

22-08-25

Software Requirements Specification:

Hotel Management 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 Hotel Management System. 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 Hotel Management System. It includes a description of the development cost and time required for the project.

1.3 Overview : The Hotel Management system is a software solution designed to streamline hotel operations, including reservation management, guest check-in / check-out, room assignment, billing, and reporting.

2. General description : The Hotel Management system will cater to the needs of hotel staff and management, providing features such as room booking, guest profiles, inventory management and 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 front desk.
- Generate reservation confirmations and send notifications to guests.

3.2 Room Management:

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

3.3. Guest Management:

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

3.4. Billing and Invoicing:

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

4. Interface requirements:

4.1. User Interface:

- Intuitive and user-friendly interface for hotel staff and guests.
- Accessible via web browsers, mobile devices, and 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 and 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 and frameworks conducive to UML modeling (e.g. Java, Spring Boot).

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 accommodate 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. Flexibility:

- The system shall use modular design to facilitate future enhancements and 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 and consistent data storage and retrieval.

8. Preliminary Schedule and Budget:

The development of the Hotel Management System is estimated to take 6 months with a budget of \$100,000. This includes project planning, development, testing, and deployment phases.

22-03-25

Software Requirement Specifications:

Credit Card Processing System

1. Introduction:

H. Purpose of this document:

- The purpose of this document is to outline the requirements and specifications for the development of a Credit Card Processing system. 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 Credit Card Processing system. It includes a description of the development cost and time required for the project.

1.3. Overview:

- The Credit Card Processing System is a secure software solution designed to manage credit card transactions, authorization, settlement, fraud detection, billing and reporting. It will provide seamless integration with merchant systems, banks, and payment networks to ensure fast, reliable, and secure payment processing.

2. General Description:

- The Credit Card Processing system will fully meet the needs of merchants, financial institutions, and customers, providing features such as transaction authorization, fraud monitoring, settlement, charge back handling, and detailed financial reporting. It will be accessible to users with varying levels of technical expertise.

3. Functional Requirements:

3.1. Transaction Authorization

- Validate and authorize transactions in real-time.
- Support different transaction types (purchases, refunds, reversals).

3.2. Fraud Detection and Prevention

- Implement fraud detection algorithm to flag suspicious transactions.
- Provide alerts and real-time monitoring for unusual patterns.

3.3. Settlement and Clearing:

- Process settlement with banks and payment networks.
- Ensure accurate reconciliation of funds between merchants and banks.

3.4. Billing and reporting:

- Generate billing statements for merchants and cardholders.
- Provide detailed financial reports (daily, weekly, monthly).

3.5. Dispute and Chargeback Management:

- Facilitate the handling of chargebacks and disputes.
- Maintain a history of chargeback cases for reference and auditing.

4.1 Interface requirements:

4.1.1 User Interface:

- Intuitive and user-friendly interface for merchants and administrators.
- Accessible via web portals, APIs, and mobile applications.

4.2. Integration Interfaces:

- Integration with payment gateways and banking systems for seamless transaction flow.
- API support for merchant integration and third-party application.

5. Performance Requirements:

5.1. Response Time:

- The system should authorize transactions within 1 second.

5.2. Scalability:

- Handle a minimum of 10,000 concurrent transactions during peak hours.

5.3. Data Integrity:

- Ensure transaction accuracy and consistency across all modules.

6. Design Constraints:

6.1. Hardware Limitations:

- The system should be compatible with standard merchant hardware (POS terminals, card readers, servers).

6.2. Software Dependencies:

- Utilize a relational Database management system (e.g. Oracle, PostgreSQL).

7. Non-Functional Attributes:

7.1. Security:

- Implement PCI DSS compliance for secure handling of cardholder data.

7.2. Reliability:

- Ensure high availability and redundancy to minimize system downtime.

7.3. Scalability:

- Support future growth in transaction volumes and merchant base.

7.4. Portability:

- Support deployment on multiple platforms and cloud environments.

7.5. Usability:

- Provide a simple, clear interface for merchants with minimal training requirements.

7.6. Reusability:

- Use modular code design to facilitate enhancements and upgrades.

7.7. Compatibility:

- The system shall be compatible with major payment networks (Visa, Mastercard, AMEX, etc).

7.8. Data Integrity:

- Ensure accurate, consistent and tamper-proof transaction records.

8. Preliminary Schedule and Budget:

The development of the Credit Card Processing system is estimated to take 9 months with a budget of ₦ 250,000. This includes project planning, development, testing, compliance certification, and deployment phases.

22-08-25

Software Requirement Specifications:

Library Management 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 Library Management System. It aims to provide a clear understanding of the system's objectives, scope, and deliverables for developers, stakeholders and users.

1.2 Scope of this Document:

- The library management system will manage book check-in/check-out, member profiles, fee calculations and inventory. It will support both staff and library users with web and mobile access.

1.3 Overview:

- The system aims to streamline library operations and provide an efficient, user-friendly platform for managing resources and user interactions.

2. General Description:

- The system will support book management, member tracking, circulation, and reporting. It will be accessible to ~~all~~ users of varying technical levels and integrate with library hardware like barcode scanners.

