Develop a problem statement and a complete IEEE standard SRS document with several requirements for the following:-

(i) Hotel Management Systems

Problem statement:

The Hotel Management System aims to automate hotel operations such as room booking, check-in/check-out, billing and staff management to eliminate manual errors, improve efficiency and enhance customer experience

IEEE SRS document:

Project title: Hotel Management System (HMS)

I. Introduction

1.1 Purpose of the Document

This document specifies the requirements for the Hotel Management System (HMS). It defines functionalities, interfaces, constraints and performance expectations for developers, testers, admins and end users

1.2 Scope of the Document

The HMS will automate hotel operations such as reservations, check-in/check-out, billing, staff management and reporting. Customers can book rooms and make payments, staff can manage operation and administrators can oversee hotel services

6 Design Constraints

- must comply with PCI-DSS standards
- use encryption for all cardholder data
- Daily secure database backups
- Restricted role-based users

7 Non-functional Requirement

- Security : End-to-End encryption, tokenization
- Reliability: Failover and redundancy
- Usability : Simple UI for merchants/customers
- Portability : Compatible with POS, web & mobile
- Scalability : Handle increased transaction loads

8 Preliminary Schedule and Budget

8.1 Schedule

- Month-1 : Requirements
- month-2 : Design
- Month 3-4: Development
- Month-5 : Testing
- month-6: Deployment

8.2 Budget

- Hardware / Server : ₹1L
- Software / Tools : ₹50k
- Development : ₹3L
- Testing : ₹1L
- Miscellaneous : ₹50k

Total : ₹6L

2. General Description

The LMS will be a centralized web/desktop system.
- Members: search, borrow, return, and reserve book
- Librarians: manage catalog, track issues/return, collect fines
- Administrators: manage member, staff and reports

3. Functional Requirements

- User registration/login
- add, update and delete books
- search and reserve reserve book
- issue and return book with due dates
- Calculate and manage fine
- member and staff management
- generate daily/weekly/monthly report

4. Interface Requirements

- UI: simple web/desktop interface
- Hardware: PC or terminal
- Software: SQL database, library catalog API
- Communication: LAN/internet

5. Performance Requirements

- support 200+ concurrent user
- book search < 2s
- issue/return process < 5s
- reports generated within 10s

6. Design Constraints

- database must backup daily
- access restricted by role (student, librarian, member)
- platform ~~should~~ independent implementation
- Compliance with data privacy policies

7. Non Functional Requirements

- Security : encrypted login & secure cloud storage
- Reliability : 99% uptime
- Usability : easy for librarians & students
- portability : works on multiple platforms
- Scalability : expandable for multiple library branches

8. Preliminary Schedule and Budget

8.1 Schedule (6 months)

month-1 : Requirements
Month-2 : Design
month-3,4 : Development
Month-5 : Testing
month-6 : Deployment

8.2 Budget

- Hardware / Server : ₹ 60k
- Software / DB Tools : ₹ 30k
- Development : ₹ 1.5L
- Testing : ₹ 40k
- Misc : ₹ 20k

Total : ₹ 3L

(iv) Stock Maintenance System

problem statement:

Manual stock tracking is error prone, time-consuming and often leads to overstocking or stockouts; hence, a stock maintenance system is required to automate inventory management, ensure accuracy and provide real-time visibility of stock levels

1. Introduction

1.1 Purpose

This document defines requirements for the stock maintenance system, to guide developers, testers and store managers

1.2 Scope

The SMS will track inventory levels, manage product details, handle stock-in/stock-out operations and generate reports. It ensures accurate stock management, reduces losses, and provides real-time visibility of inventory

1.3 Overview

Covers system description, functional & non-functional requirements, interface, performance, design constraints and preliminary schedule/budget

Q:) Library Management System

Problem Statement:

Manual library operations such as book issuing, returning and catalog management are time-consuming, error-prone and inefficient. Therefore, a Library Management System is required to automate book tracking, member management, and transactions, ensuring accuracy and quick access to resources.

1. Introduction

1.1 Purpose

This document specifies requirements for the Library Management System (LMS), to guide developers, testers, ~~trainers~~ librarians and administrators.

1.2 Scope

The LMS will automate book cataloging, issue/return tracking, member management, fines and reporting. It provides student and staff quick access to resources, reduces manual errors and improves efficiency

1.3 Overview

Covers descriptions, functional and non-functional requirements, interface details, performance, constraints and preliminary schedule and budget

3. Use Specific Functional Requirements

-) submit applications online
-) upload and verify documents
-) review, approve, reject
-) track application status
-) generate approval schedule

4. Interface Requirements

-) web portal integration
-) mobile app interface
-) payment gateway integration

5. Performance Requirements

-) must handle 1000+ concurrent users

6. Design Constraints

must meet e-governance security protocols

7. Non-functional

must be secure, reliable, user-friendly, scalable

8. Preliminary schedule and budget

8.1 schedule (6 months)
month 1: design
month 2,3: development
month 4: testing
month 5,6: deployment

2. General Description

The SMS will be a centralized system for managing inventory
- Admin: Add/update/remove stock items, manage users and generate reports
- Staff: update stock entries, check availabilities & manage transactions
- System: Alert for low stock & maintain logs

3. Functional Requirements

- User registration/login
- add/update/delete product details
- record stock-in and stock-out operations
- maintain supplier & customer records
- generate alerts for low stock
- generate daily/weekly/monthly reports

