Lab-1

Hotel Management System

Software Requirement System

① Introduction

a. Purpose of document :- The purpose of this document is to define the requirement for the HMS. It describes the functional and non-functional requirements, constraints, design, system specification to be developed for efficient hotel operations and implementation of the system.

b. Scope of document :- The HMS will automate the process of booking, check-in, check-out, room allocation, billing and reporting for hotel operations. It will allow customers to reserve online or at front desk, maintain a centralized data for room availability, customer records and transaction history, enable hotel staff to manage booking, customer details and payments.

c. overview :- The HMS is the web-based system. It will integrate modules for booking, room management, payments/billing, reporting and user management.

② General Description :-

a. Product Functions :- · Online and offline room booking
· check-in/check-out management
· Room availability display
· Customer record management
· Billing and invoice generation
· Daily, monthly and yearly reports

b. User Characteristics :-

· Receptionists :- Basic computer skills, responsible for daily operation like check-in, check-out, booking.

· Admin :- Full system control, including management, staff, managing rooms, reports and billing settings.

· Guest :- can search for available rooms, make bookings and view payment details.

③ Functional Requirements :-

a. Booking Management

· Users can search ~~boot~~ room by date, type and price

· Customers can reserve room online

· Staff can add, modify or cancel bookings.

b. check-in/ chek out

- Staff can assign the rooms to guests at check-in
- System automatically updates room status after check-out.

c. Billing

- System calculates total charges and generates invoices
- Supports multiple payment methods.

d. Reporting

- Generate occupancy and revenue reports.

e. User Managment

- Admin can create, edit and delete user accounts
- Assign user roles.

④. Interface Requirement

a. User Interface

- Clean dashboard with navigation menu.
- Search and filter options for rooms and bookings
- Color coding for room availability.

b. Hardware Interface

- PC with minimum 4GB RAM, 2GHz processor

c. Software Interface

- Works on browsers.
- Integration with payment gateway APIs.

⑤ Performance Requirements

- The system must handle at least 500 concurrent bookings without performance degradation.
- Response time for queries should not exceed 2 sec
- The system must be available 24/7 with 99% uptime.

⑥. Design Constraints

- The system must comply with data protection laws
- The database must support ACID transactions
- Use of open-source frameworks is encouraged for cost savings.

⑦. Non-functional Requirements

- Security: Password encryption, role-based access control.
- Reliability: Backup system every 24 hours
- Usability: Simple, intuitive UI for staff.
- Scalability: Ability to handle more branches in future.

(8). Preliminary Schedule

Phase	Duration
Requirement Analysis	1 week
System Design	2 week
Development	4 week
Testing	2 week
Deployment	1 week
Total	10 week

(9). Budget

Item	cost
Requirement & Design	₹ 20,000
Development	₹ 60000
Testing	₹ 15000
Deployment & Training	₹ 10000
contingency	₹ 10500
Total	₹ 1,15500

Lab-2

Credit Card Processing System.

Software Requirement System

① Introduction

a. Purpose of document :- Specify functional, non-functional and interface requirements for CCPS that authorizes, captures, settles and reconciles credit card transactions.

b. Scope of document :-

- Process card-present and card-not-present payments
- Support auth, capture, void, refund, partial refund, recurring billing and payouts
- Provide risk scoring/fraud checks, tokenization and chargeback handling.
- Integrate with card network
- offer merchant portal, reporting and APIs

c. overview :- Modular, service-oriented web platform with secure APIs, merchants dashboard, risk engine, settlement/reconciliation service, and observability.

②. General Description :-

a. Users and Roles :-

- Cardholder
- Merchant (owner, staff, finance)
- Support Agent
- Risk Analyst
- System Admin
- Acquirer

b. Operating Environment

- Cloud + HSM for crypto
- Web
- REST

c. Assumptions / Dependencies

- Connectivity to acquirer gateway and card schemes
- KMS | HSM availability for key management.
- Compliance programs.

