Software Requirement Specification (SRS)

Hotel Management System

1. Introduction

1.1 Purpose

The aim of this document is to explain the requirements of the Hotel Management System. The system should help hotel staff and guests to manage bookings, check-ins, check-outs, billing and other basic hotel activities. This document will be useful for developers, project manager and also for testing purpose

1.2 Scope

The Hotel Management System is a software which will keep track of hotel rooms, reservations, customers and payments. It will reduce manual work at reception and make the process faster. The system can be used in small as well as medium size hotels.

1.3 Overview

The system will have features like reservation, room allocation, guest details, billing and report generation. It will be user friendly and easy to use for both staff and guests.

DATE: PAGE:

2. General Description

The Hotel Management System will mainly help receptionists to do tasks like booking rooms, checking room availability, generating bills and storing guest details. Managers can also use the system to view reports of occupancy and revenue. Guests can book rooms by contacting the reception or through online booking.

3. Functional Requirements

3.1 Reservation Module

* User can reserve rooms in advance
* Confirmation should be given for successful booking
* Cancellations and changes should be possible

3.2 Room Module

* Show status of room (empty, occupied, under cleaning)
* Assign available rooms to guests.

3.3 Guest Module

* Store guest information like name, contact, ID proof
* Keep record of guest's previous visits.
* Support smooth check-in and check-out.

3.4 Billing Module

* Generate bills for room charges and services
* Support cash or, UPI or Card payments.
* Print receipts.

DATE: PAGE:

3.5 Report Module

* Daily and monthly reports on revenue.
* Occupancy details of rooms.

4. Interface Requirements

4.1 User Interface
* Simple and easy to use screens.
* Desktop based interface.

4.2 External Interface
* Printer for printing bills.
* Payment gateway integration.

5. Performance Requirements

* The system should respond within 2-3 seconds.
* It should be able to handle atleast 500 users simultaneously.

6. Design Constraints
* Database should be relational
* It should run on normal hotel computers.

7. Non-functional Requirements

* Security - Only authorized staff should login
* Reliability - System should not crash frequently.
* Usability - Easy for staff with basic computer knowledge.
* Scalability - Can be expanded for more branches.

8. Schedule and Cost

The project can be completed in around 5-6 months. Estimated cost is approximately $90,000 to $120,000. This includes phases such as requirement analysis, system design, module development, integration, testing, deployment, and staff training.

DATE: PAGE:

Credit Card Processing System

Introduction

1.1 Purpose of this Document

The purpose of this document is to describe the requirements and specifications of a Credit Card Processing System. This document explains the objectives, scope, and deliverables in a simple way so that developers and users can understand the overall working of the system.

1.2 Scope of the Document

This document defines the working and goals of the Credit card Processing System. It includes features like transaction authorisation, fraud detection billing and report generation. It will also mention the cost and time required for development.

1.3. Overview

The Credit Card Processing System is software designed to handle credit card payments securely and efficiently. It will allow merchants to process customer payments, verify transactions, manage accounts, and generate financial records.

DATE: PAGE:

2. General Description

The system will be useful for banks, merchants and customers. It will include functions like transaction authorization, payment settlement, fraud detection, and billing. The system is designed to be secure, user-friendly, and accessible for both technical and non-technical users.

3. Functional Requirements

3.1 Transaction Processing

* Authorize transactions instantly
* Validate card details and account balance.
* Handle refunds and cancellations.

3.2 Fraud Detection

* Detect suspicious transactions using rules.
* Alert administrators about possible fraud.

3.3 Account Management

* Maintain customer credit card profiles.
* Manage credit limits and outstanding balances.

3.4 Billing and Reporting

* Generate monthly bills for customers.
* Provide transaction history reports for merchants and banks
* Accept multiple payment modes for clearing bills.

4. Interface Requirements

4.1 User Interface

* Simple and clear interface for customers, merchants and bank staff.
* Accessible via web browsers.

4.2 Integration Interfaces

* Integration with payment gateways.
* Support for third-party merchant.

5. Performance Requirements

* The system should process transactions within 3 seconds.
* Support atleast 5000 concurrent transactions during peak hours.
* Ensure accurate recording of every transaction without loss.