4. Interface Requirements

- UI: web/desktop interface with dashboard
- Hardware: PC or terminal
- Software: SQL database, barcode scanner integration
- Communication: LAN / Internet

5. Performance Requirements

- Handle 1000+ concurrent users
- update stock transactions in < 1s
- generate reports within 5s
- real-time alerts for low stocks

6. Design Constraints

- Daily backup of stock data
- Role-based access (admin / staff)
- Platform-independent technology
- Compliance with security / record policies

7. Non-Functional Requirements

- Security: Encrypted login & secure transaction logs
- Reliability: 99% uptime
- Usability: Simple for staff with minimal training
- Portability: Compatible with PCs & web browsers
- Scalability: Support multiple warehouses / branches

8. Preliminary Schedule & Budget

8.1 Schedule (6 months)

Month-1 : Requirements
month-2 : Design
Month-3,4 : Development
month-5 : Testing
month-6 : Deployment

8.2 Budget

- Hardware / servers : ₹ 50K
- Software / DB Tools : ₹ 30K
- Development : ₹ 1.5L
- Testing : ₹ 40K
- Misc : ₹ 20K

Total : ₹ 2.9L

classmate
Date
Page

§2 Budget

Requirement & design : ₹12k
Development : ₹5k
Testing and QA : ₹ 10k
Deployment & Training : ₹ 13k
Maintenance : ₹10k

2/9

(V) Passport Automation System

Problem statement:

Manual passport application and process often lead to long queues, delays and errors in document handling.

SRS

1. Introduction

1.1 Purpose

Simplify and digitise passport application, verification & issue

1.2 Scope

The system will support online application, document verification, payment integration and tracking

1.3 Overview

Includes applicant portal, admin portal, verification system, and integration with government database

2. General Requirements

→ E-governance application integrated with govt records.
→ Functions include application submission, status tracking, document verification and passport issuance
→ Users include: applicant, passport officials & administrators

5 Performance Requirements

- Support 100+ concurrent users
- real-time room updates (<2s)
- Bill generation within 5s
- Secure online payments in 3-5s

6 Design Constraints

- Compliance with data privacy and security laws
- Daily database backups
- Restricted admin/staff access
- Platform independent development (Java/Python/.NET)

7 Non-functional Requirements

- Security : Encrypted passwords & transactions
- Reliability : 99% uptime
- Usability : Simple, intuitive, UI
- Portability : works on browsers & mobiles
- Scalability : Expandable to multi-branch hotels

8 Preliminary Schedule & Budget

8.1 Schedule (6 Months)

- Month-1 : Requirement Analysis
- Month-2 : System Design
- Month-3,4 : Development
- Month-5 : Testing
- Month-6 : Deployment

8.2 Budget

- Hardware & Hosting : ₹50,000
- Software & Licenses : ₹25,000
- Development : ₹1,50,000
- Testing & Deployment : ₹50,000
- Miscellaneous : ₹ 25,000

Total : ₹ 3,00,000

2 General description

The CCPS acts as an intermediary between customers, merchants and financial institutions

- Customer : initiates payment using credit card
- Merchant : Accepts payments and requests authorization
- Bank / Issuer : validates, authorize and settles payment
- Admin : Monitors transactions and generates reports

3 Functional requirements

- Validate card details (number, CVV, Expiry)
- Authorize ~~trans~~ transactions with issuing bank
- Support online / offline transaction modes
- Generate transaction receipts
- Handle refunds / reversals
- Fraud detection and alerts
- Daily / weekly / monthly transaction reporting

4 Interface Requirements

- UI : web dashboard for merchants & admin
- Hardware : POS machines, PCs, mobile devices
- Software : Integrations with banking APIs, payment gateway
- Communication : Secure internet with encryption (TLS/SSL)

5 Performance Requirements

- Authorize transactions within 2-3 seconds
- Handle 500+ concurrent transaction
- 99.9% uptime for continuous operation

1.1 Overview :

This SRS outlines the system description, functional and non-functional requirements, interface specifications, design constraints, performance needs and a preliminary schedule and budget.

2 General Descriptions

The HMS is a centralized platform available as a web application and optionally desktop. Users include:
→ Customers ← Book rooms, check-in/check-out, pay bills
→ Staff - manage reservations, billing, services
→ Administrators ← manage staff, rooms, reports

3 Functional Requirements

* User registration and login
* room availability display & booking
* check-in / out management
* automated bill generation
* online / offline payment support
* Staff / service management (admin)
* customer feedback collection
* report generation (daily / weekly / monthly)

4 Interface Requirements

User-interface : responsive web UI
Hardware : PCs, tablets, smartphones
Software : SQL database, payment gateway
Communication : Internet for online booking

(ii) Credit Card Processing System

Problem statement:

Manual or outdated credit card transaction method are prone to delays, error and security risk; hence, a credit card processing system is required to securely validate, authorize, and settle transactions in real time while ensuring compliance with financial standard.

IEEE SRS document:

I Introduction

1.1 Purpose of the document

This document defines the requirement for the Credit Card Processing Systems, intended for developers, testers, banks, merchants and regulatory auditors.

1.2 Scope

The CCPS will validate, authorize and process credit card transition between customers, merchants, and banks. It will support real-time authorizations, fraud detections, billing and secure settlement.

1.3 Overview

Covers system description, functional and non-functional requirements, interface specifications, performance, constraints & schedule