Date __/__/__
Page _____

OOMD Lab

SRS

Hotel management system

Problem statement: Managing hotel operations manually or using disparate systems often lead to inefficiencies, errors and poor customer service. Hotel staff struggle with tasks such as room booking, check-in/check-out billing, etc. resulting in delays and inaccurate information. There is a need for an integrated, automated Hotel management system that streamlines these operations, reduces manual effort, improves accuracy and provides real-time data access to hotel management.

Introduction

- Purpose: The main objective of the Hotel Management SRS is to provide a base for the foundation of the project. It gives a comprehensive view of how the system is supposed to work and what is expected by the end users. The client's expectations and needs are analyzed to define clear, precise functional and non-functional requirements. This is done to ensure that the end users' needs are met.

This SRS also serves as a reference for future developers and maintainance teams.

→ • Scope: Hotel management system automates hotel operations, focusing on online room reservations. It supports 3 user types: customers, Receptionists, and hotel managers. Customers can check

Date __/__/__
Page _____

room availability, book and pay online. Receptionists manage and update bookings, while managers oversee financial reports and update room details like cost and category.

The system includes a Booking Management module, a database server, and a report generator. It's main goal is to streamline daily hotel processes, improve customer service, and replace cumbersome physical record-keeping.

→ Overview: The remaining sections cover the overall description of the system, including product perspective, user characteristics, assumptions, and constraints. Detailed description of functional and non-functional needs, external interfaces, system attributes, performance, etc are given.

→ General Description: HMS is a comprehensive software solution designed to streamline and automate the day to day operations of a hotel. It facilitates efficient management of room reservations, check-ins/check-outs, housekeeping etc. The system aims to improve the overall guest experiences by enabling quick access to room availability, booking details and service requests.

→ Functional requirements

Date __/__/__
Page _____

- users (customers, receptionists, managers) can securely register and login.
- customers can search and book available rooms online.
- The system updates room availability in real time.
- Customers can make online payments and receive confirmations.
- Receptionists can manage bookings and handle check-in/check-out.
- Access is controlled based on user roles.

→ Interface Requirements

- web-based, user-friendly interface for all user roles.
- Integration with third-party payment gateways.
- Connection to email/SMS services for notifications.
- Export reports in PDF and Excel formats.

→ Performance requirements

- The system shall handle upto 500 concurrent users without performance degradation.
- Room availability search results shall load within 3 seconds.
- The system shall have 99.9% uptime availability.
- Data backup and recovery shall occur every 24 hours.

→ Design constraints

- The system must be developed as a web-based application.
- It must be compatible with modern browsers.
- The backend must use a relational database

Date / /
Page

Non functional attributes:

- Reliability: The system shall operate continously with 99.9% uptime
- usability: The interface shall be intuitive and easy to navigate for all user types
- security: data shall be encrypted, and users must authenticate to access the system.
- portability: The system shall be accessible on different devices and major web browsers

→ preliminary schedule & budget

- week 1: Requirement, Analysis
- week 2-3: system design
- week 4-6: Development
- week 7-8: Testing and deployment
- week 9: Documentation and Training

Total Duration ~ 9 weeks

Budget

- Requirement and design: $1,000
- Development: $4,000
- Testing and deployment $2000
- Documentation: $500
- Total estimated Budget $7,500

19/8/25

Date / /
Page

Credit card processing SRS

Problem statement: Managing credit card is challanging. Many biusnesses face challenges in securely and efficeatly processing credit card transactions due to complex payment workflows, fraud risks, etc. Manual or outdated systems lead to transaction delays, errors & increased risk of fraud, resulting in poor customer experience and potential revenue loss. A reliable, fast and secure credit card processing system is needed to streamline payments, ensure compliance and protect sensitive data.

1. Introduction

The purpose of the document is to describe the requirements for the credit card processing system (CCPS). It defines the system's functionalities, constraints, performance expectations, overall architectures. This SRS will serve as a reference for developers, project managers, testers, & to better understand the system.

Scope:

The CCPS will provide a secure platform to authorize process and settle credit transactions between merchants, customers, banks. The system ensures transaction integrity, fraud detection, compliance with industry regulations and reporting capabilities.

General description

→ The system will process credit card transactions in real-time.

→ It will interact with customers, merchants, issuing banks and acquiring banks.

→ The system will be modular, with components for authorization, fraud detection, settlement, reporting administration.

→ The system must comply with PSI DSS and ISO 8583 messaging standards.

→ It will be deployed on secure servers with high availability and failure mechanisms.

→ Functional requirements

1. Transaction authorization
   Input: card details, CVV, expiry date, amount
   Process: validate details, communicate with issuer bank
   Output: Approval / decline
2. Fraud detection
   Monitor transaction patterns
   Block suspicious or high risk activity
3. Settlement and clearing
   Handle daily settlements between banks
   Generate financial reports
4. Refund and chargebacks
   Allow merchants to initiate refunds
   Handle customer disputes
5. Reporting
   Provide transactions history, analytics & fraud reports

Interface requirements.

