SRS: Hotel Management System

Problem statement:
Design and implement a Hotel Management system that automates room booking, check-in / check-out, billing, staff and customer management and inventory control to improve efficiency and customer satisfaction. It will a web / desktop application with database and friendly frontend, ensuring fast response (<3s), scalability, security, reliability, portability and maintainability.

SRS:

1. Introduction:

1.1 Purpose of this document:
The purpose of this document is to provide a clear, concise and detailed software requirements specification (SRS) for the Hotel Management system (HMS). This document defines the system's functionality, features, constraints and requirement to ensure the successful development and implimentation of the HMS.

1.2 Scope of this document:
This document covers all essential aspects related to the development of the HMS including the functional and non-functional requirements, user interfaces, design constraints and other system attributes. The HMS will be designed to automate core hotel operations such as room booking, check-in / check-out, customer, record management, billing and payments, staff allocation and inventory control.

1.3 Overview
The Hotel Management System (HMS) is intended to streamline and automate the overall management of hotel

operations. It allows customers to search and book rooms, perform check-in and check-out and make payment, while hotel staff and administrators can manage bookings, customer records, staff schedules, billing and inventory.

2. General Description:

The Hotel Management system will cater to the needs of hotel staff and management, providing features such a 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 the front desk.

3.2 Room Management:

- Assign rooms to guests based on availability and preferences.

3.3 Guest Management:

- Maintain guest profiles with personal information, preferences and booking history.

3.4: Billing and Invoicing:

- Generate accurate bills for room charges, additional services, and taxes.

4. Interface Requirements:

4.1 User Interface:

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

4.2 Integration Interface:

- Integration with payment gateways for secure transactions.

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.

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.

8. Preliminary Schedule and Budget: The development of the Hotel Management system is estimated to take 6 months with budget of $ 100000. This includes project planning, development, testing and deployment phases.

SRS: Credit Card Processing

Problem statement:

Design and implement a Credit Card Processing that automates authorization, settlement, billing, fraud detection and reporting. It will be a web-based, backend-integrated application with secure APIs and databases for banks and merchants. The system must ensure real-time processing (<2s), high availability (99.99%), scalability and PCI DSS compliant security, developed using an incremental/agile method.

SRS:

1. Introduction:

1.1 Purpose of this Document:

The purpose of this document is to outline the requirements and specifications for the development of Credit Card Processing. It provides a clear understanding of the system's objectives, scope, and deliverables to ensure successful implementation.

1.2 Scope of this Document:

This document defines the overall working and main objectives of Credit Card Processing. It includes details on the system's cost, time of development and constraints. The CCP will automate the handling of authorization, authentication, transaction settlement, billing, fraud detection and reporting. ensuring security, spud and compliance with financial standards.

1.3 Overview:

The Credit Card Processing is a software solution designed to securely process and manage credit card transactions. It will handle authorization requests from merchants, authenticate cardholders, perform fraud detection, manage settlements

between banks, generate billing statements and produce reports.

2. General Description:

The Credit Card Processing will serve banks, merchants, and cardholders by offering real time, secure and efficient transaction processing. It will include features such as transaction authorization, fraud detection, billing and statement generation, settlement management, and reporting tools.

3. Functional Requirements

3.1 Transaction Authorization:

- Validate card details and available balance.
- Approve or reject transactions in real time.

3.2 Fraud Detection:

- Handle settlement between issuing banks and merchant banks

3.3 Settlement Management:

- Monitor transaction patterns of unusual activity.

3.4 Biling and Invoicing:

- Generate accurate monthly statements for cardholders.

3.5 Reporting:

- Generate transaction logs, audit reports and compliance summaries.

4. Interface Requirements:

4.1 User Interface

- Intuitive and user friendly portal for cardholders, merchants and administrators.
- Accessible via web browsers, mobile app and desktop applications.

4.2 Integration Interfaces:
- Integration with payment gateways and banking APIs for secure transactions.
- Support for third party fraud detection tools.

5. Performance Requirements:
5.1 Response Time
- Transactions should be processed within 2 seconds.
5.2 Scalability
- Support at least 10000 concurrent transactions per second.
5.3 Data Integrity
- Ensure consistency and accuracy of transaction data across all modules.

6. Design Constraints
6.1 Hardware Limitations:
- Compatible with standard banking infrastructure and cloud servers.
6.2 Software Dependencies:
- Use relational / NoSQL databases (MySQL, PostgreSQL, MongoDB).
- Backend in Java / Python, secure API communication with TLS/SSL.

