Teacher Sign / Remarks

# SRS DOCUMENT

1. Hotel Management system

Problem stmt: To design & implement HMS. Managing hotel operatn's manually is time consuming, inefficient, error prone. Tasks such as room booking, check-in, check-out, billing, staff scheduling & reporting are often done on paper (or) with basic tools, which can lead to double booking, billing mistakes, loss of records & poor customer service. There is a need for a computerized Hotel Management system that can handle all these tasks in one place.

SRS.

1. Introducn

1.1. Purpose of the Document

The purpose of this document is to define the functional & non-functnal requirements for the Hotel Management system (HMS). The system should make it easier to manage reservatns, track room availability, generate accurate bills, manage staff records, & provide timely reports, while also improving customer experience & reducing workload for hotel staff.

1.2 Scope of the Document

The HMS is a web/desktop-based applicatn that enables hotels to manage:

- Customer rsrvatns & cancellatns
- Room allocatn & status tracking
- Billing & invoice generatn
- Staff duty schedules & payroll
- Restaurant/food ordering & integratn with room service
- Generating daily, weekly & monthly business reports

The system will:

- Reduce manual errors in booking & billing.
- Improve customer experience through faster service
- Enable hotel administrators to monitor performance in real-time
- Support both small hotels & large chains with scalability.

1.3 Overview.

The system will consist of 3 major modules.

1. Customer Module - Handles bookings, cancellations & payments
2. Staff Module - Manages employee records, schedules & tasks
3. Administration Module - Manages hotel operations, financial reporting & analytics

The system provides a secure login for diff user roles (Admin, Receptionist, Customer & Manager).

(2) General Description

- Users of the system
  - Hotel administrators/managers.
  - Receptionists
  - Customers
  - Support staff (housekeeping, restaurant etc.)
- Assumptions & Dependencies.
  - The system requires internet (or) LAN connectivity for real-time updates
  - Database must support multi-user access
  - Payment gateway integration is assumed for online payment

(3) Functional Requirements.

1. User Authentication
   - Admin, staff & customer must login securely
   - Diff roles should have diff access rights.
2. Room Booking & Cancellation.
   - Customer can check room availability
   - Receptionist can allocate rooms manually.
   - System should automatically update availability after booking/cancellation.
3. Check-in / Check out Process.
   - Record customer details at check-in
   - Update room status upon check-out.
4. Billing & Payments.
   - Generate automated invoices for customers.
   - Support multiple payment modes (cash, card, online)
   - Maintain payment history.
5. Staff Management:
   - Store staff details (personal, employment, payroll).
   - Manage work shifts & leave requests
6. Restaurant / Food service
   - Allow room service orders to be linked to the customer's bill.
   - Maintain menu & update stock levels.
7. Reporting:
   - Generate occupancy reports, revenue reports & staff performance reports
   - Support export to PDF/Excel

(4) Interface Requirements.

- User Interface
  - Responsive web UI (accessible on desktop, tablet, mobile)
  - Simple navigation with role-based dashboards.
  - Graphical reports with charts.
- Hardware Interface.
  - Works on hotel computers with min processor: Intel i5, 8GB RAM
    Storage: 500 GB HDD.
  - Supports barcode / QR scanners for check-in/out.
- Software Interface:
  - OS: Windows / Linux.
  - Database: MySQL / PostgreSQL.
  - Web Framework: Django / Java Spring / Node.js
  - Payment Gateway: Razorpay / PayPal / Stripe

(5) Performance Requirements

- The system should handle at least 200 concurrent us
- Response time for room availability check should be < 2 seconds.
- Payment transactn should process within 5s.
- Database should support up to 1 million records without performance degradatn.

(6) Design constraints

- System must comply with GDPR/Indian IT Act for data privacy.
- Only authorised users can access sensitive data.
- The UI should follow accessibility standards (WCAG)
- System should integrate with third-party APIs (payment, email, SMS).

(7) Non-functional Attributes

1. Security :- • Data encrypt using AES-256 for customer data
   • Secure HTTPS communicatn
   • Role-based access control
2. Reliability: • 99.5% uptime requirement
   • Daily automated database backup.
3. Scalability :- • Ability to expand from single-hotel to multi-chain management
4. Maintainability - Modular design for easy up.dates
   • Documentatn for all modules.
5. Usability :- • Easy-to-use interface for non-techn hotel staff.
   • Multilanguage support

(8) Total estimated budget for HMS is ₹15,00,000.

| Phase | Task | Duratn. |
|---|---|---|
| Phase 1 | R.A | 2 weeks |
| P-2 | System Design | 3 weeks |
| P-3 | Development | 8 weeks |
| P-4 | Testing & QA | 4 weeks |
| P-5 | Deployment & training | 2 week |

