

1. Hotel Management System.

Problem Statement: To develop a software solution to overcome problems faced by current manual processes such as lack of real-time data, inefficient room assignments, slow check-ins, poor communication b/w the people, lost revenue due to unrecorded transaction.

Introduction

The Hotel Management System is a tool for booking the rooms of Hotel rooms online.

1.1 Purpose

The hotel management system is built to develop a centralised, user-friendly, and efficient system that automates hotel operations like room reservation, booking and billing.

1.2 Scope

Front Door Service - to book rooms, track transaction and billing

Housekeeping: to check room schedules, and clean

Reporting: Generating reports on occupancy rates,

1.3 Overview

The HMS will be a comprehensive software solution designed to automate & streamline all core operations of a hotel.

2. General description

The HMS will be a comprehensive system with a simple, intuitive user interface. It will be accessible via a web browser and is designed for various user roles, including administrators, front desk staff, and housekeeping.

3. Functional Requirements

The system will provide

- Reservation Management:

- Guest & Front Desk Operations
- Billings & Payments
- Room Status & Housekeeping
- Reporting

4. Interface Requirements

- User Interface (UI): Clean, intuitive, responsive and easy to navigate UI.
- Software Interface: To provide payment gateway & email service providers for automated communications.
- Hardware Interface: System compatible with web browsers.

5. Performance Requirements

- The system must handle 50 concurrent users
- Guest check-in/check-out transaction within 5 seconds.
- 99.9% uptime reliability.

6. Design Constraint

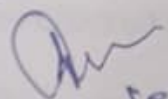
- The system must be built on a secure web architecture
- All data especially guest information must adhere to data protection regulations & standard

7. Non-functional requirements

- Security: System must be protected against vulnerabilities
- Scalability: architecture must accommodate growing demand
- Usability: Easy flow of usage must be assured
- Maintainability: The code must be well-structured
- Reliability: Must have minimal failure rate

8. Preliminary Schedule and Budget

- Schedule: The project is estimated to take approximately 9 months from start of requirements analysis to final deployment
- Budget: The high level budget is 10 to 13 lakh rupees which includes all personnel, software & contingency costs


19/10

Credit Card Processing

Problem Statement:

Current credit card processing methods often lack security, real-time transaction validation, and centralized reporting, leading to increased fraud risks, operational delays, and difficult reconciliation. Merchants need a reliable, secure & efficient system that can handle a high volume of transactions.

1. Introduction

1.1 Purpose

This document outlines the software requirements for a credit card processing system.

The software is used to manage credit card details of a bank.

1.2 Scope

CCPS will handle the entire lifecycle of credit card transaction, from authorization to settlement.

- Transaction Processing: Authorizing, capturing, refunding, and voiding credit card payments.
- Security: Adhering to PCI DSS standards.
- Reporting: Generating detailed transaction

1.3 Overview

CCPS will be secure, robust, scalable backend ~~server~~ middle ware b/w a merchant's application & financial institutions, facilitating

The secure transfer of cardholder data.

2. General Description

The CCPS will operate as a Software-as-a-Service platform. Merchants will integrate the CCPS API into their applications to process credit card transactions. The system will validate card information, communicate with the payment networks, and return a transaction status.

3. Functional Requirements

- Transaction Authorization
- Transaction Capture & Settlement
- Refund & Void
- Tokenization
- Reporting

4. Interface Requirement

- API Interface : for client request/response service
- User Interface : web-based dashboard
- External Interface : major payment networks.

5. Performance Requirement

- Transaction speed : 1.5 second.
- Throughput : 100 transaction per second.
- Availability : 99.99% uptime

6. Design constraint

- **Security**: The system must be built in strict accordance with PCI DSS level 1 standards.
- **Architecture**: A microservice architecture is required for fault tolerance.
- **Technology Stack**: The system must have a robust.
- **Data Handling**: must not store sensitive cardholder data after it has been tokenized.

7. Non-functional requirements

- **Performance**: Handle 5,000+ transactions/minute, <2s response time.
- **Reliability**: 99.99% uptime, automatic failover.
- **Security**: PCI DSS compliance. AES-256 storage.
- **Scalability**: Horizontal scaling for peak loads.
- **Usability**: Intuitive UI, developer-friendly APIs.
- **Maintainability**: Modular design, easy updates.

8. Preliminary Schedule & Budget

- **Schedule**: The project is estimated to take 6 months.
 - > Requirements: 3 weeks
 - > Design: 4 weeks
 - > Development: 12 weeks
 - > Testing: 6 weeks
 - > Deployment: 3 weeks
- **Budget**: The high level budget is 80 lakhs including all personnel, software & contingency costs.