③. Functional Requirement

a. Payments

- Authorize transaction
- Sale
- Refund
- Multi-currency support.

b. Risk and Compliance

- Tokenization of PAN
- Blacklist | whitelist management
- Dispute | Chargeback intake.

c. settlements and Reconciliation

- Batch settlement files to acquirer
- Fees calculation, payout statements
- failed payout handling

d. Refund Processing

Requirement :- The system must allow full or partial refunds

outcome :- Refunded amounts are credited back to the cardholder

④. Interface Requirement

a. User requirement

- Merchants Portal
- Hosted Payment Page
- Support Console

b. ~~Hardware~~ Software Interface

- Database : SQL database for transactions and logs.
- Payment Gateway :- Integration with acquiring banks and card network.

5. Performance Requirement :-

a. System response time

- Authorization : ≤ 3 sec
- Refund initiation : ≤ 5 sec

b. Scalability :- supports at least 1000 concurrent transaction without degradation

6. Design Constraints :-

- Must comply with PCI DSS 4.0 standards
- Compatible with EMV chip & magnetic stripe and contactless
- operates on server with minimum 8GB RAM and 1 TB storage.

7. Non-Functional Requirment :-

- Security
- Privacy
- Reliability
- ~~Maintainability~~
- ~~Portability~~

(8). Preliminary Schedule and Budget.

Phase 1: Design 1 month

Phase 2: Implementation 3 months

Phase 3:- Testing and Compliance Audit - 2 mo

Phase 4:- Deployment and merchant Training - 1 month

Item	Estimate
Engineering	₹1.2 - ₹1.8 Cr
Security & compliance	₹35 - ₹60 L
Infra	₹20 - ₹35 L
License / Tools	₹8 - ₹12 L
Contingency	₹20 - ₹35 L
Total	₹2.05 - 2.85 Cr

Lab-3

Library Management System

(1). Introduction

a. Purpose :- Define functional, non-functional and interface requirements for a system that manages library collections, members services, circulation, e-resource and reporting for academic/public libraries.

b. Scope :-

- Support acquisitions, cataloging, circulation
- Provide member onboarding, self-service portal, discovery search and OPAC.
- Integrate with barcode scanners, e-book providers and payment gateway for fines
- offer staff console, reports/analytics and REST APIs

c. Overview :- Modular web platform with staff UI, OPAC portal, cataloging tools, circulation engine notification service and reporting. Supports SSO and standards.

②. General Description

a. Product Perspective :- Works as central library middleware with integrations to identify provider, LMS of the university and external metadata services.

b. Users and Roles.

- Visitors / Guest
- Member / student / Faculty
- Librarian / Circulation staff
- Cataloguer / Acquisition officer
- Finance clerk
- System Admin

c. Operating Environment

- cloud or on-prem Linux
- web + responsive OPAC
- REST / GraphQL API's.

③. Functional Requirements

a. Catalog and inventory

- Create / edit bibliographic records, import via CSV
- Manage items
- Authority control

~~b. Acquisitions :-~~

b. Book Management :

- Add, update, delete book records in the catalog
- Categorize books by title, author, subject

c. Member Management

- Register new members and ~~mini~~ maintain member profiles.
- Track member borrowing history and preferences.

d. Lending and Returns

- Allow users to borrow and return books.
- Calculate and apply fines for overdue time

e. Billing and fees

- Generate bills for late fees and membership renewals.
- Accept multiple payment methods and provide receipts.

④ Interface Requirements

a. User interface

- Intuitive and user-friendly interface for librarians, staff, members.
- Accessible via web browsers, mobile devices and desktop applications.

b. Integration Interface

- Integration with payment gateways for fine and fee transactions.
- Integration with third-party digital library platform.

⑤. Performance Requirement.

a. Response time :- The system should respond to user actions within 2 seconds.

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

c. Data Integrity :- Ensure data consistency and accuracy across all modules.

⑥. Design constraits

a. Hardware Limitation - The system should be compatible with standard library hardware

b. Software Dependencies - Utilize a relational database management system. Use programming language and frame works.

(7). Non-Functional Attributes

- Security - Robust authentication and authorization mechanisms
- Reliability : high availability and fault tolerance.
- Scalability : Designed for future growth
- Portability : Multi-platform support.
- Usability : User-friendly interface with clear navigation.

(8). Preliminary Schedule and Budget.

The development of the library Management system is estimated to take 6 months with a budget of $ 80,000. This includes project planning, development, testing and deployment phases.

