Software Requirement System

(1) Hotel Management System

- Introduction

1.1 Purpose of this document

- This purpose of this document is to provide a clear, consise, and detailed Software Requirement Specification (SRS) for the Hotel Management System. This document defines the functionality, features, constraints and general system requirements for Hotel Mangement System to ensure the succesful development and implementation of the system.

1.2 Scope of the document

- This document covers all essential information releated to the development of Hotel Management System, including functional, nonfunctional requirements, user interfaces, design constraints and other system attributes. The HMS will be designed to automate Hotel operations such as room book, checkin, checkout, billing, payment process, customer management, and report generation. System will also manage guest reservation, room availability, staff allocation and payment transactions.

1.3 Overview

- The hotel management system is designed to automate hotel operations such as ~~room booking~~ or manage guest reservations, room availability, staff allocation.

Systems will include an interface for generating reports, managing staff activities, and maintaining accurate record of all transactions.

The product will include:
(1) Web based User Interface: Accessible by hostel staff, manager and guests
(2) Backend system: A database for storing booking, customer, staff and payment details.
(3) Admin Panel: for management and administrative functions

2. General Description
2.1 General Functions
- Allow guest to book rooms online or offline
- Enable staff to manage reservation, checkin & check
- Manage room availability and status
- Handle billing, invoicing and multiple payment, repo

2.2 User Characteristics
The HMS will have three main types of users:
- Admin: full system control, including managing rooms, staff reports
- Receptionist/Staff: Handles checks in, check-outs, bookings and payments
- Guest: Can search for available rooms, make bookings and view payment details.

2.3 Feature and Benefits:
Room Search and Booking
Automated Check In/Out
Biling and Payment Management

Customer Management and Report Generation
Benefits: Increased operational efficiency, reduce human error.

3. Functional Requirements

3.1 User Registration and Authentication
Requirement: User must be able to register and log into the system.
Outcome: Users can access role-based features (e.g booking, checkin)

3.2 Room Management
Requirement: Admins and staff can manage rooms
Outcomes: Accurate, upto date room availability
Details: Rooms will have attribute such as type, price, capacity and status.

3.3 Booking and Reservation Management
Requirement: Track and manage bookings in real-time.
Outcome: Prevent double booking and maintains accurate room allocation.
Details: System updates availability, instantly upon booking or cancellation.

3.4 Check-In and Check Out
Requirement: Manage guest arrivals and depatures efficiently.
Outcome: Accurate occupancy records and guest experience.
Details: System updates room status automatically

3.5 Billing and Payment
Requirement: Generate bills and process payments.
Outcome: Accurate financial records and faster transactions.
Details: Supports multiple payment modes (cash, card, online).

4. Interface Requirements

4.1 Software Interfaces

- Database: Stores room, booking, and user data
- Admin Interface: Web based dashboard
- Guest Interface: Web/mobile interface for booking and payment.

4.2 Communication Interfaces

- Web Interface: Uses HTTP/HTTPS protocol
- Data Streams: Rest APIs between frontend and backend.

5. Performance Requirements

5.1 System Response Time

- Search results for available rooms within 2 sec
- Check in/out and payment: within 5 seconds

5.2 Scalability

Supports up to 1000 concurrent users without performance loss.

5.3 Database Performance

Optimized queries for fast booking, guest lookup and transaction processing.

6. Design Constraints

6.1 Technology Constraints

- Developed using HTML5, CSS3, Javascript for frontend.
- Backend using Python (Django) and Java
- Compatible with Chrome, Firefox and edge

6.2 Hardware Constraint

- Minimum Server: 8GB Ram, 1TB storage

7. Non-functional Attributes

7.1 Security

- Encrypt sensitive data (AES, bcrypt)
- Role-based access control to prevent unauthorised actions

7.2 Reliability

99.9% uptime during peak hours

7.3 Scalability

Ability to scale horizontally for increased demand.

