Generate reservation confirmations & send notifications to guests.

3.2. Room Management :

- Assign rooms to guests based on availability & preferences.
- Track room status (clean, occupied, vacant) in real-time.

3.3 Guest Management :

- Maintain guest profiles with personal information, preferences & booking history.
- Facilitate guest check-in & check-out processes.

3.4 Billing & Invoicing :-

- Generate accurate bills for room charges, additional services & taxes.
- Accept various payment methods & generate invoices for corporate clients.

4. Interface Requirements :-

4.1 User Interface :

- Intuitive & user-friendly interface for hotel staff and guests.
- Accessible via web browsers, mobile devices, & desktop applications.

4.2 Integration Interfaces :

- Integration with payment gateways for secure transactions.
- Integration with third-party booking platforms for seamless reservation management.

5. Performance Requirements :

5.1 ~~Response Time~~: The system should respond to user actions within 2 seconds.

5.2 Scalability :- Handle a minimum of 1000 concurrent users during peak hours.

5.3 Data Integrity : Ensure data consistency & accuracy across all modules

6. Design Constraints :

6.1 Hardware Limitations :

- The system should be compatible with standard hotel hardware (computers, printers, POS terminals)

6.2 Software Dependencies :

- Utilize a relational database management system (e.g., MySQL) for data storage.
- Use programming languages & frameworks conducive to UML modeling (e.g. Java Spring Boot)

7. Non-Functional Attributes :

7.1 Security : Implement robust authentication & authorization mechanisms to protect data

7.2 Reliability : Ensure high availability & fault tolerance to minimize system downtime

7.3 Scalability : Design the system to accommodate future growth & expansion

7.4 Portability : Support multiple platforms & devices for user accessibility

7.5 Usability : The system shall have a user-friendly interface with clear navigation

- 7.6 **Reusability** :- The system shall use modular code design to facilitate future enhancements & maintenance.
- 7.7 **Compatibility** :- The system shall be compatible with common web browsers (Chrome, Firefox, Safari).
- 7.8 **Data Integrity** :- The system shall ensure accurate & consistent data storage & retrieval.

8. **Preliminary Schedule and Budget** : The development of the HMS is estimated to take 6 months with a budget of \$100,000. This includes project planning, development, testing and deployment phases.

SOFTWARE REQUIREMENT SPECIFICATION (SRS) FOR CREDIT CARD PROCESSING

1 Introduction

1.1 Purpose of the Document :- The purpose of this document is to outline the requirements & specifications for the development of Credit Card Processing System (CCPS). This system will ensure secure, efficient and AI-assisted processing of credit card transactions for merchants, banks & customers.

1.2 Scope :- The CCPS will handle all activities involved in credit card payments, including authorization, authentication, clearing, settlement, billing & repositing. It will integrate with payment gateways, ensure compliance with PCI-DSS (Payment Card Industry - Data Security Standard) standards & use AI/ML techniques for risk scoring & anomaly detection to strengthen cybersecurity.

1.3 Overview :- The system provides a secure platform to process credit card transactions with features like tokenization, fraud risk scoring, encryption & customer billing. It is intended for merchants, financial institutions & cardholders.

2. General Description:- The system will allow merchants to accept credit card payments securely, banks to authorize/settle transactions, & cardholders to track their usage. AI/ML models will assist in dynamic credit risk scoring & transaction monitoring to reduce fraudulent or suspicious activities.

3. Functional Requirements:

3.1 Transaction Authorization:

- Validate customer credentials (PIN, CVV, OTP or biometric)
- Forward authorization requests to issuing banks securely.

3.2 Tokenization & Encryption:

- Replace sensitive card details with unique tokens for transactions.
- Encrypt all communication b/w cardholder, merchant & bank

3.3 Clearing & Settlement

- Perform batch processing of daily transactions
- Ensure accurate transfer of funds b/w banks & merchants.

3.4 Billing & Invoicing:

- Generate transactions invoices for merchants & cardholders.
- Support multiple currencies & international transactions.

3.5 Risk Scoring & Anomaly Alerts (AI/ML)

- Assign risk scores to each transaction based on amount, location & history
- Flag anomalies for manual review

3.6 Reporting & Analytics

- Generate reports for merchants (sales, trends, etc.)
- Provide banks with risk & fraud analysis dashboards.

4. Interface Requirements

4.1 User Interface

- Web portal for merchants & banks
- Mobile app for cardholders to view usage & billing.
- Real-time dashboards for risk monitoring

4.2 Integration Interfaces

- Integration with payment gateways (e.g., Visa, Mastercard, RuPay)
- API-based connectivity with banks & merchant systems.

5. Performance Requirements

5.1 Response Time :-

Transaction processing should not exceed 2-3 seconds.

5.2 Scalability :-

Support 10,000 + concurrent transactions.