7. Non-functional attributes:
7.1 Security:
- PCI DSS compliance, AES/RSA encryption and tokenization.
7.2 Reliability:
- Ensure 99.99% uptime with fault-tolerant architecture
7.3 Scalability:
- Designed to support global transaction volumes.
7.4 Portability:
- Accessible across multiple platforms and devices.

1.5 Usability :

- Provide simple navigation for merchants and cardhol

1.6 Reusability :

- Use modular components for easy maintenance

8. Preliminary Schedule and Budget :

The development of the Credit Card Processing is essential to take 8 months with a budget of $ 300000. This includ planning, development, security compliance, testing and deployment phases.

SRS : Library management System

Problem statement :

- Design and implement a Library Management System (LMS) that automates book cataloging, member registration, borrowing / returning, fine calculation and reporting. It will be a web / desktop application with a relational database and user friendly interface for admins, librarians and members. The system must ensure fast search (< 2s) scalability, security and reliability, developed using an incremental model with phased module integration.

SRS :

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 (LMS). It will provide a clear understanding of the system's objectives, scope, and deliverables to ensure successful development and implementation.

1.2 Scope of this document :

This document defines the overall working and main objectives of the Library Management System. It covers functional and non-functional requirements, user interface design constraints and system attributes. The LMS will automate the management of books, journals and members, enabling efficient handling of borrowing, returning, overdue fines, cataloging and reporting.

1.3 Overview :

The Library Management System is a software solution designed to simplify and automate library operations. It will enable members to browse, borrow, and return books, while librarians

and administrators can manage book inventory, manage member records, track due dates, calculate fines and generate reports.

2. General Description:

The Library Management System will serve librarians, administrators and library members by providing features such as book management, member management, borrowing and returning process, fine calculation and reporting. It will be accessible to users with varying technical expertise through a user-friendly interface.

3. Functional Requirements:

3.1 Book Management:

- Add, update or delete book records
- Track book availability (available, issued, overdue)

3.2 Member Management:

- Register and update member details
- Assign unique IDs for tracking member activities.

3.3 Borrowing and Returning:

- Record borrowing and returning of books.
- Update due dates and calculate overdue fines automatically.

3.4 Fine Management:

- Automatically calculate fines for overdue books
- Notify members of outstanding dues.

4. Interface Requirements:

4.1 User Interface:

- Simple, intuitive web/desktop interface for librarians, admins and members.
- Accessible through browsers and desktop applications.

4.2 Integration Interfaces:
- Integration with database for storing book, member and transaction data.
- Support for online payment gateway (optional) for fine collection

5. Performance Requirements:
5.1 Response Time:
- System should respond to user queries (eg book search) within 2 seconds.

5.2 Scalability:
- Support at least 1000 concurrent users.

5.3 Data integrity:
- Ensure accuracy and consistency of book and member records across all modules.

6. Design Constraints:
6.1 Hardware Limitations:
- Compatible with standard desktops and servers.

6.2 Software Dependencies:
- Use relational database management system (eg: MySQL)
- Backend in Java / Python; fronted in HTML5, CSS, JavaScript.

7. Non-Functional Attributes:
7.1 Security:
- Secure login with encryption and role-based access.

7.2 Reliability: System uptime ≥ 99.9%
7.3 Scalability: Support expansion of resources and modules.
7.4 Portability: Accessible across multiple platforms
7.5 Usability: User-friendly navigation for librarians and members

8. Preliminary Schedule and Budget:
The development of LMS is estimated to take 6 months with budget of $80000 including project planning, development and testing.

SRS: Stock Maintaining System

Problem statement:

Design and implement a Stock Maintaining System (SMS) that automates inventory operations such as stock entry, update, sales/purchase tracking, reorder alerts and reporting. It will be web/desktop application with a relational database backend and user-friendly interface for admins, staff and managers. The system must ensure fast updates (<2s), scalability, accuracy, security and reliability, developed using an incremental model with phased module integration.

SRS

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 maintaining system (SMS). It will provide a clear understanding of the project objectives, scope and deliverables to ensure successful development and implementation.

1.2 Scope of this document:

This document defines the overall working and objectives of Stock Maintaining System. The SMS will automate stock entry, updates, sales and purchase tracking, inventory control, reorder alerts and reporting. It will help businesses maintain accurate, real-time inventory records.

1.3 Overview:

The Stock Maintaining System is software solution designed to streamline inventory management. It will allow staff to add and update stock records, track sales and purchases, generate low-stock alerts and provide reports for analysis.

2. General Description:

The Stock Maintaining System will serve administrators, staff and managers by providing feature such as inventory management transaction recording, alert generation and reporting. It will be accessible to users with varying technical expertise through a simple and user-friendly interface.

