1. SRS document on Hotel Management System

Introduction
The Hotel Management System provides ease of access to bookings and cancellation by customers.

Purpose:
The purpose of Hotel Management System's SRS document is to provide a base, a blueprint of the whole system. Making the project more understandable regarding the working of the system, user expectations, and the initial cause of the project.

Scope:
The HMS has a wide scope covering tourism, catering, airlines, clubs etc. making it a promising career option.

Overview:
The HMS project is intended for Booking of rooms easily through an online platform where, the receptionist, manager and customer all use the platform and make everyday process easier.

General Description:
The main perspective of this project is to provide an online platform for Booking of hotel rooms with secure payment options and easily done.

Functional Requirements:
* Reservation and booking
* Check-in / Check-out
* Food and restaurant service
* Availability of rooms
* User authentication
* Feedback

Interface Requirements
- User Interfaces
* device with input/output functionalities with a web browser
- Software Interfaces
* any operating system (preferably window)
* any database storing program software installed
* any development end (programing language)

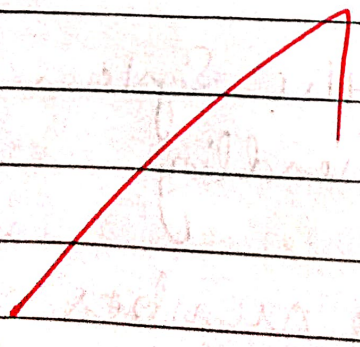
Performance Requirements.
* time it takes for the system to respond
* Efficiency of the System
* Transaction handling
* Data Storage
* handle more number of concurrent users.
* Optimal Resource Utilization.

Design Constraints
These are the limitations or restrictions placed on designing and implementation of a software system.

* Specific programming language
* assumed to always be developed under the windows operating system
* Integrating with existing systems
* Reporting Capabilities
* User roles and permissions

Non Functional Attributes
* secure data & payments
* Reliable backups
* Easy to use and maintain.

Schedule & Budget
These are the rough estimate of the time required for the project, and the budget (cost) to complete it.



2. SRS document on Credit Card Processing

Problem Statement

Manual and outdated credit card processing methods cause delays, transaction errors, frauds, risks, customers demand quick, secure and reliable payment options, while businesses require accurate transaction tracking and fraud.

SRS document

Introduction

Purpose

The purpose of this document is to define the requirements for a credit card processing system. It explains why the system is needed, its objectives and how it ensures secure and efficient processing of credit card transactions.

Scope

The System ensures secure authorization, authentication, and settlement of credit card payments for merchants and customers it ensures real time validation of cards fraud detection and accurate transaction logging. The system will support online and point of small transactions.

Overview

The system provides a secure platform for processing payments. It iterates with banks, payment gateways, and merchant

applications for approval or decline transa based on card details and account status.

General Description.

Objectives.

- Authenticate card holder details
- Process payments quickly and reliably
- Reduce fraud and maintain accurate transaction records

Users

- Merchants: initiate transaction
- Customers: make payments
- Bank and Payment Gateways: validate and settle payment.

Functional Requirements

- User authentication.
- Transaction authorization.
- Payment Settlement.
- Fraud detection
- Transaction history maintainence.

Interface Requirements.

- Merchant interface: for initializing and tracking transactions.
- payment gateways API: for secure communication with banks.
- Database Interface: For storing card details, logs and receipts.

Performance Requirements

* Each transaction must be processed in < 3 seconds.
* Handle 300 current transactions
* Maintain 99.9% uptime for uninterrupted processing.

Design Constraints

* must comply with PCI-DSS standards for data security.
* Use SSL/TLS encryption for data transmission
* Support your major credit card s (visa, mastercard, AMEX, etc)

Non Functional Requirements.

* Security: End to end encryption and decryption.
* Reliability: continuous uptime with automated failures.
* Scalability: Support for increasing transaction volumes.
* Usability: Simple integration for merchants.

Preliminary Budget and Schedule.

Schedule

Requirement analysis	1 week
Design & Architecture	1 week

Development	3 weeks
Testing & Security	2 weeks
Development	1½ weeks
Total	8 weeks

* Budget

Development	₹ 1,80,000
Security testing	₹ 50,000
Deployment	₹ 20,000
Maintenance	₹ 30,000
Total	₹ 2,80,000

3. SRS document on Library Management System.

Problem Statement:
A Library management System is required to automate book cataloging, borrowing, returning and record management to reduce manual errors and improve efficiency.

SRS Document:
Introduction
- Purpose:
The purpose of this document is to define the requirements of LMS. It serves as a guide for developers, testers librarians and administrators to design, implement and maintain as a system that automate library operations such as cataloging, borrowing, returning and fine managment.

Scope:
The LMS is a digital solution that manages a library's resources efficiently. It will handle book cataloging, member registrations, borrowing/return transactions, fine calculations and generate reports for administrators. The system will provide role-based access for students, libraries and administrators.

General description.
The LMS automates library operations like cataloging, member registration, borrowing, return, fine calculation. It provides real time book availability, search functionality, and reporting tools, ensuring efficient and secure access for students, librarians and admins.

Functional Requirements.
* allow librarians to add, update and delete book records.
* allow students to search for books by title, authors or subject.
* allow members to borrow or return books.
* calculate fines.
* maintain borrowing history for each member.
* support role-based access.

Interface Requirements
- User Interface
  * student - search catalogue, borrowing history
  * librarian - issue / return books, manage inventory.
  * Admin - generate reports, configure system.
- Hardware Interface - Barcode / RFID scanner
- Software Interface - Database, mySQL, library catalog APIs.