8. Preliminary Schedule and Budget

8.1 Development Schedule

Phase	Duration
Requirement Analysis	1 week
System Design	2 week
Development	4 week
Testing	2 week
Total	10 weeks

9. Budget (Rough Estimation)

Item	Cost (INR)
Requirements & Design	₹ 20,000
Development	₹ 60,000
Testing	₹ 15,000
Deployment & Training	₹ 10,000
Contingency (10%)	₹ 10,500
Total	₹ 1,15,500

Lab-1

Software Requirement for Software Engg.

Software Requirement Specification for Credit Card Processing System

1 Introduction

1.1 Purpose of this Document

The purpose of this document is to provide a clear, concise, and detailed Software Requirement Specification (SRS) for the Credit Card Processing System. The document defines the functionality, features, constraints, and system requirements to ensure the secure, fast and reliable processing of card card transactions, including hotels, e-commerce platform, and retail businesses.

1.2 Scope of this document

Credit Card Processing System will:

1) Process card-present and card not present payments.
2. Support refunds and partial captures.
3. Integrate with major card networks like Visa, Master Card, Amex.
4. Provide transaction logs and reports for reconciliation.
5. Ensure compliance with PCIDSS security standards.

1.3 Overview

The Credit Card Processing System is designed to process transactions securely and efficiently.

Product will include:

(i) Merchant Dashboard: Web interface for viewing transaction details, refunds, and reports.

(ii) Transaction Processing Engine: Backend system handling authorization, clearing and settlement.

(iii) Database: Secure storage of transaction records and merchant data.

2. General Description

2.1 General functions

Product Perspective: Middleware between merchants and acquirers/schemes. Works with POS terminals, payment pages/SDKs and back office tools.

2.2 User Characteristics

Merchant Admin: Manages payment setting. view detailed reports, refunds.

Cashier/Operations: Initiates transaction and checks payment status.

2.3 Features and Benefits

Fast Authorization: Transactions proceed in 3 seconds

Secure Payment Handling: Tokenization and encryption for card details

Refund: Full or partial refunds

Detailed reporting: for reconciliation and financial analysis.

3. Functional Requirement

3.1 Transaction requirement

Requirement: System must send an authorization request to the issuing bank via payment gateway

Outcome: Merchant receives approval or decline instantly.

3.2 Payment Capture and Settlement

Requirement: System must capture authorized funds and transfer them to the issuing bank via gateway.

Outcome: Merchant receives payment within the settlement period.

3.3 Refund Processing:

Requirement: System must allow full or partial refu

Outcome: Refunded amounts are credited back to cardholder.

3.4 Fraud Detection and Prevention

Requirement: System must flag suspicious transactions

Outcome: High risk payments are reviewed or declined.

4. Interface Requirement

4.1 Software Interfaces:

Database: SQL, database for transaction and logs.

Payment Gateway API: integration with bank and card networks.

4.2 Communication Interfaces
Protocol: HTTPS for secure communication
Data format: JSON

5. Performance Requirements
5.1 System Response Time
Authorization ≤ 3 seconds
refund initiation ≤ 5 seconds
5.2 Scalability
Support at least 1,000 concurrent transactions without degradation.

6. Design Constraints
6.1 Technology Constraints
- must comply with PCIDSS standards.
- Compatible with EMV chips, magnetic stripe

6.2 Hardware Constraints
- Operate on servers with minimum 8GB RAM and 1TB storage.

7. Non-functional Attributes
7.1 Security
- AES-256 encryption for stored data
- Tokenization for PAN (Primary Account Number)

7.2 Reliability
99.5% uptime granted
7.3 Scalability
Ability to handle seasonal spikes.

8. Performing Schedule and Budget.

Phase	Task	Duration
Phase 1	Design	1 month
Phase 2	Implementation	3 month.
Phase 3	Testing and Compliance Audio	2 month
Phase 4	Deployment and Merchant Training	1 months

Budget

Estimated cost : $150,000

Includes development, PCI compliance audit, deployment and initial merchant training.