3. Functional Requirements:

3.1 Stock Management.

- Add, update and delete stock items.
- Track available quantities in real-time

3.2 Transaction Management:

- Record sales and purchases.
- Automatically update stock levels.

3.3 Reorder Alerts:

- Notify when stock levels fall below defined thresholds
- Provide suggestions for reorder quantities

3.4 Reporting:

- Generating reports on stock levels, sales, purchases and reorder history.

4. Interface Requirements:

4.1 User Interface:

- A simple intuitive interface for staff and managers.
- Accessible via web browsers and desktop applications.

4.2 Integration Interfaces

- Integration with relational database for storing stock and transaction records.
- Optional integration with barcode scanners or POS systems.

5. Performance Requirements

5.1 Response Time: stock updates and queries must reflect within 2 sec.

5.2 Scalability support up to 200 concurrent users.
5.3 Data Integrity: Ensure accuracy and consistency across all inventory records.

6. Design Constraints
6.1 Hardware Limitations:
• Compatible with standard office computers and servers.
6.2 Software Dependencies
• Use relational database (MySQL)
• Backend in Java/Python, frontend in HTML 5 / CSS / Javascript

7. Non-Functional Attributes:
7.1 Security: Role based authentication and secure data handling
7.2 Reliability: System uptime ≥ 99.9%.
7.3 Scalability: Must support expansion in stock size and user base
7.4 Portability: Compatible with web and dd
7.5 Usability: Easy navigation for staff and managers.

8. Preliminary Schedule and Budget:
The development of the Stock Maintaining System is estimated to take 5 months with a budget of $50000. This includes project planning, development, testing and deploy

SRS : Passport Automation System

Problem statement :

Design and implement a Passport Automation System (PAS) that automates application submission, document verification, appointment scheduling, status tracking and issuance. It will be a web/desktop application with secure database and user friendly interfaces for applicants, officials and administrators. The system must ensure fast processing (<2s), high security, scalability and 99% reliability, development using an incremental model.

SRS

1. Introduction :

1.1 Purpose of this document:

The purpose of this document is to outline the requirements and specifications for the development of Passport Automation System. It provides a clear understanding of objectives, scope and deliverables of system to ensure its successful development and implementation.

1.2 Scope of this document:

This document defines the overall working and objectives of Passport Automation System. The PAS will automate the entire process of passport application, document submission, verification, appointment scheduling, status updates and passport issuance.

1.3 Overview

The Passport Automation System is designed to streamline passport services by enabling application status and secure notification. Officials will be able to verify documents, process application status, and manage records, while administrators can oversee operations, generate reports.

2. General Description:

The passport Automation system will serve applicants, officials and administrators offering features such as application submission, document verification, appointment booking, fee payment, status tracking and reporting

3 Functional Requirements:

3.1 Application Management:

- Allow applicants to fill and submit passport application
- Enable secure uploading of required documents online.

3.2 Document Verification:

- Officials verify uploaded documents against application data
- Provide status updates

3.3 Appointment Scheduling:

- Applicants can select and book appointment slots.
- Send automated reminders and confirmation notifications

3.4 Payment Processing:

- Support secure online payments for application fees.
- Generate receipts and maintain transaction records

3.5 Status Tracking:

- Allow applicants to check real-time status of applications.
- Notify users via email / SMS on updates.

4. Interface Requirements:

4.1 User Interface:

- Intuitive and accessible interface for applicants, officials.
- Support for web browsers and mobile devices.

4.2 Integration interfaces:

- Integration with government identity databases for verification
- Integration with payment gateways for secure transaction

5. Performance Requirements:

5.1 Respons Time: System must respond to user actions within 2 sec.

5.2 Scalability: Handle at least 5000 concurrent users during peak time.

5.3 Data Integrity: Ensure accuracy and consistency of applicant records and transactions.

6. Design Constraints:

6.1 Hardware Limitations:

- Compatible with government IT infrastructure and servers.

6.2 Software Dependencies:

- Use relational database (eg: MySQL)
- Backend in Java/Python, frontend in HTML5/JS framework

7. Non-Functional Attributes:

7.1 Security: End to End encryption, secure login and role based access control.

7.2 Reliability: Maintain 99.9% uptime with backup & recovery

7.3 Scalability: Designed to expand with increasing applicants and data.

7.4 Portability: Support cross-platform access

7.5 Usability: simple, user-friendly forms and navigation.

7.6 Reusability: Modular architecture for future enhancement

8. Preliminary Schedule and Budget:

The development of the Passport Automation System is estimated to take 7 months with a budget of $ 200000. This includes project planning, development, testing, security compliance and deployment.

28/8/25