20/8/25

Credit card

Problem stmt :- To Design and Implement a Credit Card Processing system. Managing credit card manually (or) with outdated systems creates risks such as delays, error, fraud & poor customer service. With the increasing volume of digital payments, banks & financial institutions need a secure, automated Credit card Processing system that can handle card applicatn, approvals, transactns, billing, fraud detectn, & customer account management efficiently

Introductn

1.1 Purpose of the Document

The purpose of this document is to define the functional & non-functional requirements of the Credit card Processing system (CCPS). The system will manage the complete credit card lifecycle including issuance, transactions, billing, fraud detectn, & reporting. This document will guide developers, testers, and stakeholders in ensuring the system meets business needs.

1.2 Scope of the Document

(*) Allow customers to apply for credit cards online.
(*) Process transactns (purchases, cash withdrawals, online payments)
(*) Generate monthly stmts & bills.
(*) Provide fraud detectn & alert mech.
(*) Enable administrators to manage credit limits, block/unblock cards, & generate reports
- Support integratn with banks, payments gateways & mobile apps.

The system will ensure
- Security of sensitive financial data
- Reliability in transactn processing
- Scalability to handle millions of transactns daily

1.3 Overview.

1. Customer Module - Apply for cards, view stmts, pay bills.
2. Transact^n Module - Handles real-time transact^n authorizat^n
3. Admin Module - Card Management, fraud detect^n & reporting.

2. General Descript^n.

User of the system
- Customer (Card Holders)
- Bank administrators.
- Credit card processing staff
- fraud detect^n / security officers.

Assumpt^s & Dependencies.

System requires integrat^n with core banking & payment gatew
Transact^ns must comply with PCI DSS security standards
Internet & mobile connectivity required for customer acces.

3. Functional Requirements

1. User Authenticat^n & Authorizat^n.
   - Customers log in securely to their account
   - Admins have role-based access.
2. Credit Card Applicat^n & Approval
   - Customers can apply online
   - System checks eligibility & processes applicat^n.
3. Transact^n Processing
   - Authorize & process card swipes, online payments & withdrawals
   - Update customer balance in real time.
4. Billing & Stmts
   - Generate monthly stmts
   - Calculate interest, fees & rewards points
   - Provide online bill payment facility
5. fraud Detect^n.
   - Monitor unusual spending patterns
   - Trigger alerts & temporarily block suspicious a/c.
6. Admin Controls
   - Approve / deny credit card applicat^ns
   - Adjust credit limits.
   - Block / unblock cards.
7. Report & Analytics
   - Transact^n reports, customer spending reports, fraud alerts
   - Export to PDF / EXCEL.

(4) Interface Req.

- User Interface.
  - Web & mobile-friendly portal for customers
  - Admin dashboard with analytics & fraud alerts
- Hardware Interface
  - ATMs, POS (Point of sale) machines & servers.
- Software Interface
  - Integrate with banking core system
  - Payment gateways (Visa, MasterCard, RuPay)
  - SMS / EMail services for alerts.

(5) Performance req.

- Should handle at least 10,000 transact^ns per sec
- Transact^n approval response time should be < 2s
- System availability 24/7 with 99.9% uptime

(6) Design Constraints

- Must comply with PCI DSS & RBI guidelines
- Encrypt^n (AES-256) for transact^n data
- Limited access to sensitive customer data

7 Non functional Requirements

1. Security - 2 factor Authentication (OTP, biometrics)
2. Reliability - Backup servers & disaster recovery plan
3. Scalability - cloud support for high transactn volume
4. Maintainibility: Modular architecture for easy updates
5. Usability: simple & intuitive UI for customers.

8 Preliminary schedule & Budget

| | | | Budget |
|---|---|---|---|
| Phase 1 | Req Analysis | 3 weeks | ₹4,00,000 |
| P2 | System Design | 4 weeks | ₹15,000 |
| P3 | Development | 12 weeks | ₹3,00,000 |
| P4 | Testing & QA | 5 weeks | ₹2,000 |
| P5 | Deployment & training | 3 weeks | ₹6,000 |
| Total Duratn | | ~27 weeks (~7 months) | ₹30,00,000 (~$36,000) |

## Library Management System (LMS)

To design & implement a LMS that automates & streamlines the management of books, members & transactns such as issue, return & reservatn.

Manual library operatns often lead to problems such as misplaced records, difficulty in tracking issued books, delays in searching, & errors in calculating fines. With an ↑ no of users & resources there is a need for a secure & automated system that provides efficient book cataloging, easy search & retrival, member management, fine calculatn & reportings.