→ User interfaces → web portal for merchant and administrator

→ customer interactions through merchant POS or online stores

→ software interfaces. Bank APIs for authorization and settlement

→ fraud detection system integration

→ Hardware interfaces → POS devices for card swiping

→ secure backend servers.

→ Communication interfaces: All communication must be encrypted using SSL/TLS

→ Performance requirements:

→ System must process transactions within 2-5 seconds.

→ must support atleast 1000 concurrent users

→ uptime requirement: 99.9% availability

→ Ability to scale upto 1 million transactions per day.

→ Design constraints

→ compliance with PCI DSS standards

→ must use AES-256 encryption for data storage and RSA encryption for key exchange.

→ must be developed using technologies compatible with high-availability servers.

→ must comply with KYC (know your customer) and AML regulations.

→ Non functional requirements:

→ Security: End-to-end encryption, tokenization, multifactor authentication

→ Reliability: Redundant infrastructure with backups and recovery

→ Scalability: Ability to expand capacity without performance loss.

→ Usability: User-friendly web portal and reporting dashboard.

→ Preliminary schedule ~~and Budget~~

~~Phase~~	Duration
Requirement Analysis	2 weeks
System design	3 weeks
Development	8 weeks
Testing	4 weeks
Deployment	2 weeks
Maintenance	ongoing

Budget

Cost component	Estimated cost (USD)
Requirement & Design	10000
Development	40000
Testing & Infrastructure	15000
Maintenance	20000
Total	15000
	100,000

Date __/__/__
Page _____

Library management system

Problem statement

→ Traditional library systems rely on manual record-keeping which is time consuming, error-prone, and inefficient, necessitating and automated library management system.

1. Introduction:

The purpose of the document is the requirements for developing a library management system that automates book cataloguing, user management, issue/return tracking and reporting for libraries.

Scope:

→ The LMS will provide librarians and users with an efficient platform to manage book borrowing, reservations, catalog searches, fines and membership management. The system aims to reduce manual effort, improve accuracy, and enhance the users experience

General description:

→ The system will maintain record of books, members, issues and returns.

→ It will allow users to search for books by title, author or category

→ Librarians will manage book inventories, membership registrations, and overdue prints.

→ The system will support both on-site and online portals.

→ It will run a secure background server will

Date __/__/__
Page

Q UI

→ functional requirements
→ Book management
→ Add, update, delete and categorize books
→ Track Availability status
→ Member management
→ Register new members
→ maintain profiles with borrowing history.
→ Book issue & return:
→ Record issue/return transactions
→ automatically calculate due dates and overdue fines.
→ search and reservation:
Search by book title, author, subject, or ISBN
→ allow users to reserve books if ~~or~~ unavailable
→ Reports and notifications
→ Generate reports on book usage, fines & member activity.

→ Interface requirements
→ user interface.
→ web portal for librarians & members.
→ mobile app integration for easy access.
→ software interface: Database for storing members and book records.
Notification system for emails/SMS alerts.

→ Hardware interfaces: Barcode/RFID scanners for book identification.
→ secure servers for backend storage.
→ communication Interfaces: Encrypted communication via HTTPS.

Date __/__/__
Page

→ Performance requirements
→ system should support 500 concurrent users
→ Database must handle 50,000+ books, 10,000+ members.
→ Book search results should be displayed within 2 seconds.
→ system availability should be 99%.

→ Design Constraints
→ must comply with data privacy regulations.
→ system should use SQL/NoSQL databases with backup.
→ Only authorized libraries should be able to modify records
→ Developed using widely supported ~~tech~~ technologies (Java; Python, Node.js)

7. Non functional requirements
→ security: Encrypted user data, role based access control.
→ usability: Integrate interface for both librarians and members.
→ Reliability: Automated data backup and recovery.
→ Scalability: Support for expansion to multiple library branches.

→ Preliminary schedule & Budget.
→ ~~schedule~~
Phase

Preliminary schedule

Phase	Duration
Requirement Analysis	1.5 weeks
System design	2 weeks
Development	6 weeks
Testing	3 weeks
Deployment	1.5 weeks
Maintainance	Ongoing
Total	3.5 months / 14 weeks

→ Preliminary Budget

cost component	cost (USD)
Requirement & design	5000
Development	25000
Testing & QA	8000
Deployment & Infrastructure	12000
Maintainance	10000
Total	60000

Stock maintaince system

Problem statement: Manual stock tracking often leads to inaccuracies, delays, and inefficiencies requiring and automated stock maintainance system to manage inventory effectively.

→ Introduction:

→ Purpose of this document is to outline the requirements for the stock maintainace system (SMS) that enables organisations to monitor, update and manage stock levels ensuring real-time accuracy and minimizing losses due to overstocking or stockouts.

