SRS Document.

Hotel Management System.

1. Introduction

1.1. Purpose: To specify requirements for "HotelSphere" Hotel Management System.

1.2. Document Conventions
→ Following the IEEE SRS standard.

1.3. Audience
→ Intended for project team (managers, developers) and stakeholders (hotel staff) to understand the project's scope and requirements.

1.4. Product Scope
→ Web-based system for managing reservations, front desk operations, room service, and billing.

1.5. References
→ IEEE std
→ Client requirements and project

2. Overall Description

2.1. Product Perspective.
→ A standalone system that integrates with third-party APIs from Online Travel Agencies (OTAs) and payment gateways.

2.2. Product functions
→ Manages reservations, check-in/check-out, room status, billing and reporting.

2.3. User Classes and characteristics
→ Front Desk Staff : Manages daily operations; needs a simple interface.
→ Housekeeping staff: Updates room status; needs a mobile friendly interface.

→ Hotel Manager: Oversees all operations; needs comprehensive reports.

→ Guests: Views bookings and invoices; needs a secure, easy-to-use interface.

2.4. Operating Environment.

→ Client - server environment

→ Server runs on Windows and a database.

→ Client - web-browser.

2.5 Design and Implementation Constraints.

→ Must comply with hotel data privacy standards

→ Limited to the capacity of hotel hardware infrastructure.

2.6. User Documentation.

→ User manuals for staff and admin

→ Online help/ documentation.

2.7. Assumptions and Dependencies

→ Reliable internet connection (if online booking)

→ Availability of database server.

3 Requirements.

3.1. User Interface

→ Simple, intuitive UI with menus for bookings, check-in, billing and reports.

3.2. Hardware Interfaces

→ PC or server with database connectivity.

→ Printers for receipts and bills.

3.3. Software Interfaces

→ Database: MySQL

→ Payment gateway integration.

4 System Features

→ Room Booking

→ Customer management

→ Billing and payments

→ Staff management

5 Other Non functional Requirements

5.1. Performance Requirements

→ Support up to 100 concurrent users

→ Booking transactions should complete within 2 seconds

5.2. Safety Requirements

→ Regular Data backups

→ Fail-safe mechanisms in case of server crash.

5.3. Security Requirements

→ Role-based access control

→ Encrypted storage of sensitive data

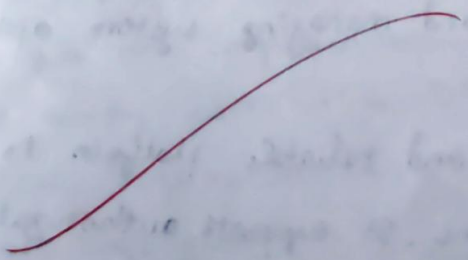
5.4 Software Quality Attributes

→ Usability: Easy-to-use interface

6 Other requirements

→ Compliance with local hotel laws and taxation policies

→ Multi-language support for international customers.



II. Credit Card Processing System

1. Introduction.

1.1. Purpose

The Credit card Processing System (CCPS) is designed to securely authorize, authenticate, and settle credit card transactions for merchants and customers. It ensures accuracy, fraud prevention, and compliance with financial regulations

1.2 Document Conventions

- CCPS = Credit Card Processing System
- POS = Point of Sale
- OTP = One-Time Password

1.3 Intended Audience and Reading Suggestions

- Merchants: For accepting card payments
- Banks/Payment Gateways: For validating and settling transactions
- Developers / Testers : For building and validating the system.
- Admins: For monitoring and managing system operations.

1.4 Product Scope

The CCPS provides a secure and reliable platform to handle card transactions in real time. It supports authorization, authentication, fraud detection, billing, and settlement between banks and merchants.

1.5. References

- PCI DSS Security Standards
- ISO 8583 Financial Messaging Standard.

2. Overall Description

2.1 Product Perspective

- The CCPS works as middleware between merchants, customer banks and payment gateways. It integrates with POS devices, ATMs and online payment portals.

2.2 Product Functions

- Transaction authorization and validation.
- Customer authentication (PIN/OTP)
- Settlement between ~~fraud~~ bank and merchant.
- Reporting and fraud detection.

