1) Hotel Management System

1.) Introduction :-

The Hotel management System (HMS) is a software solution designed to streamline hotel operations, including reservations, guest and reporting. This document defines requirement scope and deliverables. Development is estimated a take 6 months with a budget of $100,000.

1.1) Purpose

To provide a clear specification of HMS functionalities for developers and stakeholders

1.2) Scope

The system covers room booking, guest profiles, billing, reporting and integration with payment gateways and booking platform.

2) System Overview - Functional Requirements

HMS will improve efficiency for hotel staff and enhance guest experience. Its main features include:

- Reservation management
  - Online / front desk booking
  - Room status tracking
- Room management
  - Real-time assignment of rooms
- Guest management
  - Guest profiles
  - Smooth check-in / check-out

3) Interface Requirements

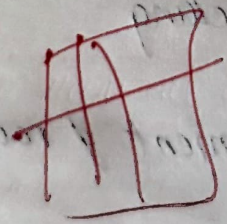
- User Interface : Intuitive, user-friendly, works on web, mobile and desktop
- Integration : Connects with payment gateways & third party booking sites.

4) Performance Requirement
- Ruponse time < 2 seconds.
- Handles at least 1000 concurrent users
- Ensure data accuracy and consistency

5) Design Constraints
Hardware : Compatible with standard hotel systems (computers, POS terminals)
Software : Uses relational DBMS (eg. MySQL), java and spring Boot.

6) Non-Functional Requirements
- security : Strong authentication
- Reliability : High availibility
- Scalability : supports growth in users
- Usability : Easy navigation for staff and guests
- Portability : Runs on multiple platforms.

7) Schedule and budget
- Timeline : 6months (planning → developement → testing → deployment)
- Budget : $100,000 covering all phases



2) Credit card processing System

1) Introduction

The Credit Card Processing System (CCPS) is a software application designed to securely handle credit card transactions. It supports payment authorization, settlement, fraud detection, and reporting. This document outlines the requirements, scope, and deliverables for the system. The estimated development effort is 6-9 months with a budget of approximately $150.000.

1.1) Purpose

The purpose of CCPS is to provide a fast, secure, and reliable platform for processing credit card transaction for banks, merchants, and customers.

1.2) Scope

The system covers transaction authorization, payment settlement, fraud detection, user account management and integration with external payment network (Visa, mastercard etc)

2) System Overview - functional Requirements

CCPS will facilitate smooth payment processing with strong security and compliance to industry standards. Key functions are :-

- Transaction management
  - Authorization, authentication
  - support and refunds

- User & merchant management
  - Maintain merchant profiles, transaction limits
  - Manage customer card info
- Fraud detection and security
  - Real time fraud detection
  - Multi-factor auth
- Billing and Reporting
  - Generate transactions statements
  - Provide financial reports

3) Interface Requirements

- User Interface: web-based dashboards for merchants, banks, simple interface for customers.
- Integration interfaces: APIs for connecting to banks payment networks, and third party financial systems.

4) Performance Requirements

- Response time < 2 seconds
- Ability to handle 10,000 concurrent users during peak periods
- 99.9% uptime with minimal downtime.

5) Design constraints

- Hardware: Runs on high-availability servers with redundant systems
- Software: RDBMS, secure backend.

6) Non-Functional Requirements

- Security : Strong encryption
- Reliability : Fault tolerant
- Scalability
- Usability : Intuitive dashboards
- Data Integrity : ensure accuracy of transaction.

7) Schedule & Budget

- Timeline : 6-8 months (planning → development → Testing → deployment)
- Budget : $150,000 including security certificates, infrastructure.

5) Library Management System:

1) Introduction

The library management system (LMS) is a software application designed to manage day-to-day library operations such as cataloging, book issuance/returns, member records, and user-friendly platform. Development is expected to take 5-6 months with a budget of $80,000

1.1) Purpose

To automate library function including book management, member services, and reporting, while reducing manual effort.

1.2) Scope

The system supports catalog management, borrowing, returning of books, digital resourse integration.

2) System Overview - Functional Requirements

Main Features:-

- Catalog management
  - store records
- member management
  - Register
  - renew, and manage members
- Circulation management
  - Borrow, return, reserve books
- Reporting
  - Transaction logs

3) Interface Requirements

- User interface:
  - Librarian dashboard
  - member portal
- Integration
  - Barcode/RFID
  - APIs for linking

4) Performance Requirements

- search results <2sec
- support 500+ concurrent users

5) Design constraints

- Hardware: PCs, barcode/RFID scanners
- Software: MySQL/Oracle database, java/Py backend.

6) Non-functional Requirements

- security
- Scalability
- Data integrity
- Portability
- Reusability

7) Schedule and budget

Timeline: 5-6 months (Weeks)

Budget: $80,000 (development, hardware integration, training

a) Stock Maintenance System

1) Introduction

Purpose

The stock maintenance system is an automated application built to automate stock / inventory tracking, updates and Reporting in an organization. should take 4-5months and around $70.000

1.1) Purpose

- Automate stock/inventory tracking, updates and reporting in an organization

1.2) Scope:

covers stock entry, issue, reorder levels, supplier details and reporting.

2) System overview - functional Requirements

Main features:-

- Stock management
  - Add, update, and delete stock items.
  - Categorize items by type, supplier, or usage
- Inventory tracking
  - Monitor current stock levels
  - Track items issued, returned
- Reorder management
  - maintain supplier details
  - compare supplies price and performance

- Reporting
  - Inventory status reports
  - Monthly/annual stock usage

5) Interface Requirements
- User Interface :
  - Admin Dashboard
  - Store staff
- Integration
  - Barcode/RFID support
  - API to integrate

4) Performance Requirements
- System must process stock updates in <2seconds
- Support at least 800 concurrent users
- Ensure data consistency

5) Design constraints
- Hardware : PCs, barcodes, printers
- Software : RDBMS, backend in java/.NET/Py

6) Non-functional Requirements.
- Security
- Reliability
- Usability
- Data integrity.

7) Schedule and Budget

- Timeline : 4-5 months
- Budget : $70,000 (development, integration, testing).

5) Passport Automation system.

1) Introduction

The passport Automation System (PAS) is an software application which aims the process of applying, verifying and issuing passports. Time may take upto 7 months

1.1) Purpose

To automate the process of applying, verifying and issuing passports.

1.2) Scope : Covers online application, document verification appointment scheduling, fee payment and status tracking.

2) System Overview - Functional requirements

Main features:

- Application management
  - Online submission of passport applications
  - Upload supporting document

Verification & Approval

- Automated document validation
- Integration with govt. databases

Appointment Scheduling

- Book/reschedule appointments
- Slot allocation

Payment & billing

- Online payment gateway
- Receipts & invoices

Status tracking
- Real-time tracking of application progress
- Notification via SMS/email

Reporting
- Generate application statistics
- Track pending

3) Interface requirement

User interface:
- citizen portal for application
- Admin portal for staff

Integration:
- APIs to connect with verification portals
- Admin portal integration with online payment gateways

4) Performance Requirements
- process each application step in <3seconds
- Handle at least 5000 concurrent users
- Ensure real-time data consistency

5) Design constraints
- Hardware: Runs on secure govt servers. with backup systems.
- Software: Database (Oracle/PostgreSQL), backend in Java/.NET, secure
- govt security and data privacy laws

6) Non-functional Requirement

- security : strong encryption
- Reliability
- Scalability
- Compliance
- Data Integrity

7) Schedule & Budget

Timeline : 6-7 months (planning → development → testing → deployment)

Budget : ~ $120,000 (software, security, infrastructure, training).