

Week - 01.

1. Hotel Management System

a. Problem Statement

Notes

Manual hotel operations are slow & error-prone; the data registered by customers can be lost. Hence, an ~~automated~~ ~~Hotel Management System~~ is needed to streamline bookings, billing, ~~and~~ ~~customer service~~, and food ~~availability~~.

b. SRS document

a.1. Purpose of the document

~~Purpose of this is that we can enhance the system of the hotel being used to get rid of the traditional methods and achieve & benefit hotel staff, management & customers.~~

a.2. Scope

~~The world is changing so as the scope of hotel management. Today, hotel management is not only confined to hotels but has gone deep into tourism, catering, airlines, clubs, etc. making it a very~~

a.3 Overview

Hotel management System is intended for Booking of Room through an online platform.

The 3 end user.

- Receptionist - They have access to update or modifying booking details.
- manager - manager will be able to view the financial reports & able to update the room information such as any addition in room.
- Customer - Customer is able to check room's availability.

Other different services for the customer End user

02. General description

HMS provides many functionalities such as room reservation, guest check-in check-out, house keeping, staff scheduling, billing, & generating reports.

03. Functional requirements

- room booking (online & offline) - The customer can book availability - The room availability pre-book
- check in/out - The customer data after vacating
- Payment - gateway, secure & consistent processing
- Staff management
- Report generation - The stock & customer details for further improvement

04. Interface requirements

- user-friendly web interface for customers to book rooms.
- Admin dashboard for hotel manager.
- Staff portal.
- Database interface.
- Integration with third-party service.

05. Performance requirements

- the system should handle multiple users simultaneously.
- Response time.
- System uptime.
- maximum error rate allowed : less than 1%.

06. Design constraints

- System must support both web & mobile devices.
- payment method should be secure.
- limited to available hardware & internet.
- Should follow local hotel regulations & data protection laws.

07. Non-functional requirements

- Security.
- Probability.
- Reliability.
- Reusability.
- Scalability.
- Data integrity.

8. Preliminary Schedule & Budget

→ Development time : 4 - 6 months

→ Budget Estimate - 8 - 12 lakhs

→ Phases

• requirement analysis [2 weeks]

• Design [3 weeks]

• Implementation [10 weeks]

• Testing [3 weeks]

• Deployment & Training [2 weeks]

Jobs

Schedule (in weeks)

Budget (in rupees)

1. requirement analysis	2 weeks	1. Planning & Design	5,000 - 10,000
2. Design	3 weeks	2. Development	15K - 25K
3. Implementation	10 weeks	3. Database	3K - 5K
4. Testing	3 weeks	4. Hosting & Domain	5K - 8K
5. Deployment & Training	2 weeks	5. Third-party APIs	3K - 7K
		6. Testing	4K - 6K
		7. Security	2K - 4K
		8. maintenance	5K - 10K (per year)

2. Library Management System

a. Problem Statement

LBSYS faced the problem from their readers like returning the books, book availability and reader details were mixed and tracking would be difficult so by library management system we can overcome all such problems LBSYS automate library operations like book mgmt, issuing & returns.

b. SRS Document

Purpose of this document

→ The purpose of this project or system is that we can digitalize or make the things smarter by using bar code, qr scanner code, Scanners for books & customer book returning date notification through mail or SMS, and book stock for the next readers & user satisfaction.

Scope

- LMS will automate & streamline the day-to-day operations of a library.
- maintaining records of books
 - registering & managing members
 - issuing & returning of books with due date.
 - managing fine for late returns.

3. Overview

- LMS a software designed to manage the daily operations.
- It provides digital platform for organizing & handling books, members & transactions.
- This reduces the manual efforts to maintain records, issuing / returning books & tracking availability.
- This ensures accuracy, save time & enhances efficiency for librarians & users.
- Users: librarians, administrators & members.

2. General description.

- automates the tasks
- provides easy book management & borrowing services.
- ~~features: System Environment; Web app + database~~
 - Add / update / delete / search
 - Register / manage members
 - Borrow / issue / receive books

3. Functional requirements

- ~~member:~~ registration / login, search books, borrow / return, view borrowing history.
- ~~librarian:~~ Add / remove books, issue / return a book.