3. Functional Requirements:

- 3.1. Book - Management : Add / edit / delete books, categorize , and search.
- 3.2. Member management : Register and manage users, track borrowing history.
- 3.3. Circulation : Issue / return books, track due dates , send overdue alerts.
- 3.4. Fines : Auto - calculate overdue / fines and manage payments.
- 3.5. Reports : Generate reports on inventory , usage and fines.

4. Interface Requirements:

- 4.1. User Interface: Simple UI for staff and members accessible on web and mobile.
- 4.2. Integration: Support barcode scanners and notifications via email/SMS.

5. Performance Requirements:

- 5.1. Response time under 2 seconds.
- 5.2. Support for up to 500 concurrent users.
- 5.3. Maintain accurate and consistent data.

6. Design Constraints:

- 6.1. Compatible with standard library hardware.
- 6.2. Use relational databases (MySQL/PostgreSQL).
- 6.3. Built with popular frameworks (e.g. Django or Spring Boot).

7. Non-Functional requirements:

- 7.1. Security : Role-based access, data encryption.
- 7.2. Reliability : 99.9% uptime, regular backups.
- 7.3. Scalability and Portability : Multi-Device support, future proof design.
- 7.4. Usability : Clear navigation and help features.
- 7.5. Compatibility : Works with major browsers.
- 7.6. Data Integrity : Prevent duplicate or corrupted records.

8. Schedule and Budget:

Estimated development time : 5 months.

Estimated budget : ₹ 75000 (includes planning, development, testing and deployment).

22-08-25

Software Requirement Specification:

Stock Maintenance System

1. Introduction:

1.1. Purpose of this document:

- This document outlines the requirements for a Stock maintenance system (SMS), which aims to manage inventory, track stock levels, and streamline stock-related operations for businesses.

1.2. Scope:

- The system will handle stock entries, update item tracking, supplies details, alerts for low stock and generate reports. It will support both desktop and web-based interface for warehouse or store staff.

1.3. Overview:

- The SMS is designed to monitor inventory efficiently, reduce manual errors, and ensure accurate stock records through real-time updates and reporting.

2. General Description:

- This system will support features such as adding new items, tracking stock levels, managing supplier, issuing stock, receiving goods, and generating inventory reports.

3. Functional Requirements:

3.1. Item Management : Add, update, delete item records ; categorize items.

3.2. Stock Tracking : Real-time tracking of available, issued and received stock.

3.3. Supplies Management : Store supplier info and manage purchase orders.

3.4. Notifications : Alert for low stock or stock expiry.

3.5. Reports : Generate stock-level, movement, ~~and~~ reorder reports.

4. Interface Requirements:

4.1. User Interface : Simple UI for warehouse / store staff ; accessible via desktop and web.

4.2 Integration: Barcode scanner and supplier system integration for automated updates.

5. Performance Requirements:

- Respond to user actions within 2 seconds.
- Support up to 200 concurrent users.
- Maintain data accuracy across modules.

6. Design Constraints:

- Compatible with standard PCs and barcode scanners.
- Use ~~relational~~ relational databases (e.g. MySQL).
- Built with frameworks like Spring Boot.

7. Non-functional Requirements:

7.1. Security: User authentication and access control.

7.2. Reliability: Auto-backups and fault tolerance.

7.3. Availability: Handles expanding product and supplier databases.

7.4. Portability : run on multiple OS platforms.

7.5. Data Integrity : Ensure accurate and consistent inventory records.

8. Schedule and Budget:

Estimated development time : 4-5 months

Estimated budget : \$ 60,000 (includes design, implementation, testing, and deployment).

22-08-25

Software Requirement Specifications:

Passport Automation System

1. Introduction:

1.1. Purpose:

- This document defines the requirements for a Passport Automation System (PAS), aimed at digitizing and streamlining passport application, processing, and issuance.

1.2. Scope:

- This system will allow users to apply for passports online, schedule appointments, track application status, and receive updates. It will also assist passport office staff with application processing, certification, and document management.

1.3. Overview:

- PAS is designed to reduce manual processing, improve efficiency, and offer transparency in passport services through a secure and user-friendly platform.

2. General Description:

- The system will serve applicants and passport office staff. It will support functionalities like new application, status tracking, document verification.

3. Functional Requirements:

3.1. Application Management: Submit new applications, edit details.

3.2. Appointment Scheduling: Book / reschedule appointments for biometric.

3.3. Status tracking: Real-time updates on application progress.

3.4. Payment Gateway: Secure payment for application processing fees.

4. Interface requirements:

4.1. User interface: Simple, multilingual interface for applicants and staff.

4.2. Integration: Payment gateway integration.

5. Performance requirements:

- Respond to user actions within 2 seconds.
- Support up to 1000 concurrent users.

6. Design constraints:

- Compatible with government systems and standard hardware.
- Use of proven databases.

7. Non-Functional Requirements:

- 7.1. Security: End-to-End encryption, user authentication, role-based access.
- 7.2. Scalability: Capable of handling increased application load during peak times.
- 7.3. Portability: Accessible across platforms and browsers.
- 7.4. Usability: Guided workflows for easy applications and review process.

7.5. Compatibility: Support for Chrome, Firefox, Safari and edge.

8. Schedule and Budget:

Estimated development time: 6 months

Estimated budget: \$ 90,000 (includes planning, design, development, testing and deployment).