LAB-1

22/08/25

Software Requirements Specification

① SRS for Hotel management System (HMS)

1. Introduction

1.1 Purpose of this document:

The purpose of this document is to define the functional, non-functional, and system requirements for a hotel management system (HMS). The HMS will streamline hotel operations including reservations, guest services, housekeeping, billing, reporting, and integration with external services. The specification will serve as a reference for developers, testers, project managers, and stakeholders to ensure alignment during design, development, deployment, and future enhancements.

1.2. Scope of this document:

The HMS will be a centralized web and mobile platform that provides complete hotel management features. It will manage reservations, check-in / check-out, room allocation, housekeeping, point-of-sale (POS) integration, billing and invoicing, guest profiles, reporting and analytics. The system will support multi-property operations and integrate with OTAs (online travel agencies), payment gateways and accounting software. Role-based access will ensure secure and controlled use by administrators, staff, and guests.

1.3. Overview

The HMS will consist of modular subsystems for reservation management, room & housekeeping, guest management, Billing & invoicing, channel management, and reports & analytics. It will expose APIs for third-party integrations and provide intuitive UIs for staff and guest self-service.

2. General Description:

* User classes & roles:
  - Front Desk Agents: Handle check-ins, check-outs, and reservations.
  - Housekeeping Staff: Update room cleaning status and task management.
  - Hotel managers: Monitor operations, generate reports, and manage staff.
  - Finance Staff: Handle billing, payments, and accounting.
  - Guests: use self-service portals for booking, check-in /out, and feedback.

* Assumptions:
  - Stable internet access is available for cloud deployment
  - Digital payment systems (cards, UPI, wallets) are accessible
  - Hotels may use existing hardware (scanners, printers, locks).

* Dependencies:
  - OTA platforms for reservations
  - Payment gateway services
  - Government taxation and compliance APIs.

3. Functional requirements:

3.1 Reservation management:

* create, modify and cancel reservations from front-desk or online portals.
* Real-time room availability and dynamic pricing
* Group / corporate bookings with discount codes and contracts
* Automated confirmations emails / SMS.

3.2. Room and Housekeeping Management

* Assign / swap rooms, track room status (vacant, occupied, Dirty...)
* Generate housekeeping task lists per shift
* Track maintenance issues and schedule repairs.

3.3. Guest management:

* Maintain guest profiles including KYC, stay history, and preferences.
* Digital check-in with ID scanning and e-signature
* Loyalty program integration.

3.4. Billing and Invoicing:

* Generate folios for guests with room, service, and POS charges.
* Support multi-currency billing and tax rules.
* Split / merge folios, refunds, and adjustments.
* Corporate invoicing and export to accounting software.

3.5. Channel & OTA Integration:

* Real-time sync of rates and availability with OTAs.
* Automatic import of OTA reservations.
* Prevent overbooking through centralized inventory

3.6. Reporting & Analytics:

* Occupancy, ADR, RevPAR, guest demographics.
* Housekeeping productivity reports
* Financial statements with revenue by source
* Forecasting and trend analysics.

4. Interface requirements:

4.1 User Interface:

* Responsive web application for staff.
* Guest self-service portal for bookings and check-in/out
* Mobile app for housekeeping staff.
* Accessibility compliant with WCAG 2.1 AA.

4.2 Integration Interfaces:

* REST / GraphQL APIs for external systems.
* OTA integration via XML/JSON APIs.
* Payment gateway APIs for online transactions.
* Integration with POS systems, accounting software, and digital door locks.

5. Performance Requirements:

* 95% of UI requests should respond within 2 seconds.
* System should support 1000 concurrent users.
* Handle up to 10000 reservations per day
* Synchronization with OTAs should not exceed 5 seconds of delay.
* System uptime ≥ 99.9%

6. Design Contraints:

* Must comply with local tax, GST and e-invoice rules.
* PCI DSS compliance for payment handling
* Cloud-native deployment (containerized with Kubernetes/Docker)
* Use RDBMS (PostgreSQL/MySQL) with caching (Redis) for performance

7. Non-functional requirements:

* Security: Role-based access, MFA, audit logs, end-to-end encryption.
* Reliability: Active-active redundancy, automated backups (RPO ≤ 15 mins, RTO ≤ 1 hour)
* Scalability: Horizontal scaling to handle peak seasons.
* Usability: Intuitive dashboards, minimal training required
* Compatibility: Cross-browser support (Chrome, Edge, Firefox, Safari)
* Maintainability: Modular microservice architecture for easier upgrades.
* Data Integrity: Strict ACID compliance for billing and reservations.

