

## Template for SRS.

### 1. Introduction

#### 1.1 Purpose of this document

#### 1.2 Scope of this document

#### 1.3 Overview

### 2. General description

### 3. Functional requirement

#### 4. Interface requirement

#### 5. Non-functional requirement (Performance)

#### 6. Design constraint

#### 7. Non-functional

### 8. Preliminary schedule & Budget

## ① Credit card processing system:

### 1. Introduction

1.1 Purpose of the document: To define the requirements for a credit card processing system that securely authorizes, processes and settles credit card transactions for merchants. This system will interface the payment gateways, banks, and merchant applications.

### 1.2 Scope of CCPS:

\* Accept and validate card details from merchants

\* Performing authorization with issuing banks.

\* Process transactions (payment, refund, void)

\* support settlement & reconciliation

\* Provide reporting and analytics to

merchants

- \* Ensure compliance with PCI, DES, EMV standards and local financial regulations

Primary stakeholders:

- \* Merchant - accept credit card payments
- \* customer - make sure payments
- \* Payment processor - convert to networks
- \* Banks - issuing and acquiring

1.2 overview: The document describes functional and non-functional requirements, interfaces, performance and constraints for CCPs.

## 2. General descriptions

- \* product perspective: The system acts as middleware between merchants and functional network. It receives transaction request, validates them, communicate with issuing/acquiring banks, communicate with issuing/acquiring banks and returns authorization results.

### \* Product functions:

- \* Authorization - verify card, balance
- \* Capture / settlement - finalize transaction
- \* Refund / void - Reverses previous transactions
- \* Fraud detection - real-time risk scoring
- \* Reporting - daily, weekly & monthly transaction summary.

## \* User and characteristics:

- merchants - initiate transactions, issue reports
- customers - provide card details, receive receipts.
- system administrators: configure system, manage users, monitor fraud
- Acquiring bank/processor - handle settlement
- operating environment:  
Backend - Licence servers (java, node, js)  
database - postgresql / oracle  
interface - REST / RSO

## \* Assumption & dependencies:

- bank and card network APIs available 24/7
- settlement depends on clearing bank

## 3 Functional Requirements:

- ① The system shall authorize transactions without delay.
- ② The system should support payment types: purchase, pre authorization, capture, refund and void.
- ③ The system shall validate credit card format and expiration.
- ④ The system shall provide tokenization for sensitive card data.
- ⑤ The system shall integrate with fraud detection engine.
- ⑥ It generates daily settlement reports for merchants.
- ⑦ The system shall support both card-present & card-not-present transactions.

- ④ Interface requirements
- User interface: Merchant user portal, API dashboard
  - Software interface: Bank APIs, fraud detection APIs
  - Hardware interface: POS devices, card readers, HSMs.
  - Communication interface: HTTPS/TLS over WAN.

- ⑤ Performance interface requirements:
- System must process at least 15,000 TPS at peak level.
  - Response time < 2 secs
  - Settlement must complete within 2 hours
  - Fraud detection decision latency ≤ 50ms
  - Failure recovery time (FTO) ≤ 5 min.

- ⑥ Design constraints
- must be cloud-native (AWS/GCP)
  - Must use AWS for key storage and cryptography
  - support multi-region deployment for redundancy
  - Integration limited to ISO 8583 & REST for bank network.
  - Logging must be GDPR

- ⑦ Non functional requirement
- Availability > 99.99 %
  - Latency ≤ 2 sec for authentication
  - System must handle 10,000 TPS
  - Data encryption: AES 256 for storage

Phase	Duration	Deliverables
Requirement analysis	3 weeks	Finalized SPS
Architecture & design	7 weeks	System design, security
Development core processing	8 weeks	Transaction validation engine
Development API portal	6 weeks	Merchant API, dashboard
Testing & compliance audit	6 weeks	RAT, PCI
Deployment & go-live	6 weeks	Production system
Training	1 week	Training material

Total duration: 29 weeks (~7 months)

Preliminary budget:

- Personal (7 months) to people teams \$800,000
- Infrastructure (cloud, HSM) \$150,000
- Licensing: \$100,000
- PCI config: \$50,000
- Total: \$1.21 M.

## 1) Hotel management system:

### 1.1 Introduction

1.1.1 Purpose of this document: The purpose of this document is to outline the requirements and specifications for the development of a hotel management system. It will provide a clear understanding of the project objectives, scope and deliverables.