2.3 User Classes and Characteristics

- Customer: Initiates payment with card details
- Merchant: Accepts payments via POS/online.
- Admin/Bank Staff: Monitors, verifies, and manages transactions

2.4 Operating Environment

- Secure servers with 24/7 availability
- POS terminals, ATMs and web applications
- Encrypted communication channels.

2.5 Design and implementation Constraints.

- Must comply with PCI DSS and local regulations.

2.6 User Documentation

- Merchant user guide
- Admin manuals
- API documentation for integration.

2.7 Assumptions and Dependencies

- Continuous internet connectivity
- Support from banks and payment networks

3 External Interface Requirements

3.1 User interface

- POS interface for merchants
- Customer card input + PIN/OTP screen.

- Admin dashboard for monitoring.

3.2 Hardware Interfaces

- Card readers, POS machines, ATMs

3.3 Software Interfaces

- Bank APIs
- Payment gateway integration
- Fraud detection modules.

3.4 Communications Interfaces

- Secure encrypted Internet channels
- Standard ISO 8583 message format.

4 System Features

↔ Authorisation - validates card details and account balance.

→ Authentication - PIN or OTP

→ Reporting.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

- Handle upto 5000 transactions per second
- Process each transaction in less than 3 seconds.

5.2 Safety requirements

- Backup and recovery mechanisms
- Rollback for incomplete transactions

5.3 Security requirements

- Strong encryption (AES, RSA)
- Role-based access control.
- PCI DSS compliance.

14/8/2025

6 Other Requirements

- Support for Visa, MasterCard, RuPay, and other networks
- Multi-currency support for global usage.

Library Management System

1. Introduction

1.1 Purpose

→ To define the requirements for a library Management System that automates book tracking, user management, and lending operations.

1.2 Document Conventions

→ All dates follow the DD-MM-YYYY format.

1.3. Intended audience and Reading Suggestions

→ This document is intended for developers, testers, librarians and stakeholders.

1.4 Product Scope

→ The system will manage book inventory, member records, issue or return processes, and overdue tracking. It will support both staff and member interactions.

1.5 References

- IEEE SRS Standard 830 - 1998
- Library Science Best Practices
- PostgreSQL Documentation

2. Overall Description

2.1 Product Perspective

- This is a standalone web-based application replacing manual library operations.

2.2 Product Functions

- Add/update/delete books and members
- Issue and return books
- Search catalog
- Generate reports.

2.3 User Classes and Characteristics

- Librarians: Admin Privileges

- Members : Limited access to search and request books
- Admins : Full system control.

2.4. Operating Environment

- Web browser
- Server : Linux-based with Postgre SQL
- Client : Any devices with internet access.

2.5 Design and Implementation Constraints

- Must use open-source technologies.
- Responsive UI for mobile and desktop.
- Data encryption for user credentials

2.6 User Documentation

- Users manual (PDF)
- Online help section
- Admin training guide

2.7 Assumptions and Dependencies

- Users have basic digital literacy
- Internet connectivity is available

3. External Interface Requirements

3.1 User Interfaces

- Login or registration page
- Dashboard for librarians and members
- Search and book request interface

3.2 Hardware Interfaces

- A good server.

3.3. Software interfaces

- PostgreSQL database
- RESTful APIs for integration
- Authentication via OAuth 2.0

4. System features

- Book Management - categorise books by genre, author.
  - add or delete book records (edit too)
- Member management - register and manage member profiles
- Lending Operations - issue or return books
  - Borrowing history
  - Calculate fines for overdue items
- Search and Reports - search books by title, author or ISBN
  - Generate monthly usage reports.

5 Other Nonfunctional Requirements

5.1. Performance

- System should support 100 concurrent users
- Search results must load within 2 seconds

5.2 Safety Requirements

- Regular data Backups
- Graceful error handling

5.3 Security requirements

- Role-based access control
- Passwords stored using hashing

6. Other Requirements

- Integration with SMS gateway
- Support for multilingual UI.
- Audit logs for admin actions

IV Stock Maintenance System

1. Introduction

1.1 Purpose

To define the requirements for a Stock Maintenance System that enables efficient tracking, updating and efficient tracking, updating, and reporting of inventory across multiple categories and locations

1.2 Document Conventions

- Currency: INR
- Date DD-MM-YYYY format.