3. system : Auto fine calculation for late return , unique ID & track availability

4. Interface requirements

- UI : simple to use , search & dashboard
- Hardware : library PC's
- Software : MySQL . Nodejs for Backend.

5. Performance requirements

- handle 500+ member at a time.
- able to issue or collect thousand of book at a time.
- concurrent access by 50+ user.
- Data backup daily.

6. Design constraints

- must support barcode/QR Code Scanning of books.
- Budget limits to add AI features
- compliance with IT / Security features

7. Non-functional attributes

- security : Role based accc , except user credential
- usability : simple UI easy to understand and use
- reliability : Ensuring no data loss in transaction.

- scalability : support growing book inventory & members.
- maintainability : easy to update book records & UI features

8. Preliminary Schedule & budget

- requirement analysis - 2 weeks
- Design - 2 weeks
- Development - 5 weeks
- testing - 1.5 weeks
- Deployment - 1 week
- Budget estimation : 4-5 lakte.

Schedule (in Weeks)

1. Requirement and analysis	2 weeks
2. Design	2
3. Development	5
4. Testing	1.5
5. Deployment	1
6. Budget estimation	

Budget Estimation (in Rupee)

1. Planning & Design	40,000
2. Development	1,50,000
3. Database & Hosting	70,000
4. Testing & Security	70,000
5. Training & Support	80,000
6. maintenance	60,000

Credit Card Processing

Problem Statement

A System to securely process, verify, and manage credit card transactions efficiently.

SRS Document

I. Purpose of this document

The purpose of this document is to define the objectives, scope, and functional requirements of the credit card processing system. It aims to ensure secure, accurate and efficient handling of credit card transaction including authorization, verification and payment processing while minimizing errors and enhancing user convenience.

Scope

The credit card processing system will provide a secure and efficient platform for handling credit card transactions. It will cover functionalities such as card verification, transactions authorization, payment processing & record management. The system aims to minimize fraud, reduce processing errors and ensure faster and more reliable transactions for both user & merchants.

Overview

It is designed to streamline and secure online & offline payment transactions. It verifies card details, authorizes payments, and processes transactions efficiently while maintaining high level of security. The system ensures smooth interaction b/w user, bank & merchant reducing errors, preventing fraud & enhancing payment experience.

2. General Description

This security manager handles card authentication, transaction authorization and payment settlement, ensuring fast, accurate, and fraud-free transactions for customers and merchants.

3. Functional requirements

- User Authentication - Verify customer credentials before processing transactions.
- Card verification - Validate credit card details like number, expiry date & CVV.
- Transaction Authorization - Approve / decline transactions.
- Payment Processing - Handle successful payments and update balances.
- Transaction Records - maintain logs of all payments for future reference
- Fraud Detection - Identify & block suspicious or unauthorized transactions

- Notifications - Send payment confirmation or failure alerts via sms/email.

4. Interface requirements

- User interface - Simple, Secure, Confirmation page for successful & failed transactions.
- Admin interface - manage transactions, view reports, view failed & suspicious logs.
- Bank/payment gateway interface - Seamless integration with external banking API's

5. Performance requirements

- Should process each transaction within 2-3 sec.
- handle 500 concurrent transactions
- Availability: 99.9% uptime
- Response time ≥ 1 second.

6. Design constraints

- must comply PCI DSS (Payment Card Industry Data Security)
- use HTTPS for data transfer
- Encryption for sensitive data like Card No. & CVV.
- Compatible on both web & App.
- Integrate with authorized payment gateways

7. non-functional requirements

- Security - protect data using encryption, user authentication & fraud detection
- Reliability - Ensure accurate & consistent performance
- Scalability - Support future growth in no. of users
- Usability - user-friendly UI.
- Maintainability - Easy update & debugging.
- Compliance - follow all legal regulations.