Library Management System

Problem Statement

Outdated manual processes in libraries lead to inefficiencies like inaccurate records and poor services, necessitating a modern, automated Library Management System to improve efficiency & overall user experience.

1. Introduction.

1.1 Purpose.

The purpose of this document is to outline the requirements for a new Library Management System (LMS). The system aims to automate & streamline all core library functions, including book cataloging, member registration, book borrowing and returns, and fine management.

1.2 Scope

The scope of this project is to develop a software that handles:

- Book management
- Member management
- Transaction management
- Fine management
- Reporting

1.3 Overview

LMS is a comprehensive software solution that deals with all core functionalities of a library.

2. General Description

The cms will be a web-based application accessible to both library staff & members. It will feature a secure login for staff & a separate interface for members to search for books and view their account details. The system will be built on a normalized database to ensure data integrity and accessibility.

3. Functional Requirement

The system must perform the following functions:

FR-1: Book Management

The system must allow a librarian to add ~~up~~ a new book, update existing book, remove a book from catalog, and search for book by title.

FR-2: Member Management

The system must allow a librarian to ~~add~~ register, update, suspend or reinstat a member account.

FR-3: Transaction Management

The system must allow to issue a book and record the borrow date & due date.

FR-4: Fine Management

The system must calculate fines based on a predefined rate for each overdue book.

4. Interface Requirements

- 4.1 User Interface: A secure, intuitive, & easy-to-navigate dashboard for staff & members.
- 4.2 Hardware Interface: The system will be a web application and will not require specific hardware interface beyond a standard computer with internet connection.
- 4.3 Software Interface: The system will interface with a ~~RDBMS~~ ~~RDBMS~~ to store all data.

5. Performance Requirements

- Response Time: All key operations should have a response time of less than 3 seconds.
- Concurrency: The system must support at least 5 simultaneous librarian & 50 simultaneous member users with performance degradation.
- Database Capacity: Database should handle 100,000 book records.

6. Design Constraints.

- Technology stack: The system will be developed using a modern web development framework.
- Security: The system must use HTTPS to ensure secure communication.
- Scalability: The system must be designed to be scalable for future growth.

7. Non-functional attributes.

- Usability: The system's UI must be intuitive and easy to use for all users.
- Reliability: Data integrity & backup mechanisms must be in place to prevent data loss.
- Maintainability: The code must be modular to allow for easy maintenance.
- Security: The system must have robust security measures.
- Portability: The system should be accessible from any device with a web browser.

8. Preliminary Schedule & Budget.

- Preliminary Schedule: The project is estimated to approximately take 9 months from start of requirement analysis to final deployments.
- Budget:
 - Labour: £4,80,000 (Developers, QA engineers ...).
 - Software: £60,000
 - Contingency: £10,000
 - Total Estimated Budget: £5,50,000.

Stock Maintenance System

Problem Statement

Inefficient manual stock management leads to errors, stock issues, and financial losses, requiring an automated system to streamline inventory and provide accurate, real-time data.

1. Introduction

- 1.1 Purpose: To define the requirements for a new stock maintenance system to automate and streamline inventory management, sales tracking, and supplier order processing.
- 1.2 Scope: The system will manage the complete lifecycle of a product within a business, including stock intake, sales, stock adjustments, and reporting.
- 1.3 Overview: This document outlines the system's ~~functional and non-functional~~ requirements, interfaces, performance, & schedule and budget.

2. General Description

The SMS will be a centralized, web-based application accessible to different user roles. It will maintain a real-time database of all stock items, their quantities, and transaction history.

The key user roles are:

- Administrator: Manages system settings, user accounts & generates reports.
- Inventory Manager: Manages stock levels.
- Sales Staff: Records sales transactions.

3. Functional Requirements

- FR-1: Stock Management: The system must allow users to add, update, and remove stock items with details like SKU, name, and quantity.
- FR-2: Sales Transaction: The system must allow sales staff to record a sale, automatically deducting the sold items from the stock.
- FR-3: Supplier Orders: The system must enable the generation and tracking of purchase orders to suppliers.
- FR-4: Stock Alerts: The system must trigger low-stock alerts for items that fall below a threshold.
- FR-5: Reporting: The system must generate reports on sales history, current stock levels, and best-selling items.

4. Interface Requirements

- User Interface: A secure, intuitive, and easy-to-navigate dashboard for all users
- Hardware Interface: Standard computer with an internet connection
- Software Interface: The system will interface with a relational database and standard web browsers.

5. Performance Requirements

- Response Time: All key operations must have a response time of less than 2 seconds
- Concurrency: The system must support at least 10 simultaneous users without performance degradation.
- Scalability: The system must be scalable to manage minimum of 5,000 unique stock items

6. Design Constraints

- Technology Stack: The system will be built using a modern web framework.
- Security: Data must be secured using HTTPS, and sensitive information like user credentials must be encrypted.