The system aims to improve efficiency, reduce errors & enhance user satisfactn for both librarians & library members

1. Introductn.

1.1 Purpose of the Document.

The purpose of this document is to define the functional & non-functional requirements of LMS. The system will manage library resources by automating processes like book cataloging, borrowing, returning, reservatns, & fine calculatn. It will serve as a reference for developers, testers & stakeholders to ensure the project's success

1.2 Scope of the Document

The LMS will

Manage books, journals & digital resources.
Maintain member details & borrowing history.
Allow book issue, return & reservatn
Calculate overdue fines automatically.
Provide search functionality for books & members
Generate reports for librarians.

The system will ensure

Faster processing of library transactns
Reduced manual errors
Easy access to library resources for users

1.3 Overview

The system will have 3 main modules

1. Admin / librarian Module - Manage books, members & transactns
2. Member Module - Search books, request, borrow, & return
3. Report Module - Generate statistics about usage, fines & inventory

General Descripts

- Users of the system
  - Librarians (Admins)
  - Library Members (student, Teachers, Researchers)

Assumptns & Dependencies

Database required to store books, members & transactn records

Internet / LAN connectivity for multi-user access

Barcode / QR scanners can be used for faster processing

Functional Req

1. User Authenticatn
   - Librarians & members log in securely
2. Book Management
   Add, update / delete book records
   Categorize books by subject, author & availability
3. Member Management
   Register new members
   Maintain borrowing history
4. Book issue & return
   Issue books to members & record the due date
   Update system on return & calculate overdue fines
5. Search & Reservatn
   Members can search for books by title, author, ISBN or category
   Reservatn of unavailable books
6. Fine Calculatn
   Automatic fine calculatn based on overdue days
7. Reporting
   Reports for issued / returned books
   Fines collected & overdue books
   Member activity reports

4 Interface Requirements

- User Interface
  User friendly web & desktop UI
  Search box, navigatn menus & role-based dashboards
- Hardware Interface
  Works on standard PC's with min Intel i3, 4GB RAM
  Barcode / QR code scanner support
- Software Interface
  Database: MySQL / PostgreSQL
  OS: Windows / Linux
  Languages: Java / Python / PHP with web framework

5. Performance req
   System should handle 500 concurrent users.
   Book search result should load in < 2 s
   Database should support upto 1 million records

6. Design constraints
   Must comply with data protectn regulatns
   Only librarians can modify book & member data
   Maintain backup of records regularly

7 Non functional attributes

Security: Role based access control, secure login

Reliability: Daily database backup, 99.5% uptime

Scalability: support for growing no of members & resources.

Usability: simple & intuitive design for non-technical users

Maintainability: Modular structure for easy updates

Preliminary Schedule & Budget

| | | | |
|---|---|---|---|
| PHASE 1 | Req Analysis | 2 weeks | ₹ 1,50,000 |
| P2 | System Design | 3 weeks | ₹ 6,00,000 |
| P3 | Development | 8 weeks | ₹ 1,00,000 |
| P4 | Testing & UA | 3 weeks | ₹ 75,000 |
| P5 | Deployment & training | 2 weeks | ₹ 1,75,000 |
| | Total | 18 weeks | ₹ 10,50,000 |

Stock maintance system

To Design & Impriment a Stock Maintenance System that automates the tracking, updating & management of stock levels for businesses. Manual stock maintanance often leads to errors, delays, theft & mismanagement, causing losses & inefficiency. An automated system is needed to record stock entries, monitor inventory in real-time, generate alerts for low stock, maintain supplier & transactn details & provide analytical reports, ensuring smooth & reliable stock control.

SRS

1. Introductn

1.1 Purpose

The purpose of this document is to specify the requirements for the development of a Stock Maintan System (SMS). This system will help businesses track inventory, manage stock levels, generate reports & minimize human errors in stock handling.

1.2 Scope

The stock Maintanance system will:

Maintain details of items, suppliers & transactns

Record purchases & sales automatically.

Provide real-time stock updates & low stock alerts.

Generate inventory & sales reports for analysis

Ensure secure login for Admin & staff users.

1.3 Overview.

The system consists of:

Admin Module: Manage stock, supplier, reports & users

Staff Module: Record stock-in, stock-out & sales

Database: Stores all stock, supplier & transactn details

Reports Module: Provides daily, monthly & annual summaries

(2) General Descriptn

Users: Admin, Staff

System Environment: Web/Desktop application with database support

Constraints

Only authorized users can access

Should be scalable for diff business sizes

Assumptns:

Users have basic computer knowledge

All transactns are genuine & verified

(3) functn Requirements

1. User Management

Admin can create, update (or) delete user accounts

Login authenticatn for Admin & Staff

2. Stock Management

. Add new items with details (ID, name, category, price, quantity, supplier).