→ scope:

→ The SMS will allow buisnesses to record incoming and outgoing stocks, track inventory in real-time, generate reports, provide alerts for low stock. The system will be useful for warehouses, retail stores and supply chain operations to maintain efficient control.

→ General description:

→ The system will maintain records of products, suppliers, purchases and sales.

→ It will track stock levels in real time and send alerts for low or excess stock.

→ users will include store managers, employees and administrators.

→ The system will generate analytical reports to aid decision-making.

→ It will operate on a secure database with a user-friendly interface.

→ functional requirements:

→ Stock management: Add, update, delete stock items.

→ Track quantities, categories, & supplier details.

→ Purchase & sales Tracking: Record purchases from suppliers.

→ update stock automatically when sales are made.

→ stock alerts: Notify users when stock is below minimum or above maximum threshold.

→ Reporting

→ Generate daily, weekly, monthly inventory reports.

→ Provide insights on fast-moving & slow-moving items.

→ User management:- Role based access for staff and manager and admin.

→ Interface Requirements

→ UI - web based dashboard for admin and manager.

→ Simplified interface for employees to update stocks.

→ System interface:
Databases (SQL/NOSQL) for stock recordings.

→ Hardware interface:- Barcode scanners for stock entry.

→ communication interfaces:-
encrypted connection via HTTPS

→ Performance requirements:-

→ The system should support 100 concurrent users.

→ The system should handle 10,000+ stock items.

→ Response time < 2 seconds

→ uptime Requirement: 99% availability.

→ Design constraint: Must comply with data security standards

→ must support multi-location warehouse.

→ Developed using widely supported frameworks.

→ Non-functional-requirements:
Security: Role based access, encryption of sensitive data.

→ usability: stock update forms & dashboard

→ Reliability: Automated backup of stock database

→ maintainability: Modular architecture - for future updates.

→ Preliminary schedule:

Phase	Duration
Requirement analysis	1.5 weeks
System design	2 weeks
Development	6 weeks
Testing	3 weeks
Deployment	1.5 weeks
Maintainance	ongoing

Preliminary budget

cost component	Estimated cost
Requirement & Design	60000
Development	300000
Testing & QA	100000
Deployment & Infrastructure	140000
Maintainance	120000
Total:	720000

Passport Automation System -

Problem statement: Manual passport application and verification processes are slow, error-prone and inconvinent requiring an automated passport Automation system.

→ Introduction:

→ Purpose of the document is to define the requirements for PAS which automates the

Date / / Page

application verification, approval and issuance of passports. This SRS will guide developers, administrators and government authorities in implementing a secure, user-friendly and efficient system.

Scope:
→ The PMS will allow citizens to apply for passports online, schedule appointments, upload required documents, track application status and receive notifications. The system will also assist government officials in verifying documents, conducting background checks, and issuing passports in a streamlined manner.

→ General Description:
→ Citizens can submit passport applications online.
→ The system will allow secure document upload & verification
→ Users can track application status.
→ Officials can review, approve or reject applications & fees.

→ Functional requirements:
→ User registration & authentication:
Citizens register with personal details and government ID.
→ Passport application submission.
→ Fill online forms for new applications / renewals.
→ Appointment scheduling: Choose available slots for biometric data collection / interview.
→ Application Tracking:
→ Real time updates on application status.
→ Payment processing.
→ online payment of passport fee via secure gateways.
→ verification & approval: Government officials verify documents and background checks.
→ Approve / reject application with remarks.
→ Passport issuance: Generate & dispatch passport to applicant.

→ Interface requirements:
→ User interfaces: citizen portal for application and tracking.
→ official portal for verification & approval.
→ Software interfaces → payment gateway API for online payments.
→ Integration with government identity and clinical databases.
→ Hardware interfaces: Biometric devices for fingerprint & photo capture.
→ Communication interfaces: Encrypted data transmission via HTTPS.

→ Performance requirements:
→ System should handle 5000 concurrent users.
→ Response time for form submission & status check should be < 3 seconds.
→ must support upto 1 million applications

annually.

→ Design constraints:
→ must comply with government security standards and privacy regulations
→ Data must be encrypted (AES) and stored.
→ Access restricted to role-based permissions

→ Non functional requirements
→ ~~security~~: End to end encryption.
→ Reliability: Automated backup & disaster recovery.
→ Usability: Multilingual support
→ Scalability: Handle increasing applications across regions.
→ maintainability: Modular design for quick updates.

→ ~~Diff~~ Preliminary schedule:

phase	Duration
Requirement Analysis	2 weeks
System design	3 weeks
Development	10 weeks
Testing	4 weeks
Development	3 weeks
maintanance	Ongoing

Preliminary budget:

cost component	Estimated cost
Requirement & Design	12000
Development	55000
Testing & QA	20000
Deployment & Infrastructure	25000
Maintanance (1 year)	18000
Total	130,000

2/9