5.3 Data Integrity :- Ensure accurate reconciliation between merchant, bank & cardholder records.

6. Design Constraints.

- Must comply with PCI-DSS standards.
- Use secure databases (e.g., PostgreSQL, Oracle) with encryption.
- Backend development in Java/Spring Boot or Python/Django.
- AI/ML modules for risk scoring must be lightweight for real-time use.

7. Non-Functional Requirements.

- 7.1 Security :- End-to-end encryption (TLS 1.3), two-factor authentication, tokenization.
- 7.2 Reliability :- 99.9% system uptime with backup servers.
- 7.3 Scalability :- Ability to integrate with new payment providers.
- 7.4 Usability :- User-friendly merchant & customer dashboards.
- 7.5 Portability :- Accessible via web, mobile & POS terminals.
- 7.6 Auditability :- Maintain detailed transaction logs for compliance.

8. Preliminary Schedule and Budget:

- 8.1 Estimated Duration :- 6-8 months
- 8.2 Budget :- \$120,000 (development, testing,
deployment, PCI certification)

SOFTWARE REQUIREMENT SPECIFICATION (SRS) FOR LIBRARY MANAGEMENT SYSTEM

1. Introduction

1.1 Purpose of the Document:- The purpose of this document is to outline the requirements & specifications for Library Management System (LMS).

This system is designed to automate & streamline the management of books, members, transactions & reporting within a library. It will provide efficient tools for librarians, administrators & students to access, borrow & return resources securely.

1.2 Scope:- The LMS will handle catalog management, user registration, book circulation, fine calculation & reporting. It will be accessible via web & mobile platforms & integrate with modern technologies such as barcode/RFID scanning for easy tracking.

1.3 Overview:- The system will allow

- Librarians to manage books, members & transactions
- Students/Users to search, reserve & borrow books
- Administrators to generate reports & manage system configurations

a) General Description :- The LMS provides a digital platform to replace traditional paper based library operations. It ensures accuracy, time-saving & improved user experience, while also securing book & member records.

3. Functional Requirements:

3.1 Book Management

- Add, update & delete book records
- Support ISBN, title, author, genre & publisher details.
- Track book availability (Issued, Reserved, Available)

3.2 User Management

- Register new members & maintain their profiles
- Assign unique Library ID to each member
- Track borrowing history

3.3 Borrowing & Returning

- Allow users to borrow & return books
- Set borrowing limits & due dates
- Automatically calculate fines for late returns

3.4 Reservations & Renewal

- Enable users to reserve books that are currently borrowed.
- Allow online renewal of borrowed books (if not reserved by others)

3.5 Search and Catalog Access

- Search books by title, author, subject or keyword
- Support advanced filtering (availability, category, year)

3.6 Reporting & Analytics:

- Generate reports on issued books, overdue returns & fines collected.
- Provide library usage statistics (most borrowed books, active members).

4. Interface Requirements

4.1 User Interface

- Web interface for librarians & administrators
- Mobile-friendly portal for students
- Barcode/RFID scanner integration for quick issue/return.

4.2 Integration Interfaces

- Integration with student/employee dbs (for institutions)
- API based integration for e-books or online resources.

5. Performance Requirements

5.1 Response Time :- Search and transaction

processing should occur within 2 seconds

5.2 Scalability :- Support atleast 10,000 books & 5,000 members.

5.3 Data Accuracy :- Ensure real-time update of book status across all modules.

6. Design Constraints

6.1 Database: MySQL / PostgreSQL for book & user records.

6.2 Programming:- Java/Spring Boot, Python/Django or PHP/Laravel.

6.3 Compatibility: must be compatible with barcode/RFID hardware.

6.4 Platform: Cross-platform (web + Mobile)

7. Non-Functional Requirements.

7.1 Security :- User authentication (username/password or Single Sign-On)

7.2 Reliability :- Ensure backup & restore options to prevent data loss

7.3 Scalability :- Easily accommodate more books, members & branches.

7.4 Usability :- Intuitive search and borrowing interface

7.5 Portability :- Accessible on desktops, laptops & smartphones.

7.6 Auditability :- Maintain detailed logs of all book & user transactions.

8. Preliminary Schedule and Budget

8.1 Estimated Duration :- 4 - 6 months

8.2 Budget :- \$50,000 (development, testing, deployment
& maintenance)

SOFTWARE REQUIREMENT SPECIFICATION (SRS) FOR STOCK MAINTENANCE SYSTEM

1. Introduction :-

1.1 Purpose of the Document :- The purpose of the document is to outline the requirements for the development of a Stock Maintenance System (SMS). The system is intended to help businesses maintain & monitor inventory, ensuring accurate records of available stock, sales, purchases & reorder levels.