8. Preliminary Schedules and Budget:

Phase 1 (Planning & Design): 1 month
Phase 2 (MVP- Reservations, Billing, Guest Mgmt): 3 months.
Phase 3 (Integrations - OTA, Payment, POS): 2 months
Phase 4 (Reports & Advanced features): 2 months
Phase 5 (Testing & Pilot): 1 month
Phase 6 (Production Deployment): 1 month

Total timeline: ~10 months
Estimated Budget: $120k - $160k

22/8

(5) SRS for Credit Card Processing System (CCPS):

1. Introduction:

1.1 Purpose of this document:

The purpose of this document is to define the requirements for a Credit Card Processing System (CCPS) that securely handles transaction authorization, clearing, settlement, risk management, and dispute resolution. It is designed for acquirers, issuers, merchants, and payment processors to facilitate card-present and card-not-present transactions.

1.2 Scope of this document:

The CCPS will support credit, debit, and prepaid card transactions through POS devices, e-commerce platforms, and mobile payment applications. The system will interface with global card networks (Visa, Mastercard, RuPay, Amex), issuing banks, acquiring banks and merchants. It will provide a modular architecture for transaction processing, fraud detection, settlement, and reporting. The system must comply with PCI DSS, EMV, and other relevant financial regulations.

1.3 Overview:

The CCPS will consist of core modules: Transaction Ingestion, Authorization & Routing, Risk & Fraud Detection, Clearing & Settlement, Chargeback Management, and Reporting. It will provide APIs for integration with merchants and banking systems while ensuring real-time processing and high availability.

2. General Description:

* User Classes & Roles:
  - Merchants: Submit transactions and view settlement reports
  - Acquirer Operations Staff: Monitor transaction flows and settlements.
  - Issuers: Authorize and manage cardholder accounts.
  - Risk Analysts: Detect and mitigate fraud patterns
  - Finance & Compliance Teams: Ensure regulatory reporting and audits.

* Assumptions:
  - Secure network connectivity with payment schemes is available
  - Hardware Security Modules (HSMs) will be provisioned for encryption.
  - Card schemes provide certification and compliance documentat-ion.

* Dependencies:
  - Card networks (Visa, Mastercard etc.)
  - Payment gateways for merchant integration.
  - Regulatory compliance systems (AML, KYC, GDPR/DPDP)

3. Functional Requirements:

3.1 Transaction Ingestion:
* Accept card-present and card-not-present messages.
* Validate message formats, BIN ranges, and merchant credentials.
* Tokenization support for sensitive card data.

3.2 Authorization & Routing:
* Route transactions to the appropriate issuer.
* Support EMV, magnetic stripe and contactless transactions.
* Fallback handling and stand-in authorization if issuer unavailable

* Timeout, retry and response code management

3.3. Risk and Fraud management:
* Rule-based and AI/ML-driven fraud scoring
* Velocity checks, geolocation, and device fingerprinting
* Real-time blacklists and watchlists.
* 3D secure (3DS) integration for online transactions

3.4. Clearing & Settlement:
* Batch processing with card networks for clearing
* Fee and interchange calculations
* Generate settlement files for acquirers and merchants
* Reconciliation with issuer/acquirer ledgers.

3.5. Chargebacks & Disputes:
* Case creation and evidence submission workflows.
* SLA-based timelines for dispute resolution.
* Automated notifications to merchants and issuers.
* Reporting of chargeback reason codes.

3.6. Reporting and Analytics:
* merchant dashboards with daily/weekly settlements
* Fraud detection and alerting dashboards.
* Regulatory compliance reports.
* Export to CSV/PDF formats.