1.1.2 Scope of this document: This document defines the overall working and main objectives of the hotel management system.

It includes a description of the development cost & time required for the project.

1.1.3 Overview: The hotel management system is a software solution designed to streamline hotel operations, including Reservation management, guest check-in / check-out, room assignment, billing and reporting.

2. General description: The hotel management system will cater to the needs of hotel staff and management, providing features such as room booking, guest profiles, inventory management and financial reporting.

### 3. Function Requirements.

#### 3.1 Reservation management:

Allow users to make room reservation online or through the front desk.

Generate reservation confirmations & send notification to guests.

### 3.2 Room management:

- Assign rooms to guests based on availability & preferences.
- Track room status (clean, occupied, vacant) in real-time.

### 3.3 Guest management:

- Maintains guest profiles with personal information, preferences and booking history.
- Facilitate guest check-in and check-out processes.

### 3.4 Billing and invoicing:

- Generate accurate bills for room charges, additional services and taxes.
- Accept various payment methods and generate invoices for corporate clients.

## 4. Interface requirement:

### 4.1 User interface:

- Intuitive and user-friendly interface for hotel staff and guests.

- Accessible via web browsers, mobile devices and desktop.

### 4.2 Integration interface:

- Integration with payment gateways for secure transactions.
- Integration with third-party booking engines.

platforms for seamless reservation management.

## 5. Performance Requirements:

### 5.1 Response time:

- The system should respond to user actions within 2 seconds.

### 5.2 Scalability:

- Handle a minimum of 1000 concurrent users during peak hours.

### 5.3 Data integrity:

- Ensure data consistency and accuracy across all modules.

## 6. Design constraints:

### 6.1 Hardware limitations:

- The system should be compatible with standard hotel hardware (computers, printers, pos terminals).

### 6.2 software dependencies:

- Utilize a relational database management system (eg. MySQL) for data storage.
- Use programming languages and frameworks conducive to XML modeling (eg Java, Spring Boot)

## 7. Non-functional attributes:

### 7.1 security:

- Implement robust authentication and

authorization mechanisms to protect sensitive data.

#### 7.2. Reliability:

- Ensure high availability and fault tolerance to minimize system downtime.

#### 7.3 Scalability:

Design the system to accommodate future growth and expansion.

#### 7.4. Portability:

- Support multiple platforms and devices for user accessibility.

#### 7.5 Usability:

- The system shall have a user friendly interface with clear navigation.

#### 7.6. Reusability:

The system shall use modular code design to facilitate future enhancements and maintenance.

#### 7.7. Compatibility:

- The system shall be compatible with common web browsers.
- The system shall ensure accurate and consistent data storage and retrieval.

~~8. Preliminary schedule and Budget: The development of the hotel management system is estimated to take 6 months with a budget of \$100,000. This includes project planning, development, testing and deployment phases.~~

## 3) Library management system

### 1. Introduction

#### 1.1 Purpose of this document:

The purpose of this document is to define the requirements and specifications for the development of a library management system. It will provide clarity on objectives, scope and deliverables for all stakeholders.

#### 1.2. Scope of this document:

This document describes the features required to automate library functions such as book cataloging, member management, borrowing, returns and fine calculation. It also covers estimated cost and development time.

#### 1.3 Overview:

The library management system is a software solution to streamline library operations, enabling efficient handling of books, members, transactions and reporting.

### 2. General Description:

The system will cater to librarians, members and administrators, providing features like book catalog management, user profiles, loan/renew tracking and fine collection.

### 3 Functional Requirements

#### 3.1 Book management:

- Add, update, and remove book records.
- Categorize books by genre, author and availability.

#### 3.2 Member management:

- Register new members and maintain profiles.
- Track borrowing history and outstanding dues.

#### 3.3 Borrowing and Returning:

- Issue books to members and record due dates.
- calculate fines for late returns.

#### 3.4 Search and Cataloguing:

- Provide a search function by title, author or genre.

- Display availability status in real-time.

#### 3.5 Reporting:

- Generate reports on issued books, overdue books and monthly transactions.

### 4. Interface Requirements

#### 4.1 User interface:

- User friendly interface for librarians & members.

- Accessible through web and mobile platforms.

#### 4.2 Integration Interface:

Integration with barcode scanners and RFID for faster transactions.