- Communication Interface - LAN, Internet

Performance Requirements.
* Support atleast 200 concurrent users.
* process borrowing/return transactions within 2 seconds
* support database of atleast 100000 book records.
* provide 99% uptime.

Design Constraints.
* must support common OS
* Develop using java with SQL backend.
* Comply with institutional IT policies for data security.

Non Functional Attributes
* Security - Role based access control, encrypted member data.
* Usability - Simple and intuitive interface for all users.
* Scalability - Support expansion for multi-branch libraries

Preliminary Schedule & Budget
* Schedule

Requirement analysis	1 week
System Design	1 week
Development	3 weeks
Testing	2 weeks
Deployment	1 week
Total	8 weeks

* Budget

Development	₹ 1,50,000
Testing	₹ 40,000
Deployment	₹ 20,000
Maintenance	₹ 25,000
Total	₹ 2,35,000

4. SRS document on Stock Maintenance System

Problem Statement:
Manual stock management often results in errors in tracking, overstocking and stockouts, so a stock maintenance system is needed to automate inventory recording, monitoring and reporting.

SRS document:-
Introduction:
Purpose
This document defines the requirements for the Stock maintenance System (SMS). It provides details for developers, testers, administrators and business stakeholders to design, implement and maintain a reliable system that manages inventory efficiently.

Scope
The SMS is a web and desktop based system designed to automate inventory management for businesses. It will track stock levels, handle purchases and sales records, send alerts for stock shortages or surpluses and generate reports. The system will improve efficiency, reduce errors and support decision making for managers.

Overview:
* System type : webt desktop application.
* Users: Storekeepers, managers, admins
* Functions: Stock entry, update tracking, alert, reporting
*Benefits: Reduced stock errors, improved visibility, better resource utilization

General Description
The SMS automates inventory operations by recording incoming and outgoing stock, updating stock levels in real time, and generating alerts for low or excess stock. It supports role based access for storekeepers and managers, provides detailed reporting on sales and stock levels, and ensure efficient and error-free management of inventory.

Functional Requirements
* allow entry and update of stock items
* record incoming and outgoing stock.
* display real-time stock levels
* send alerts when stock is low or exceeds defined thresholds
* support role based access.

Interface Requirements
o User Interface
Storekeeper: Stock entry / update
manager: Reporting, alerts
Admin: Users and System Configure

* Hardware Interface: Barcode/RFID Scanners, label printers.
* Software Interface: Database (MySQL), reporting tools
* Communication Interface: LAN for internal use, Internet for remote access.

Performance Requirements
* Process inventory updates within 1 second.
* Handle upto 5000 stock records efficiently
* Generate 25-50 concurrent users without performance issues.

Design Constraints
Tech Stack:
- Frontend: HTML, CSS, Javascript
- Backend: Java / PHP / Python
- Database: MySQL DB
- Must support multiuser role based access.
- Should comply with data accuracy and reliability standards.

Non Functional Attributes
* Security: Authorized access for admins and staff.
* Reliability: Accurate and upto date stock data.
* Scalability: can handle growing product catalogs
* Usability: Simple, intutive interface for quick operations.

* Maintainability: Easy updates for new features or modules

Preliminary Budget and Schedule

Schedule

Requirement Analysis - 1 week
Design - 1 week
Development - 2 weeks
Testing - 1 week
Deployment - 1 week
Total - 6 weeks

Budget

Development ₹ 1,20,000
Testing ₹ 30,000
Deployment ₹ 20,000
Maintenance ₹ 20,000
Total ₹ 1,90,000

5. Passport Automation System

Problem Statement

The traditional passport application and issuance process is often slow, error prone and burdened with manual paperwork. Applicants face long queues, delays in verification and lack of transparency in applications status passport offices struggle with handling large volumes of applications, verifying documents and ensuring secure data management

Introducting

Purpose

The purpose of this document is to define the functional and non functional requirements for PAS. The System is designed to streamline the process for applying, verifying and issuing passports. It will reduce manual intervention, minimise processing delays and provide applicants with a user friendly platform to track their applications.

Scope

The model will send notifications at each stage of processing and ensuring high level of security for sensitive applicant data.

Overview

The PAS will be web based and mobile accessible system with role based access for applicants, passport offices and administrators

General Description

* Users
* System Environment
* Assumptions
* Dependencies

Functional Requirements

* User Registration and authentication
* Application management
* Payment processing
* Appointment Scheduling
* Verification and approval
* Application tracking and notification
* Report and analysis.

Interface Requirements

. User interface

1. Applica- dashboard
2. Official dashboard
3. Admin dashboard

. Hardware interface

1. Biometric devices for fingerprint / iris scan
2. Barcode / QR scanners

. Software interface

1. Database: MySQL / Oracle
2. Frontend: HTML / CSS

3. Backend: Java / Python

Performance Requirements
* System must handle 10000+ concurrent users
* Each application transaction should complete in < 3 seconds

Design Constraints
* must comply with government regulations for data security.
* Data must be encrypted at rest and during transition
* Must support multi language interface

Non Functional Attributes
* Security: Encrypted data storage and secure authentication.
* Reliability: Zero data loss during processing
* Usability: simple and intuitive for non technical users
* Scalability: should support increasing user demand.
* Maintainability: Easy updates for new passport policies.

Preliminary Schedule and Budget

Schedule	
Requirement Analysis	1 week
Design	1 week
Development	2 weeks
Testing	1 week

Deployment	1 week
Total	6 weeks

Budget	
Development	₹ 1, 50, 000
Testing	₹ 40,000
Deployment	₹ 30,000
Maintenance	₹ 30,000
Total	₹ 2, 50, 000