4. Interface Requirements:

4.1. User Interface:
* web portal for merchants and acquirer staff
* Ops dashboard for transaction monitoring
* Export options for finance teams.
* Accessibility

4.2. Integration Interfaces:

* ISO 8583 message interface with networks
* REST / GraphQL APIs for merchants and payment gateways.
* HSM integration for cryptographic operations.
* Integration with AML/KYC systems

5. Performance Requirements:

* Authorization response time: ≤ 800 ms (domestic) and ≤ 1200 ms (international)
* Throughput ≥ 500 TPS, scalable to 2000 TPS.
* Uptime: 99.99% with failover capability.
* Data consistency for financial transactions: Exactly-once semantics.

6. Design Constraints:

* PCI DSS level 1 compliance
* EMV, 3DS, PSD2 SCA compliance
* Immutable audit logs retained for 7-10 years.
* Secure key management with HSM.

7. Non-functional requirements:

* Security: TLS 1.3, mTLS for inter-service communication, end-to-end encryption.
* Reliability: Active-active architecture with disaster recovery
* Scalability: Horizontally scalable microservices.
* Portability: Cloud-agnostic containerized deployment
* Usability: Intuitive dashboards with clear error codes,
* compliance: Regular penetration testing, vulnerability scans.
* Data integrity: ACID compliance with ledgers, cryptographic verification.

8. Preliminary Schedule and Budget:

Phase 1 (Requirements & Design): 2 months
Phase 2 (Core Authorization & Routing): 3 months
Phase 3 (Fraud & Risk Engine): 2 months
Phase 4 (Clearing & settlement): 2 months
Phase 5 (Chargeback & Reporting): 2 months
Phase 6 (Testing & certification): 3 months
Phase 7 (Production Rollout): 1 month
Total Timeline: ~15 months
Estimated Budget: $300k - $500k (depends on compliance certifications and scheme integrations)

22/8

③ Library Management System (LMS)

1. Introduction

1.1 Purpose of this document:

The purpose of this document is to define the requirements for a Library Management System (LMS) that manages collections, patrons, acquisitions, circulation, and reporting. It will serve as a reference for developers, testers, administrators and stakeholders to ensure consistent understanding during development, deployment, and future enhancements

1.2. Scope of this document:

The LMS will provide a centralized platform for librarians and patrons. It will support cataloging (MARC / Dublin Core), acquisitions, circulation, patron services, e-resource management, and reporting. The system will expose a patron-facing OPAC (Open public access catalog) and staff dashboards. It will integrate with self-checkout kiosks, RFID / barcode systems, digital resources, and payment gateways for fines.

1.3 Overview:

The LMS will include core modules for Catalog & Metadata, Circulation, Patron Services, Acquisitions & Serials, Digital Resources, and Reports & Analytics. It will offer web and mobile-friendly interfaces for both patrons and staff.

2. General description:

* User classes and Roles:

- Librarians - Manage collections, acquisitions, cataloging, and circulation.
- Catalogers - Enter and maintain metadata records.
- Patrons - Search catalogs, borrow/return books, and access e-resources.
- Admins - Manage users, permissions, and system configuration
- Finance Staff: Handle fines, budgets and vendor payments

* Assumptions:
- Libraries use barcode / RFID tags for book circulation
- Campus / organization has an authentication system (SSO)
- Internet connectivity is stable for digital resources.

* Dependencies:
- Metadata services (Z39.50, SRU)
- Payment gateways for fines
- Email / SMS providers for notifications.

3. Functional Requirements:

3.1 Catalog & Metadata Management
* create / import metadata in MARC21 or Dublin Core
* Authority control, subject headings, and classification (DDC / UDC)
* Deduplication and record merging
* Integration with external metadata providers

3.2 Circulation Management:
* Borrow, return and renew items.
* Place holds and recalls.
* Fines and overdue management with policies.
* RFID / barcode scanning for transactions
* Self-checkout kiosk integration.

3.3. Patron Services

* Patron registration, profiles, and borrowing history
* OPAC with search, filters, and relevance ranking
* Wishlists, recommendations, and reading lists.
* Notifications via email/SMS for due dates and reservations.