Library Management System

1. Introduction

1.1 Purpose of this document

This purpose of this document is to outline the requirements and specifications for the development of a Library Management System. It will provide a clear understanding of the project objective, scope and deliverables.

1.2 Scope of this document

This document defines the overall working and main objective of Library Management System. It includes a description of development cost and time required for the project.

1.3 Overview

The Library Management System is a software solution designed to streamline library operations, including book cataloging, user registration, book lending / return, fine calculation, and reporting.

2. General Description

The Library Management System will cater to the needs of librarians, staff and library members. It will provide features such as book search, inventory management, user accounts, reservation of books, and financial tracking (late fees, membership fees). It will be accessible to users with varying levels of technical expertise.

3. Functional Requirements

3.1 Book Management

- Add, update and delete book records in the catalog.
- Categorize books by title, author, subject and ISBN.

3.2 Member Management

- Register new members and maintain member profiles.
- Track member borrowing history and preferences.

3.3 Lending and Returns

- Allow users to borrow and return books.
- automatically update availability status of books.

3.4 Reservation Management

- allow members to reserve books currently on loan.
- notify members when reserved books are available

4. Interface Requirement

4.1 User Interface

- Intuitive and userfriendly interface for librarians, staff and members
- accessible via web browsers, mobile devices and desktop applications.

4.2 Integration Interface ~~requirement~~

- Integration with payment gateways for fine and fee transactions.
- Integration with third party digital

library platforms (eg. e-books, journal)

5. Performance Requirements
   5.1 Response Time
   The system should respond to user actions with 2 seconds.
   5.2 Scalability
   Handle a minimum of 1000 concurrent users during peak hours.
   6.3 Data Integrity
   Ensure data consistency and accuracy across all modules.

6. Design Constraints
   6.1 Hardware Limitations
   The system should be compatible with standard library hardware (computer, barcode scanners, printers, POS terminal)
   6.2 Software Dependencies etc.
   Utilize a relational database management system. (eg. MySQL) for data storage.
   Use programming languages & frameworks conducive to UML modeling.

7. Non-functional attributes
   7.1 Security
   Implement robust authentication and authorization mechanisms to protect sensitive data.
   7.2 Reliability
   Ensure high availability and fault tolerance to minimize system downtime.

7.3 Scalability
design the system to accomodate future growth and expansion.

7.4 Portability
Support multiple platforms and devices for user accessibility.

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

7.6 Reusability
System shall use modular code design to facilate future enhancement & maintains

8 Preliminary Schedule and Budget
The development of Library Management System is estimated to 6months with a budget of $80,000. This includes project planning, development, testing and deployment phase

Stock Maintenance System

1 Introduction

1.1 Purpose of this document

The purpose of this document is to outline the requirements and specifications for the development of a Stock Maintanance System(SMS). It will provide a clear understanding of the project objectives, scope and deliverables.

1.2 Scope of the Document

This document defines the overall working and main objectives of the Stock Maintainance System. It includes a description of the development cost and time required for the project.

1.3 Overview

The Stock Maintainance System is a software solution designed to manage inventory operations, including stock entry, updates, monitoring, tracking, billing & reporting.

2 Description

The Stock Maintenance System will cater to the needs of store managers, staff and administrations. It will provide features such as stock registration, supplier management, order tracking.

3. Functional Requirements

3.1 Stock Management
- Add, update and delete product records.
- provide alerts for low stock and near-expiry items.

3.2 Supplier Management
- maintain supplier profiles with contact and payment details.
- track purchase orders and supplier history.

3.3 Purchase and Sales Management
- record new stock purchases from suppliers
- track product sales and update inventory automatically.

3.4 Reporting and Analytics.
- Generate realtime reports on ~~spoo~~ stock levels, sales and purchases.
- provide trend analysis for demand forecasting.

4. Interface Requirements

4.1 User Interface

Intuitive and userfriendly interface for store staff and administrators.

Accessible via web browsers, mobiles devices, and desktop applications.

4.2 Integration Interfaces

Integration with payement gatways for sales transaction.

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 and accuracy across all modules.