6. Design Constraints

* Should run on standard bank servers and POS terminals.
* Should use relational database.
* Prefer secure programming languages like Java.

7. Non-functional Requirements

* Security: Use encryption and authentication to protect card details.

DATE: PAGE:

* Reliability - Ensure high availability to minimize failed transactions.
* Scalability - Design for future expansion with more users and merchants.
* Data Integrity - Store and retrieve transaction data accurately and consistently.

8. Preliminary Budget and Schedule.

The development of Credit Card Processing System is estimated to take 8 months with a budget of atleast $150,000. This includes planning, development security testing and deployment.

DATE: PAGE:

Library Management System

1. Introduction

1.1 Purpose of this Document

The purpose of this document is to explain the requirements of the Library Management System. It will describe the features, scope, and objectives in a simple manner so that the working of the project is easily understood.

1.2 Scope of this Document

This document covers the working of a Library Management System that will automate the process of issuing, returning and managing books. It will also handle member records, fines and book inventory. Time and cost of development are also considered.

1.3 Overview

The Library Management System is a software designed to simplify library operations. It will help librarians manage books, keep track of issued/returned books, handle fines, and generate useful reports. Members will also be able to search and reserve books.

2. General Description

The system will be useful for librarians, students and faculty members. It will automate manual tasks like book issue, return and record keeping. It

DATE: PAGE:

will also provide easy search options and maintain data accurately.

3. Functional Requirements

3.1 Book Management.

* Add, update or delete books from library database.
* Track availability of books in real-time.

3.2 Member Management.

* Maintain member records (students and faculty).
* Issue library cards and manage memberships.

3.3 Issue and Return

* Allow issuing of books to members.
* Track due dates and calculate fines for late returns.

3.4 Search and Reservation

* Provide search by title, author or subject.
* Allow members to reserve ~~ont~~ book online if unavailable.

3.5 Reports

* Generate reports on issued books, fines, and available stock.

DATE: PAGE:

4. Interface Requirements

4.1 User Interface

* Easy-to-use interface for members and librarians
* Access through web browsers or desktop applications.

4.2 Integration Interfaces -

* Barcode scanner integration for quick book entry.
* Integration with student/faculty ID systems.

5. Performance Requirements

* System should respond to search queries within 2 seconds.
* Should handle atleast 500 users at a time.
* Ensure accurate and consistent records for books and members.

6. Design Constraints

* Compatible with standard PCs, barcode scanners and printers.
* Use of relational database
* Development in easily-maintainable programming language.

7. Non-Functional Requirements-

* Security- Protect member details and transaction history.
* Reliability- Ensure system is available during library working hours.

DATE: PAGE

* Reliability - Ensure high availability to minimize failed transactions.
* Scalability - Design for future expansion with more users and merchants.
* Data Integrity - Store and retrieve transaction data accurately and consistently.

8. Preliminary Budget and Schedule.

The development of Credit Card Processing System is estimated to take 8 months with a budget of atleast $150,000. This includes planning, development, security testing and deployment.

DATE: PAGE:

Stock Maintenance System

1. Introduction

1.1 Purpose of this Document-

The purpose of this document is to describe the requirements and specifications of the Stock Management System. It will outline the goals, features, and deliverables so that the developers and users can easily understand the project.

1.2 Scope of this Document

This document defines the working of a stock Management System that will track, manage and update stock levels in warehouses or shops. It will help reduce manual errors, monitor inventory levels and generate reports. Cost and development time are also mentioned.

1.3 Overview

The Stock Management System is designed to manage inventory in real-time. It will allow staff to add new stock, update quantities, check stock levels, and generate alets for low stock. Reports for sales and purchases will also be available

2. General Description

The system will be useful for shopkeepers, warehouse managers, and business owners. It will replace manual stock registers with an automated database, improving accuracy and saving time. The

DATE: PAGE:

system will support multiple users with different access levels.

3. Functional Requirements

3.1 Stock Entry and Update

* Add new items with details (name, category, supplier, cost, quantity).
* Update stock quantities after sales or purchases.