. Update stock levels when purchase / sales occur

. Delete discontinued items

3 Supplier Management

. Maintain supplier informatn (name, contact, add)

. Link suppliers to purchased stock

Transact^n Management
Record stock purchases.
Record sales / stock out details.
Generate Invoices.

Reports & Alerts
Generate daily/weekly/monthly stock reports
Low stock alerts when quantity falls below thresh
Sales vs purchase analysis.

(4) Interface Req
User Interface
Simple GUI with menu option for stock, Suppliers, Transact^ns & Reports

Hardware Interface
Standard computer with 4GB RAM 500GB HDD

Software Interface
OS: Windows/Linux
Database: MySQL / PostgreSQL
Lang: Java / Python / .NET (Depending on implementat^n)

(5) Performance Req
. The system must support atleast 1000 stock items without performance lag
Response time for stock search should be < 2s
The system should handle simultaneous access by multiple users

(6) Security req.
Password-protected login system
Role-Based access (Admin has full control, Staff has limited access)
Database backup & restore facility

7. Reliability: System should have uptime of 99%
Scalability: Should support more users / items when needed
Usability: Easy to use GUI for non-tech staff
Maintainability: Should allow easy updates & debugging

Total estimated budget is ₹15,00,000.

Passport automat^n system -
To design & implement a Passport Automat^n System that simplifies & automates the process of passport applicat^n, renewal & tracking. Currently passport processing is largely manual, involving long queues, paperwork & inefficiencies that lead to delays, errors, fraud, & poor customer service. An automated system is needed to allow applicants to apply online, upload documents, schedule appointments, track applicat^n status & receive notificat^ns.

SRS.

1 Introduct^n.

1.1 Purpose
The purpose of this SRS is to define the requirements for the Passport Automat^n System. The system will allow applicants to submit passport applicat^ns online, reduce paperwork, provide transparency & assist passport office staff in processing & verifying applicat^ns efficiently.

1.2 Scope
The Passport Automat^n system will support online applicat^n & renewal of passports.
Allow applicat^ns to upload documents digitally
Provide appointment scheduling for biometric & police verificat^n.
Enable real time applicat^n tracking & status notificat^

Overview

User Module: Applicat^n submission, payment, status tracking.

Admin Module: Document verificat^n, police verificat^n, approval

Database Module: stores applicant details, applicat^n history, & issued passports

Notificat^n Module: sends updates via SMS/Email

(2) General Descript^n.

Users: Applicants, passport officials, police verificat^n officers, Admins

System Environment: Web-based system accessible via browser & mobile devices

(3) Functional req

User Registrat^n & log(o)

Users can register using personal details

Secure login with username/password/OTP

Passport Applicat^n

All online applicat^n form

Upload req documents

Pay applicat^n fee online

Verificat^n Process

Officials verify documents

Police verificat^n integrated

Appointment Scheduling

System auto-generates available slots

User can reschedule appointments

Status tracking & Notificat^ns

Applicants can track applicat^n progress

Automatic notificat^ns via SMS/email

(4) Interface req

· User Interface

· Web & mobile-friendly UI

· Forms for applicat^n i/p

· Dashboard for users & admins.

Hardware Interface

· Biometric devices

· Secure servers for database storage

Software Interface.

Payment gateway

SMS/email server provider

National ID database

(5) Performance Req.

System should support 10,000+ concurrent users.

Response time must be less than 2 s for user actions

Database should handle millions of records efficiently

(6) Design Constraints

Must comply with gov security & privacy policies

Must follow standard encryp^n protocols for data transmission

Support multi-lang functionality

Operate on standard web browsers & mobile devices

(7) Non-Functional Attributes

Security: End to end encryp^n, role based access

Reliability: Data backup & recovery mechanisms.

Maintainability: Modular design for updates

Scalability: Cloud support for handling ↑ users

Usability: Simple & user friendly UI for citizens

(8) Preliminary Schedule & Budget.

1. RA - 1 Month
2. System design - 1 Month
3. Implementation & Module Dev - 2 Months
4. Testing & Debugging - 1 Month
5. Dev & training - 1 Month

Budget Estimate

Software dev - $50,000.

Hardware & servers - $30,000 -

Security & compliance - $20,000

Maintance (1st yr) - $15,000 -

Total Estimated Budget :- $115,000

28/8