6 Design Constraints

6.1 Hardware Limitations

The system should be compatible with standard store hardware (computers, barcode scanners, printers).

6.2 Software Dependencies

Utilize a relational database management system (e.g MySQL) for data storage.

Use programming languages and frameworks conducive to UML modeling.

7 Non-functional Attributes

7.1 Security

Implement authentication and authorization mechanisms to secure transactions and stock data

7.2 Reliability

Ensure fault tolerance and data backup to minimize downtime.

7.3 Scalability
Design the system to handle growth in product range and user base.

7.4 Portability
Support multiple platforms and devices for user accessibility.

7.5 Usability
Provide a clean, easy to use interface with minimal training required.

7.6 Reusability
follow modular design principles to support future enhancements.

7.7 Data Integrity
Ensure accurate, consistent storage and retrieval of stock data.

8 Preliminary Schedule and Budget
The development of the Stock Maintenance System is estimated to take 5 months with a buget of $70,000. This includes project planning, development, testing, and deployment phases.

PASSPORT AUTOMATION SYSTEM

1 Introduction

1.1 Purpose of this Document

The purpose of this document is to outline the requirements and specifications for the development of a Passport Automation System (PAS). It will provide a clear understanding of the project objectives, scope, and deliverables.

1.2 Scope of this Document

This document defines the overall working and main objectives of the Passport Automation System. It includes a description of the development cost and time required for the project.

1.3 Overview

The Passport Automation System is a software solution designed to automate passport application, verification, approval, and delivery processes. It enables applicants to apply online, track status, schedule appointments.

2 General Description

The Passport Automation System will cater to the needs of passport applicants, passport office staff, and government authorities. Such as online application submission, document management.

3 Functional Requirements

3.1 Application Management
(1) Allow users to submit new passport applications online.
(2) Enable applicants to upload required documents (ID proof, address proof, etc).
(3) Assign unique application IDs for tracking.

3.2 Appointment Scheduling
(1) Allow applicants to book / reschedule appointments at passport centres.
(2) Generate appointment confirmations and reminders.

3.3 Verification and Approval
(1) Provide tools for staff to verify documents and application details.
(2) Route applications through various approval stages.

3.4 Payment and Billing
(1) Calculate application fees (new issue, renewal)
(2) Support online payments and generate digital receipts.

3.5 Status Tracking and Notifications
(1) Allow applicants to track application status in real-time.
(2) Send automated email / SMS notifications regarding updates.

4. Interface Requirements

4.1 User Interface

(1) Intuitive and user-friendly interface for applicants and officials.

(2) Accessible via web browsers and mobile applications.

4.2 Integration Interface

(1) Integration with payment gateways for secure online fee transactions.

(2) Integration with government identity database.

5 Performance Requirement

5.1 Response Time

The system should respond to user actions within 3 seconds.

5.2 Scalability

Handle a minimum of 5000 concurrent users during peak hours.

5.3 Data Integrity

Ensure consistency and accuracy of application records across all modules.

6. Design Constraints

6.1 Hardware Limitations

The system should be compatible with standard government office hardware (computer, scanners, printers).

6.2 Software Dependencies
(1) Utilize a relational database management system (eg Oracle, MySQL).

7 Non functional Attributes
7.1 Security
Implement robust authentication, encryption and role based access control.

7.2 Reliability
ensure high availability and backup mechanisms to avoid data loss.

7.3 Scalability
Design the system to handle increasing number of application over time.

7.4 Portability
Support web-based and mobile platforms for easy accessibility.

7.5 Usability
Provide a simple, step by step application process for users with minimal technical knowledge.

7.6 Reusability
Use modular code design for easy integration of future government services.

7.7 Compatibility

Ensure compatibility with common web browser (Chrome, firefox, edge)

7.8 Data Integrity

Guarantee secure and accurate handling of personal information

8. Preliminary Schedule & Budget

The development of the Passport Automation System is estimated to take 8 months with a budget of $ 150,000; includes project planning, development, integration with government system, testing, and deployment phases.