1.3 Audience

- This document is intended for developers, inventory managers, business stakeholders, and QA teams

1.4 Scope

- The system will manage stock levels, record transactions, generate alerts for low inventory and support reporting for decision-making. It will streamline operations for warehouses, retail outlets and procurement teams.

1.5 References

- Inventory management
- ISO/IEC 25010 Software Quality Model.

2. Overall Description

2.1 Product Perspective

This is a centralized web-based application replacing manual or spreadsheet-based stock trading and tracking

2.2 Product Functions

- Add/update/delete stock items
- Monitor stock levels and movement.
- Generate alerts and reports
- Support multi-location inventory

2.3 User classes and characteristics
- Admin: full access to all modules
- Inventory Manager: Manage stock and view reports
- Staff: Limited access to update stock entries

2.4 Operating environment
- Web browsers (Chrome, edge)
- Server: Linux with mySQL
- Client: Desktop or mobile devices

2.5 Design and implementation Constraints
- Must support role-based access
- Use open-source stack (Node.js etc)
- Responsive design for mobile compatibility.

2.6 Assumptions and Dependencies
- Users have basic inventory knowledge
- Stable internet connection.
- MySQL and Node.js installed on server.

3. External Interface Requirements

3.1 User interfaces
- Login and Dashboard
- Stock entry and update forms
- Alerts and reporting module.

3.2 Hardware Interfaces
- Printer for stock reports

3.3 Software interfaces
- MySQL database
- Authentication via JWT

4 System features
→ Stock Management - add/edit/delete stock items
→ Categorize items
→ Inventory tracking

→ Alerts and Notifications
→ Reporting

5 Nonfunction Requirements

5.1 Performance Requirements

- Support 200 concurrent users
- Stock updates reflected within 5 seconds

5.2 ~~Safety~~ Security Requirements

- Role-based access control
- Encrypted credentials and secure sessions

5.3 Safety requirements

- Daily backups
- Rollback mechanisms for erroneous entries

6 Other Requirements

- Support multi language UI
- Audit logs for stock changes
- Integration with ERP systems

Passport Automation System

1. Introduction

1.1 Purpose

To define the requirements for a Passport Automation system that streamlines the application, verification and issuance of passports through a secure and user-friendly digital platform.

1.2 Document Conventions

- Currency INR.

1.3 Audience

- This document is intended for government officials, developers, testers, and system integrators.

1.4 Scope

- The system will automate passport-related services including application submission, document verification, appointment scheduling and status tracking. It will reduce manual processing and improve transparency.

1.5 References

- Government of India Passport Guidelines
- Aadhar API documentation.

2. Overall Description

2.1 Product Perspective

- A web-based application with national identity databases and police verification systems.

2.2 Product functions

- Online passport application
- Document upload and verification
- Appointment booking
- Status tracking and notifications

2.3 User Classes and Characteristics

- Applicants: Citizens applying for passports
- Officials: Verify documents and approve applications
- Admins: Manage system setting and user roles

2.4 Operating Environment

- Web browsers
- Desktop or mobile devices

2.5 Design and Implementation Constraints.

- Must comply with government security standards
- Aadhar and PAN integration required
- Multilingual support.

2.6 User Documentation

- Online help center.

2.7 Assumptions and Dependencies

- Applicants have valid identity documents
- Available internet access

3. External Interface Requirements

3.1 User interfaces

- Application from interface
- Document upload portal
- Appointment calendar
- Status dashboard.

3.2 Hardware interfaces.

- Printers for application receipts

3.3 Software interfaces

- Aadhar and PAN APIs
- Police verification system
- Payment gateway

System features

- Stock Management - add/edit/delete stock items
- Inventory Tracking - record of incoming and outgoing stock.
- Alerts and Notifications
- Reporting

5 Other Non-functional

→ Performance requirements.
- High availability with minimal downtime.
- Fast processing.

→ Usability.

6 Design Constraints
- Compliance with national security & data privacy regulations

2/9/2025