LAB-1

SRS DOCUMENT

HOTEL MANAGEMENT SYSTEM

A> Problem statement

The hospitality industry is constantly evolving with guest expectations rising and competition fiercer than ever. With the rising demand for efficient operations, streamlined bookings and personalized guest experiences, selecting the best hotel management software becomes crucial to enhance efficiency and guest satisfaction.

B> IEEE standard Requirement Document

I Introduction

⇒ Purpose of the Document

- Supercharge efficiency: streamline operations from front desk to back desk office.
- Elevate guest experience: convenience and personalisation
- Reduce errors & save time: automate repetitive tasks
- Enhance security: Data protection and business control

⇒ Scope

- Reservation Management
- Room assignment and availability
- Billing and invoicing

- Housekeeping management
- Inventory management
- Reporting and analytics

⇒ Overview

A HMS acts as the central nervous system of a Hotel, streamlining operations by managing all aspects of a business in one place.

II General Description

Having an efficient HMS allows you to spend more time taking care of the needs of your guests and avoid time on repetitive administration tasks. It helps minimise frustrations and easily manage your hotel property.

III Functional Requirements
- Food services: provide menu to guests
- Booking / Reservations: sign in, booking options
- Housekeeping: schedule and organise housekeeping services
- Reporting and analytics

IV Interface Requirements

⇒ Software interface
- web server: windows / Linux Os
- Database server: MongoDB, etc.
- Development end: Java, HTML, JS etc.

→ Hardware interfaces
- Server side: Monitors, RAM 4 GB, Disk 10GB
- Client side: Processor, Monitor, RAM 512M Disk 2GB
- Communication interface HTTP/HTTPS

V Performance Requirements
- Quick data base updation
- Query results should be quick
- Efficient login system
- Responsive customer inquiry
- Visually pleasing UI

VI Design constraints
- Reduced connection b/w staff and guests
- Reliance on Internet for cloud based software
- Cyber attack risk

VII Non-Functional Requirements
- Performance - efficiency of the software
- Safety - client safety, software security
- Security - protection from external threats
- User satisfaction - Best service to the client to enhance experience

VIII Preliminary Schedule and Budget

⇒ Schedule

Phase 1: Planning & Requirement gathering (1 month

Phase 2: System design (1 month)

Phase 3: Development (3 months)

Phase 4: Testing (1 month)

Phase 5: Deploy and Train (1 month)
Phase 6: Maintenance and support (ongoing)

→ Budget estimation
- Personell cost ≈ 2 Lakhs (Project manager, Developers, UI/UX, QA, System Admin)
- Hardware & licenses ≈ 1 Lakh
- Contingency and Miscellaneous ≈ 50,000

Total estimate ≈ 3 Lakhs

CREDIT CARD PROCESSING

A) Problem Statement

Credit card processing is vital for business to accept electronic payments securely and efficiently. In this digital world, CCP is essential for verification, ensuring funds, easy money transfer.

B) SRS Document

I Introduction

⇒ Purpose
CCP is a critical component in processing transactions, whether online or instore or over the phone. It is slowly becoming one of the most convenient and acceptable modes of online payment.

⇒ Scope
- Stakeholders - card owners, merchant, bank card networks
- Transaction process
- Tech infrastructure
- Security standards
- Fee and revenue models

II General Description

The processing involves, initiation, data transmission, authorisation, approval, fund transfer.

III Functional Requirements
- User authentication - validate card details
- Payment processing - generate invoice & report
- Transaction management - invoice generation
- Security & compliance - fraud detection, OTP
- Funding & settlement - allow refund and dispute resolution.

IV Interface Requirements
- UI
- Payment screen
- Transaction history
- Payment API

V Performance Requirements
- Transaction speed - within 2-5 seconds
- Increased throughput
- Concurrency handling without performance degradation
- Availability
- Scalability

VI Non-functional Requirements
- Increased security - comply with PCI DSS standards
- Maintability - system must allow modular updates without downtime
- Portability - cloud & on-premise environment
- Auditability & compliance

VIII Preliminary schedule & Budget

⇒ Schedule

Phase	Duration
Requirements gathering	1 mo.
System design	1.5 mo.
Core development	3 mo.
Frontend Dev	2 mo.
All integration	2 mo.
Testing	2 mo.
Deployment	1 mo.