Budget (in Rupee)

Planning & Design	- ₹ 80,000
Development	₹ 1,00,000
Database & Hosting	60,000
Security	50,000
Testing & QA	40,000
Maintenance	60,000

Schedule (in Weeks)

Planning & Design	2
Development	4
Integration & Testing	3
Deployment	1
Maintenance	Ongoing

A

4. Stock Maintenance System

Problem Statement:

A stock maintenance system is needed to efficiently manage, track, and update inventory levels ensuring stocks.

SRS documenting

1. Purpose of the document

This document defines the objectives, scope and requirements of the stock maintenance system to ensure efficient and accurate inventory management.

Scope

The stock maintenance system will manage and track inventory levels, update stock records, generate reports, and ensure accurate and efficient handling of stock data for smooth business operations.

Overview

This is designed to simplify inventory mgmt by tracking stock levels, updating records, and generating reports. It ensures accuracy, reduce manual errors, and improve overall efficiency in managing stocks.

2. General Description

The stock maintenance system help manage inventory by recording, updating and tracking stock details. It ensures accurate data, reduces manual effort, and streamlines the overall stock management process.

3. Functional requirements

- features - add, update and delete stock details
- Track current inventory levels.
- Generate stock reports
- Set alerts for low stock levels

4. Interface requirements

- Simple, user-friendly dashboard
- Secure login for admin & user
- Clear displaying stock details & reports

5. Performance requirements

- Handles 100+ concurrent users smoothly
- Process stock updates within 2 seconds
- Ensures 99% system uptime

6. Design Constraints

- Use secure authentication and encryption
- Web and mobile compatibility.

→ must integrate with existing databases

Non-functional requirements

- Security - Protect stock data
- Reliability - Accurate & Consistent update
- Scalability - Support future stock expansion
- Usability - Easy for non-technical users

Budget (in Rupee) : 3 Lakhs

Planning & Design	40,000
Development	1,50,000
Database & Hosting	40,000
Testing & Security	30,000
maintenance	40,000

Schedule (in weeks)

Planning	1 week
Development	4 weeks
Testing	2 weeks
Deployment	1 week
maintenance	ongoing

A

5. PassPort Automation System

Problem Statement

It aims to simplify & speed up the passport application process.

SRS Documents

①.

Purpose of this document

The purpose of this document is to define the objectives, scope, and requirements of the PassPort Automation System to streamline the application, verification, and approval process efficiently and securely.

Scope

The System will allow users to apply for passports online, schedule appointments, track application status, and make secure payments. It will also enable authorities to verify documents, manage records and approve or reject applications efficiently.

Overview

This provides a centralized platform for managing passport application and related services. It ensures fast processing, accurate data handling, secure transactions and

easy access for both applicants & officials.

2. General Description

This system digitizes the entire passport application workflow - from form submission and document verification to payment & approval. It minimizes paperwork, reduces delays, and enhances user experience while ensuring security and transparency.

3. Functional requirements

- User registration & login
- Online passport application submission
- Uploading & verifying documents
- Online payment gateway integration
- Application tracking & status updates
- Admin panel for verification & approval

4. Interface requirements

- User interface - Clear, responsive portal for applicants
- Admin interface - Dashboard for managing and verifying applications
- External interface: Integration with payment gateways & government databases.

5. Performance requirements

- Process applications within 3-5 sec per request

- Handle 1,000+ concurrent user efficiently
- maintain 99.9% uptime for uninterrupted service.

Design Constraints

- Must comply with government data security policies
- Use HTTPS & strong encryption for sensitive data.
- Compatible with both desktop & mobile platforms

Non-functional requirements

- Security - Protect sensitive user data with encryption
- Reliability - Ensure consistent & accurate application processing.
- Scalability - Support increasing no. of users & applications
- Usability - provide an intuitive & simple interface
- Compliance - follow legal standards & govt norms.

Budget (In rupee)

Planning & Design	70,000
Development	2,00,000
Database & Hosting	60,000
Security & Compliance	80,000
Testing & Q.A	50,000
maintenance	70,000

Project Schedule (In Weeks)

Requirement Analysis	1
Design & Planning	2
Development	5
Integration & Testing	3
Deployment	1
maintenance	Ongoing.