Stock Maintenance System

①. Introduction

a. Purpose :- The purpose of this document is to outline the requirements and specification for a stock maintenance system. It will provide a clear understanding of the projects objectives, scope and deliverables.

b. Scope of this document :- This document defines the functionality of the stock maintenance system, which includes managing stock levels, tracking inventory, purchase orders, sales, transfer and reporting.

c. Overview :- The SMS is a software application designed to efficiently manage and monitor stock levels, automate procurement and replenishment.

② . General Description

The system will be used by warehouse staff, supervisors, finance teams and managers. It will stock entry, batch management, order fulfilment, adjustment and reporting while being accessible to users with different technical skills.

③ . Functional Requirement

a. Product and Inventory Management

- Add, update and delete stock items
- Maintain batch/lot and expiry details

b. Procurement and Receiving

- Record purchase orders and goods receipt notes
- Update stock levels automatically upon receipt

c. stock Movements

- Track stock transfers both between warehouse.
- Record stock issues, sales and adjustements

d. Alerts and Replenishment

- Generate alerts for low stock levels
- Suggest reorder quantities.

④. Interface Requirements

a. User interface

- Intuitive dashboard for stock monitoring
- Accessible on web browsers and handheld devices

b. Integration Interface

- Integration with POS, E.R.P and accounting system.

⑤. Performance Requirement

a. Response Time :- Transaction should be processed with 2 seconds

b. Scalability - handle at least 500 concurrent users.

c. Data Integrity - Maintain accurate and up-to-date inventory records

⑥. Design Constraints

a. hardware limitations :- must support barcode scanners, label printers and mobile handheld devices

b. software dependencies :- utilizes RDBMS, such as MySQL/PostgreSQL with backend framework like Node.js, Django or

spring Boot.

7. Non-functional Attributes.

- Security : Role-based access control
- Reliability : Fault tolerance with backup and recovery.
- Portability : compatible with web and mobile devices.
- Usability : Simple, clear navigation for warehouse staff.
- Data Integrity : Use Audit trails and validation rules.

8. Preliminary Schedule and Budget

The development of this stock Maintenance System is estimated to take 5 months with a budget of $100,000 including design, implementation, testing and deployment.

Lab-5.

Passport Automation System

① Introduction

a. Purpose of this Document :-

The purpose of this document is to outline the requirement and specification for a passport automation system. It aims to provide clarity on objectives, scope and deliverable of the project.

b. Scope of this Document

This document defines the main objective of the Passport automation System. It includes application submission, verification, biometric capture, fee payment appointment scheduling, passport printing and status tracking.

c. Overview :- The passport Automation System is a software solution intended to automate and streamline the end-to-end process of passport issuance, ensuring efficiency, transparency and compliance with government regulations.

(2). General Description :-

The system will be used by application fronts - office operators, verification officers and administrators. It will manage passport applications biometric capture, background verification, fee collection and status noti-fications.

(3). Functional Requirement

a. Applicant Management

- Create accounts, submit applications and upload documents
- schedule and manage ~~applicar~~ appointments

b. Verification and Processing

- Documents verification by officers
- Police/background verification integration

c. Biometric Capture

- capture applicant's photograph, fingerprint and digital signature

d. Passport Issuance

- Manage printing and personalization
- update delivery status and notifi-cations

(4). Interface Requirement

a. User Interface

- Applicant portal for form submission and status tracking
- Back-office interface for verification and processing.

b. Integration Interface

- Integration with payment gateways, police verification systems and courier services.

(5). Performance Requirement

a. Response Time - Portal actions should respond with 2-3 seconds.

b. Scalability - Support at least 2000 concurrent users.

c. Data Integrity - Ensure consistency in applicant data across modules.

(6). Design constraints

a. hardware Limitation - Support biometric devices such as fingerprint scanners, cameras and printers.

b. Software Dependencies :- Secure database and workflow frameworks.

7. Non-functional Attributes

- Security :- Multi-factor authentication, encryption of sensitive data
- Reliability : Ensure minimal downtime. redundancy
- Portability : Access via web and mobile platforms.
- Usability : step-by-step guided application forms.

8. Preliminary Schedule and Budget.

The development of the Passport Automation System is estimated to take 7.7 months with a budget of $150,000. covering planning, development, Security compliance and deployment.

Thu 22/08/25