3.4. Acquisitions & Serials:

* Manage vendor orders, invoices, and budgets.
* Subscriptions management for periodicals.
* Track receipts and renewal cycles.

3.5. Digital Resource Management:

* Manage eBooks, audiobooks, and licensed databases.
* DRM and proxy integrations (EZproxy / Shibboleth).
* Usage tracking and statistics.

3.6. Reports & Analytics:

* Circulation reports (top borrowed books, overdue trends)
* Collection usage and turnover rates
* Acquisition budget reports
* Patron activity dashboards.

4. Interface Requirements:

4.1 User Interface:

* Staff web application with circulation and cataloging modules.
* Patron-facing OPAC accessible via desktop and mobile
* Accessibility compliant (WCAG-2.1 AA)
* Multi-language support.

4.2. Integration Interfaces:

* Z39.50/SRU for metadata import
* SIP2 / NCIP for self-service kiosks
* Payment gateway APIs for fines.
* REST APIs for external applications.
* SSO integration (SAML / OAuth2).

5. Performance Requirements:
* Catalog search response time ≤ 1s (95th percentile)
* Borrow/return operations ≤ 2s.
* Support collections of up to 5 million records.
* Handle 10K concurrent patron sessions.
* uptime ≥ 99.9%.

6. Design constraints:
* Must comply with MARC21, RDA, Dublin Core, and Z39.50 standards.
* Relational database with indexing for fast search.
* ElasticSearch / Solr for discovery
* Support for both on-premise and cloud deployment

7. Non-functional attributes:
* Security - Patron data encryption, role-based access, GDPR/DPDP compliance
* Reliability - Regular backups, failover clusters.
* Scalability - Horizontal scaling for search and circulation
* Usability - Simple, intuitive interfaces for patrons & staff
* Compatibility - Barcode/RFID scanners, label printers.
* Maintainability - Modular service-based architecture,
* Data integrity - Enforce referential integrity accross catalog and circulation

~~8. Future Enhancements~~

8. Preliminary Schedule & Budget:

* ~~AI-powered recommendation engine~~

* Phase 1 (Requirement & Design): 1 month
* Phase 2 (Core Catalog & Circulation): 3 months
* Phase 3 (Patron Services & OPAC): 2 months
* Phase 4 (Acquisitions & Digital Resources): 2 months
* Phase 5 (Reporting & Analytics): 1 month
* Phase 6 (Testing & Pilot): 1 month
* Phase 7 (Deployment): 1 month

Total Timelines: ~ 10 months

Estimated Budget: $100K – $150K

22/8

④ SRS for Stock Maintenance System (SMS):

1. Introduction:

1.1. Purpose of this document:

The purpose of this document is to define the requirements for a Stock Maintenance System (SMS) that efficiently manages inventory levels, stock movements, suppliers, and valuation. The system will provide real-time visibility into stock availability across warehouses, stores and supply chains, ensuring accurate tracking and reducing operational inefficiencies.

1.2 Scope of this document:

The SMS will manage item masters, procurement, receiving, stock adjustments, transfers, replenishment planning, and reporting. It will support barcode/QR scanning, optional RFID integration, and provide APIs for integration with ERP, accounting systems, and e-commerce platforms. Both web and mobile interfaces will be provided for administrators, warehouse staff, and managers.

1.3. Overview:

The system will include modules for Item & warehouse management, Inbound & outbound operations, Stock movements and adjustments, Replenishment & Planning, and Valuation & Reporting. It will support multi-warehouse environments, bin management, and role-based access.

2. General Description:

- User classes & Roles:

* warehouse staff - Manage receiving, put-away, picking, and cycle counts.
* Storekeepers - Track stock levels at store locations
* Procurement Team - Manage purchase orders, suppliers and reordering
* ~~Br~~ Managers - Monitor KPIs, approve transactions and analyze reports.
* Finance & Auditors - validate inventory valuation and compliance

• Assumptions:
* Barcode scanners and printers will be available
* Warehouses/stores may use handheld mobile devices.
* Stable internet connectivity with offline fallback for floor operations.