7. Non-Functional Attributes

- Usability: The user interface should be simple & intuitive, minimizing the need for extensive training.
- Reliability: The system must be available 99.5% of the time, with robust data backup.
- Maintainability: The code should be modular and well-documented to facilitate future updates and bug fixes.
- Security: Role-based access control must be implemented to ensure data security.

8. Preliminary Schedule and Budget

8.1 Schedule:

- Phase 1 (1 month): Requirement Analysis
- Phase 2 (3 months): Development & Implementation
- Phase 3 (1 month): Testing & Quality Assurance
- Phase 4 (2 weeks): Deployment & Training

8.2 Budget:

- Labour: £ 4,00,000 (2 developers, QA engineer for 4 months)
- Software & Tools: £ 50,000
- Contingency: £ 50,000
- Total Estimated Budget: £ 5,00,000

Passport Automation System

Problem Statement

The existing manual passport application process is inefficient and lacks transparency, causing delays and frustration. An automated system is needed to streamline applications, improve tracking, and enhance overall efficiency and user experience.

1. Introduction

1.1 Purpose

To define the requirements for a new Passport Automation System to digitize and streamline the entire passport application, verification and issuance process.

1.2 Scope

The system will manage the full lifecycle of a passport application, including online application submission, document verification, appointment scheduling, and status tracking.

1.3 Overview

The document outlines the system's functional and non-functional requirements, interfaces, performance criteria, design constraints, & a schedule. This document guides the development of passport automation system and ensure all stakeholders understand the project's goals.

2. General Description

The Passport automation system will be a web-based application with a secure backend database. It will serve different user roles, each with specific access rights:

- Applicant: Submits and tracks their application
- Verification Officer: Verifies submitted documents and application details.
- Administrative Staff: Manages appointments, fees, and application status.
- System Administrator: Manages user accounts.

3. Functional Requirements

- FR-1: Online Application: The system must allow applicants to fill out & submit an application form online.
- FR-2: Document upload: Applicants must be able to upload scanned copies of required documents.
- FR-3: Appointment Scheduling: The system must provide a module for applicants to book a physical appointment for biometric data collection.
- FR-4: Status Tracking: The system must allow applicants to track the real-time status of their application using a unique reference number.

- FR-5: Verification and Approval: The system must provide an interface for officers to review, verify, and approve or reject application.
- FR-6: Fee Payment: The system must integrate with an online payment gateway for fee collection.

4. Interface Requirements.

- User Interfaces: The system will feature a user-friendly public portal for applicants and a secure administrative dashboard for staff.
- Hardware Interfaces: Standard computers, webcams, and fingerprint scanners for biometric data capture at physical centers.
- Software Interfaces: The system will interface with a secure database, an online payment gateway.

5. Performance Requirements.

- Response Time: Page loads and key transactions must complete within 3 seconds.
- Concurrency: The system must handle a minimum of 500 concurrent users without performance degradation.
- Scalability: The system must be able to process and store data for a minimum of 10,000 applications per day.

6. Design Constraints

- **Security**: The system must comply with all relevant data privacy laws and government security protocols. All data must be encrypted in transit.
- **Technology Stack**: The system will be built using a robust, government-approved technology stack.
- **Accessibility**: The public-facing portal must be accessible to users with disabilities.

7. Non-Functional Attributes

- **Usability**: The interface must be intuitive for users.
- **Reliability**: The system should have 99.9% uptime.
- **Security**: Multi-factor authentication will be implemented for all staff accounts.
- **Portability**: The system must be deployable on different server environments.

8. Preliminary Schedule and Budget.

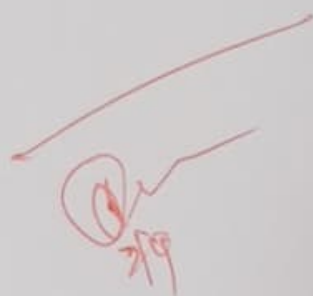
Schedule:

- **Phase 1 (2 months)**: Requirement Analysis & Design.
- **Phase 2 (4 months)**: Development & Implementation.

- Phase 3 (2 months): Testing & Quality Assurance.
- Phase 4 (1 month): Deployment & Training.

8.2 Budget:

- Labour: £15,00,000 (team of 5 for 8 months)
- Software & Licenses: £2,00,000
- Infrastructure: £3,00,000
- Contingency: £3,00,000
- Total Estimated Budget: £23,00,000



A handwritten signature in red ink, followed by the date '2/9' written below it.