#### 5. Performance Requirements:

- System should respond within 2 seconds.
- Handle at least 500 concurrent users.
- Ensure accurate and concurrent usage.
- etc

#### 6. Design Constraints:

- Use MySQL / PostgreSQL for database.
- Compatible with standard desktop hardware and barcode scanners.

#### 7. Non-Functional Attributes:

- Security: Role-based authentication for librarian and member access.
- Reliability: Backup and recovery feature.
- Scalability: Capable of handling large book inventories.
- Usability: Easy navigation and functions.

#### 8. Preliminary schedule and Budget:

~~Estimated development time: 6 months.~~

~~Estimated budget: \$80,000.~~

## g) stock maintenance system:

### 1. Introduction

1.1 Purpose: To outline requirements for a stock maintenance system to manage inventory levels, supplier information and stock movements efficiently.

### 1.2 Scope:

Covers procurement, stock tracking and reporting to ensure availability and minimize loss.

### 1.3 Overview:

A software system to manage inventory, supplier relations, stock in/out records, and alerts for low inventory.

## 2. General description

Designed for warehouses, retail stores and manufacturing units to maintain optimal stock levels and track usage patterns.

## 3. Functional Requirements

### 3.1 Inventory management:

- Add, update and delete stock records.
- Real-time stock level tracking.

### 3.2 Supplier management:

- Maintain supplier details and transaction history.
- Track pending orders and delivery schedules.

### 2.3 Stock transactions:

- Record stock inward and outward movement.
- Generate alerts for low stock levels.

### 2.4 Reporting:

- Generate daily, weekly and monthly inventory reports.

### 3. Interface requirements

- Web and desktop interface.
- Barcode / RFID integration for stock tracking.

### 5. Performance requirements:

- Response time within 5 seconds.
- Handle at least 10,000 product records efficiently.

### 6. Design constraints:

- Use of relational database (MySQL or Oracle)
- Compatible with existing POS hardware.

### 7. Non-functional attributes:

- Security: Role based access (admin, stock manager)
- Reliability: Backup and recovery features
- Scalability: Accommodate growth in inventory size.

### 8. Preliminary schedule and budget:

Estimated time: 4 months

Estimated budget: \$70,000

## 3) Passport automation system:

### 1. Introduction.

#### 1.1 Purpose:

To provide requirements for an automated system to handle passport applications, verification, and issuance efficiently.

#### 1.2 Scope

Includes application submission, document verification, appointment scheduling, fee payment and tracking.

#### 1.3 overview

The systems aim to digitize and streamline passport services for applicants and authorities.

## 2 General description:

The system will cater to citizens applying for passports, verification officers and administrative staff.

## 3. Functional requirements

### 3.1 Application processing:

- Online form submission with document upload.

Application status tracking in real-time.

### 3.2 Appointment scheduling:

- Schedule appointments at nearest passport office.

• Reschedule or cancel appointments.

### 3.3 Verification and approval:

- Officer dashboard for document verification and approval.
- Integration with police verification database.

### 3.4 Payment processing:

- Secure online payment for passport fees.

### 3.5 Reporting:

- Generate daily and monthly reports on applications processed.

## 4. Interface Requirements:

- Web-based and mobile friendly interface.
- Integration with government ID databases for verification.

## 5. Performance requirements

- Handle at least 5000 concurrent users.
- Response time within 3 seconds.

## 6. Design constraints:

- Must comply with government security and privacy standards.
- Use of secure database with encryption.

## 7. Non-Functional attributes

- Security: multi-factor authentication and encrypted data storage.
- Reliability: High availability and disaster recovery system.
- Usability: clear navigation for citizens and officers.

### 3. Preliminary schedule and budget

~~Estimated development time 8 months~~

~~Estimated budget: \$150,000~~

22/01/28

Initial site selection

Geological survey

Geological survey

Soil testing

Soil testing

Land acquisition

Land acquisition

Permit applications

Permit applications

Site preparation

Site preparation

Construction

Construction

Infrastructure

Infrastructure

Equipment

Equipment

Personnel

Personnel

Logistics

Logistics

Contingency

Contingency

Total cost

Total cost

Timeline

Timeline

Completion date

Completion date

Cost per month

Cost per month

Cost per day

Cost per day

Cost per hour

Cost per hour

Cost per minute

Cost per minute

Cost per second

Cost per second

Cost per millisecond

Cost per millisecond

Cost per microsecond

Cost per microsecond

Cost per nanosecond

Cost per nanosecond