1.2 Scope :- The SMS will manage product inventory, supplies, purchase orders, sales records and stock alerts. It will reduce manual errors, prevent overstocking/understocking & generate reports for decision-making.

1.3 Overview :- The system provides features for stock entry, updates, automatic low-stock alerts, billing, supplier management & reporting. It will be used by store managers, salespersons & administrators.

2. General Description :- The SMS provides a centralized solution for tracking & updating inventory across multiple departments or branches. It ensures transparency, reduces losses & improves business efficiency.

3. Functional Requirements

3.1 Product Management

- Add, update & delete product details (name, ID, category, price, quantity).
- Track stock levels in real time.

3.2 Supplier Management

- Maintain supplier profiles with contact & payment details
- Generate purchase orders & update stocks on delivery.

3.3 Stock Monitoring

- Track incoming & outgoing stock.
- Generate alerts when stock drops below reorder level.

3.4 Sales & Billing

- Process sales & update stock automatically
- Generate customer invoices.

3.5 Reporting

- Generate stock reports (daily, weekly, monthly)
- ~~Analyze sales trends & purchase history.~~

4. Interface Requirements

- Web-based application dashboard for managers & staff
- Barcode scanner integration for product entry
- Mobile app support for real-time monitoring

5. Performance Requirements

- 5.1 Response Time :- Updates must reflect within 2 seconds.
- 5.2 Scalability :- Support 50,000+ stock items
- 5.3 Accuracy :- Ensure 100% accuracy in stock updates.

6. Design Constraints

- 6.1 Database :- MySQL / PostgreSQL
- 6.2 Languages :- Java / Python / PHP
- 6.3 Must integrate with POS systems.

7. Non Functional Requirements

- 7.1 Security: Role-based access control (admin, staff).
- 7.2 Reliability: Backup & restore options
- 7.3 Usability: Intuitive dashboards and alerts.
- 7.4 Portability: Web & mobile support.
- 7.5 Auditability: Maintain logs of all stock transactions.

8. Preliminary Schedule and Budget

- Duration :- 4-5 months
- Budget :- \$ 40,000

SOFTWARE REQUIREMENT SPECIFICATION (SRS) FOR PASSPORT AUTOMATION SYSTEM.

I. Introduction

- 1.1 Purpose of the Document :- The purpose of the document is to specify the requirements for a (PAS). The system will digitize and streamline the process of passport application, verification & issuance. It ensures transparency, reduces manual workload, improves processing speed & strengthens security.
- 1.2 Scope :- The PAS will provide services for citizens, passport officials & police departments. Citizens can apply online, schedule appointments, pay fees, & track status. Officials will verify documents, conduct approvals & issue passports. Police will update verification status directly on the system.
- 1.3 Overview :- The PAS integrates multiple stakeholders (citizens, police, passport officers) onto a single platform, ensuring faster processing, secure data handling & end-to-end automation of passport services.

Q. General Description :- The system will replace traditional paper-based operations with a web-based & mobile-friendly solution. It will include features such as online applications, appointment scheduling, digital verification, online fee payment & automatic status update.

3. Functional Requirements

3.1 User Registration & Application

- Citizens can register with unique credentials.
- Submit passport applications with personal & document details.

3.2 Appointment Scheduling

- Select preferred dates & centers for passport verification.
- Receive automated SMS /email confirmations.

3.3 Document & Police Verification

- Passport officers check uploaded documents for validity.
- Police log into the system to provide background verification ~~repository~~.

3.4 Fee Payment

- Secure online payment gateway integration.
- Generate receipts for applicants.

3.5 Status Tracking

- Real-time status updates (Application Submitted → Verified → DPS patched).
- Notifications for every status change.

3.6 Passport Issuance

- Generate passport records after successful verification.
- Maintain digital db of all issued passports.

4. Interface Requirements :-

4.1 Citizen Interfaces:- Web + mobile application for application, payment & tracking.

4.2 Official Interfaces:- Portal for document and police verification.

4.3 Integration Interfaces:- API's to connect with national ID db's & police verification systems.

5. Performance Requirements :-

5.1 Response Time :- < 3 seconds per request

5.2 Scalability :- Must support 100,000 + applications per month.

5.3 Accuracy :- Ensure 100% correctness in data storage & status updates.

6. Design Constraints:

- Must comply with government data security laws.
- Backend database :- Oracle / MySQL with encryption.
- Platform :- Cross-platform (web, mobile)

7. Non-Functional Requirements:

7.1 Security :- Biometric /OTP authentication, encrypted storage.

7.2 Reliability :- High availability (99.9% uptime)

7.3 Usability :- Multi-language support.

7.4 Portability :- Works across desktops, laptops, and smartphones.

7.5 Auditability :- Maintain detailed audit logs for transparency.

8. Preliminary Schedule and Budget:

The total estimated budget is \$100,000.

~~2020~~