• Dependencies:
* ERP/Accounting systems for financial integration
* E-commerce platforms for stock synchronization.
* Courier/shipping APIs for outbound logistics.

3. Functional requirements:

3.1. Item & warehouse Management:
* Item master with SKU, barcode/QR, description, category, and units.
* Batch and serial number tracking with expiry dates
* Multi-warehouse support with zones, bins and storage rules.
* Min/Max levels and ABC classification

3.2. Inbound & Outbound Management:
* Good receipt Note (GRN) for purchased items.
* Put-away suggestions based on bin rules
* Pick/pack/ship workflows with shipment labels.

* Supplier and customer returns with reason codes.

3.3. Stock Movements & Adjustments:

* Stock transfers between warehouses / stores.
* Cycle counts, periodic stock-taking, and adjustments.
* Audit trails for all movements.
* Reservation of stock for orders and production.

3.4. Replenishment & Planning:

* Automated reordering based on EOQ and safety stock
* Forecasting demand using historical sales.
* Vendor performance tracking and lead time management
* Purchase requisition suggestions.

3.5. Valuation & Reporting:

* FIFO, LIFO and weighted average cost valuation.
* Landed cost allocation for imports.
* Reports: stock aging, shrinkage, inventory turnover, stockout trends.
* Dashboards with KPIs (inventory accuracy, holding costs, service level)

4. Interface Requirements:

*

4.1. User Interface:

* Web portal for managers and procurement staff.
* Mobile-friendly UI for warehouse staff with offline mode
* Barcode / QR code scanning support
* Bulk import/export of data via CSV / Excel
* Print labels and stock reports.

4.2. Integration Interfaces:

* REST APIs for ERP, e-commerce, and accounting integration.
* Webhooks for stock updates to sales platforms.
* Carrier APIs for shipment tracking
* Hardware integration with barcode scanners, label printers, and RFID devices.

5. Performance Requirements:
* Inventory transaction processing latency ≤ 1 second
* Bulk uploads (50k+ records) should complete in ≤ 10 minutes.
* Support up to 500 concurrent handheld devices.
* Uptime ≥ 99.9%.

6. Design Constraints:
* Must support GS-1 compliant barcode standards.
* Compatibility with handheld Android-based scanners.
* Relational database (PostgreSQL/MySQL) with message queues.
* Optional time-series database for telemetry and IoT integration.

7. Non-functional requirements:
* Security - Role based access, audit trials, encryption of sensitive data
* Reliability - offline caching for handheld devices with sync on reconnect
* Scalability - Partitioning by warehouse and region
* Usability - simple handheld workflows requiring minimal training
* Maintainability - Modular design for easy feature addition.
* Data integrity - Double-entry ledger for stock movements.
* Compliance - Support for tax rules, inventory audit requirements.

8. Preliminary Schedule & Budget:

* Phase 1 (Requirements & Design): 1 month
* Phase 2 (Core Inventory & Item Management): 3 months
* Phase 3 (Inbound / Outbound Operations): 2 months
* Phase 4 (Replenishment & valuation): 2 months
* Phase 5 (Reporting & Dashboards): 1 month
* Phase 6 (Testing & Deployment): 1 month

Total Timeline: ~ 10 months

Estimated Budget: $120k – $170k

(5) SRS for Passport Automation System (PAS):

1. Introduction:

1.1 Purpose of this document:

The purpose of this document is to define the requirements for a Passport Automation System (PAS) that digitizes and streamlines the passport application, verification, issuance, and delivery process. It will serve as a guideline for developers, government agencies, and stakeholders to ensure accurate implementation and compliance with national regulations.

1.2. Scope of this document:

The PAS will provide online services for citizens to apply for passports, upload documents, make payments, book appointments, and track status. It will integrate with government ID registries, police verification systems, and courier services. Staff modules will support verification, approval, printing and grievance redressal. The system must ensure high security, scalability, and legal compliance.

1.3. Overview:

The PAS will consist of modules for Application Intake, Appointment Scheduling, Verification Workflows, Printing & Dispatch, Status Tracking, Grievances, and Reporting. It will provide multilingual citizen-facing portals and secure staff dashboards.