3.2 Stock Monitoring

* Show current stock levels
* Generate alerts when stock goes below the minimum threshold.

3.3 Sales and Purchase Records

* Record sales transactions and update stock automatically.
* Maintain purchase history from suppliers.

3.4 Reports

* Generate daily/weekly/monthly stock reports.
* Provide sales v/s purchase analysis.

3.5 User Management

* Different access rights for admin, staff and managers.

DATE: PAGE:

4. Interface Requirements

4.1 User Interface

* Easy-to-use dashboard for stock overview.
* Support for both desktop and mobile browsers.

4.2 Integration Interfaces

* Integration with barcode scanners for fast entry.
* Possible integration with POS (Point of Sale) systems.

5. Performance Requirements

* System should update stock records within 2 seconds.
* Handle atleast 2000 stock items and 500 users at a time.
* Ensure accurate updates after every sales/purchase entry.

6. Design Constraints

* Should run on standard PCs, barcode scanners and printers.
* Database should be relational
* Prefer languages such as Java/PHP.

7. Non-functional Requirements

* Security: User authentication to avoid unauthorized access.
* Reliability: System should not lose data even if power failure occurs.
* Scalability: Capable of expanding to multiple warehouses/branches.

DATE: PAGE:

* Data Integrity - Maintain consistency of stock data across all modules.

8. Preliminary Schedule and Budget

The development of the stock Management System is estimated to take 6 months with a budget of $90,000 - $110,000. This includes planning, development, testing and deployment.

DATE: PAGE:

Passport Automation System

1. Introduction

1.1 Purpose of this Document

The purpose of this document is to describe the requirements and specifications of the Passport Automation System. It will explain the objectives, scope, and features so that developers and users can understand the system easily.

1.2 Scope of this Document

This document covers the working of a Passport Automation System that will help citizens apply for, track, and receive passports efficiently. It will also help passport office staff manage applications, verify documents, and schedule appointments. Development cost and time are also mentioned.

1.3 Overview

The Passport Automation System is software designed to simplify the passport application process. It will allow online application submission, document verification, appointment scheduling, and tracking of passport status. The system aims to reduce manual paperwork and waiting time.

2. General Description

The system will be useful of citizens applying for passports and staff managing applications. It will replace manual processes with a digital workflow, ensuring accurate record keeping and faster processing.

DATE: PAGE:

Users with varying technical skills will be able to use the system.

3. Functional Requirements

3.1 Application Submission

* Citizens can fill and submit passport application forms online.
* Upload required documents (ID proof, address proof, photographs).

3.2 Appointment Scheduling

* Schedule appointments at passport offices for verification.
* Reschedule or cancel appointments if needed.

3.3 Verification and Approval

* Staff can verify documents and application details.
* Approve or reject applications based on verification.

3.4 Passport Issuance and Tracking

* Generate passports for approved applications.
* Provide online tracking for citizens to check application status.

3.5 Notifications

* Send email / SMS alerts for appointment reminders and status updates.

DATE: PAGE:

4. Interface Requirements

4.1 User Interface

* Simple and easy-to-use web interface for citizens.
* Staff interface for managing applications and verification.

4.2 Integration Interfaces

* Integration with government ID databases for document verification.
* SMS and email services for notifications.

5. Performance Requirements

* The system should respond to user actions within 3 seconds.
* Should handle atleast 5000 concurrent users during peak application periods.
* Ensure accurate storage and retrieval of citizen application data.

6. Design Constraints

* Should run on standard PCs at passport offices and support online access.
* Database should be relational
* Development should be done in programming languages such as Java and Python.
* Web framework for online access.

DATE: PAGE:

7. Non-functional Requirements.

* Security - → Secure login for staff and citizens
  → Encrypt sensitive personal data.

* Reliability - High availability to ensure continuous service.

* Usability - Easy navigation and form filling for citizens.

* Data Integrity - Maintain accurate records of all applications, verifications and issued passports.

8. Preliminary Schedule and Budget.

The development of the Passport Automation System is estimated to take 9 months with a budget of $200,000. This includes planning, development, testing, deployment and staff training.

22/8/25