LIBRARY SYSTEM MANAGEMENT

A> Problem Statement

Manual library management is slow, error-prone and inefficient. Tasks like book search, issue/return tracking, fine calculation become difficult with paper records. An automated system is needed to manage books & users more efficiently.

B> SRS Document

I Purpose

The purpose of this system is to computerise the management of library operations. It will allow librarians to manage books, members and borrowing records efficiently.

II General Description

Client-server with central book catalog database of books. Functions include catalog management, borrowing/returning user registration etc.

III Functional Requirements

- Add, update & delete book records
- Register members & maintain accounts
- Track borrowing & returning
- Calculate fines for late returns

IV Interface Requirements

- Web & Desktop UI
- Barcode scanner integration for books
- secure login system

V Performance

- support 100+ concurrent users
- search results in <2s
- 99% uptime

VI Design constraints

Database MySQL

Role based access (member, librarian, admin)

VII Non-functional Requirements

- usability
- security
- Reliability
- scalability

VIII Preliminary schedule & Budget

⇒ schedule

5 months → Total

Design : 1 month

Development : 2 months

Testing and Deployment : 2 months

⇒ Budget : ₹35,000

Requirements & Design : ₹1000

Development : ₹18000

Testing & QA : ₹5000

Deployment & training : ₹1000

Maintenance : ₹4000

STOCK MAINTENANCE SYSTEM

A) Problem Statement

Tracking stock manually is inefficient and prone to errors. A stock maintenance system is needed to monitor inventory in real time, automate stock updates, and provide accurate reports for decision making.

B) SRS Document

I Purpose

Automate stock tracking, replenishment, and reporting to reduce errors & improve efficiency.

II General Description
- Centralized inventory database
- Functions include stock entry / update, reorder alert
- Users include store managers, warehouse staff & administration

III Functional Requirements
- Add / update stock records
- Generate low stock alerts
- Track stock movement
- Generate reports

IV Interface
- Web & desktop UI
- Barcode / RFID integration

V Design constraints
- SQL database
- Role based access

VI Non-functional Requirements
- Reliable
- Scalable
- User-friendly

VII Preliminary Schedule & Budget

⇒ Schedule ⇒ 5 months
Design : 1 month
Development : 2 months
Testing & Deployment : 2 months

⇒ Budget : ₹35,000
Requirements & Design : ₹4000
Development : ₹18000
Testing & QA : ₹5000
Deployment & training : ₹4000
Maintenance : ₹4000

PASSPORT AUTOMATION SYSTEM

A) Problem statement

Manual passport application & processing ~~after~~ often leads to long queues, delays & errors in document handling. A passport automation system is required to digitize the process allowing online applications, secure document uploads, automated tracking & faster verification.

B) SRS Document

I Purpose
Simplify & digitize passport application, verification & issue to support online applications, document verification, payment integration & tracking

II General description
- E-governance application integrated with gov. records
- Functions include application submission, status tracking, fee payment verification & passport issuance.
- Users include applicants, passport officials & administrators

III Functional Requirements
- Submit applications online
- Upload & verify documents
- Process payments securely
- Track application status
- Generate appointment schedule

IV Interface
- Web portal integration
- Mobile app interface
- Payment gateway integration

V Performance
Must handle 10000 + concurrent users

VI Design constraints
Must have e-governance security protocols

VII Non-functional requirements
Must be secure, reliable, user-friendly and scalable

VIII Schedule & Budget

⇒ Schedule ⇒ 6 months
Design: 1month
Development: 3 months
Testing: 1month
Deployment: 1month

→ Budget: ₹ 1,00,000
Requirements & Design: ₹12,000
Development: ₹ 55000
Testing & QA: ₹ 10000
Deployment & training: ₹ 13000
Maintenance: ₹ 10000