2. General Description:

- User Classes & Roles:

* Citizens / Applicants - submit applications, schedule appointments, track status.
* Application Center Staff: Assist citizens, validate applications, collect biometrics.
* Verification officers: Review applications, documents and background checks.
* Police Authorities: Conduct physical address and identity verification.
* Admins: Manage system configurations, security and user access.
* Supervisors: Monitor SLAs, track pending applications, generate reports.

• Assumptions:
* citizens have access to internet-enabled devices.
* National ID registry is available for identity verification
* Government approved payment gateways are accessible

• Dependencies:
* APIs for e-KYC, ID verification and address validation
* Courier / shipping providers for passport delivery
* SMS / email providers for notifications.

3. Functional Requirements:

3.1. Application Intake:
* online account ~~creating~~ creation and authentication
* multi-step application forms with validation.
* Document upload with file format and size restrictions
* Automatic fee calculation based on service type (normal / tatkal)
* Payment integration with digital payment gateways.

3.2. Appointment Scheduling & Biometrics:

* Real-time appointment availability across centers
* Reschedule or cancel appointments with limits
* QR/Barcode-based appointment slip.
* Capture biometrics: fingerprint, photograph, digital signature
* Quality checks for biometric data.

3.3 Verification workflows:

* Automated ID verification via e-KYC APIs.
* Document verification by staff with approval/rejection workflow
* Police verification requests routed to local police stations.
* SLA-based monitoring of verification timelines.

3.4. Printing, Issuance & Dispatch:

* Secure print queue with MRZ and RFID chip encoding
* Personalization of passports with holograms and serials.
* Dispatch tracking via courier APIs.
* Delivery confirmation updates for citizens.

3.5. Status Tracking, Notifications & Grievances:

* Citizens can track real-time status via portal/mobile app.
* SMS/email notifications at key stages (submission, approval, dispatch).
* Grievance redressal portal with ticket management.
* Feedback surveys post-delivery

3.6. Reports & Analytics:

* Application volumes by region, type and status.
* SLA compliance dashboards.
* Fraud detection ~~errors~~ alerts (multiple applications, identity mismatch).
* Regulatory and audit reports.

4. Interface Requirements:

4.1. User Interface:

* Multilingual, mobile-responsive citizen portal
* Staff dashboards for intake, verification and printing
* Accessibility compliance (WCAG 2.1 AA)
* Kiosk-friendly design for public centers

4.2. Integration Interfaces:

* APIs for National ID registry, police verification, and digital signatures.
* Payment gateway APIs for online transactions.
* Courier APIs for delivery tracking.
* PKI-based document signing

5. Performance Requirements:

* Portal response time ≤ 1.5 seconds (95th percentile)
* Support up to 50,000 concurrent citizen sessions.
* SLA compliance tracking within ±5% accuracy
* Uptime ≥ 99.95%

6. Design constraints:

* Must comply with data protection and privacy laws (GDPR/DPDP)
* Biometric data must be securely encrypted and stored in compliance with ISO standards.
* Secure key management using HSM.
* Immutable audit trail for legal compliance

7. Non-Functional Requirements:

* Security - MFA, end-to-end encryption, strict access controls.
* Reliability - Multi-region disaster recovery setup
* Scalability - Elastic cloud scaling during peak demand

* Usability - Guided forms, contextual ~~help~~ help, error prevention.
* Maintainability - Modular microservice architecture.
* Data integrity - Strong consistency for application data.
* Compliance - Adherence to ICAO passport issuance standards.

8. Preliminary Schedule and Budget:

* Phase 1 (Requirements & Design): 2 months
* Phase 2 (Citizen portal & Application Intake): 3 months.
* Phase 3 (Verification & Police Integration): 3 months.
* Phase 4 (Printing & Dispatch module): 2 months.
* Phase 5 (Reports & Grievances): 1 month
* Phase 6 (Testing & Pilot): 2 months.
* Phase 7 (Production Deployment): 1 month

Total Timeline: ~14 months
Estimated Budget: $250k - $400k

22/8