

HOTEL MANAGEMENT SYSTEM

Problem Statement:

Managing hotel operations manually often leads to inefficiency such as double booking of rooms, delayed check in / check out processes, inaccurate billing and poor customer experience. With increasing competition in the hospitality industry, hotels need a centralized system to streamline core processes like room booking, billing, staff scheduling, reporting etc. A hotel management system (HMS) is required to reduce operational workload, improve data accuracy and provide a seamless experience to customers.

1. Introduction

1.1. Purpose of Document

The purpose of this document is to outline the requirements for the hotel management system. It defines the system's functionality, scope, interface & constraints to ensure smooth development and deployment.

1.2. Scope of this Document

This HMS will automate all major hotel operations including room booking, guest registration, billing, staff management and report generation. It will improve operational efficiency.

1.3 Overview

The HMS is a centralized platform for managing hotel operations. It allows customers to book rooms

online, while hotel staff can manage room availability, payments etc. efficiently.

2. General Description

The HMS is designed for hotel administrators, reception staff & customers. Features include room booking, guest maintenance, billing & reporting. The system will be easy to use, provide role-based access and ensure secure data handling.

3. Functional Requirements

- Room Booking Management: CRUD operations
- Guest Management: Registers guests, check-in/check-out and maintain guest history
- Billing and Payments: Generate bills, apply discounts
- Staff Management: Assign tasks, manage schedule
- Reporting: Generate daily, weekly & monthly reports

4. Interface Requirements

- UI: Web portal and mobile app interface
- System Interface: Integration with payment gateway, external booking sites
- Data Interfaces: Secure database for storing guest details, room information & financial records.

5. Performance Requirements

- System should handle at least 500 concurrent users
- Transaction processing within 2s.
- System uptime 99.5%.

6. Design Constraints

- must comply with data privacy laws
- should support multi-language & multi-currency
- must ensure data backup every 24 hours.

7. Non-Functional Attributes

- Security: Role based auth. & data encryption.
- Reliability: Fault tolerant architecture with backup.
- Scalability: Support future expansion
- Usability: User friendly interface.

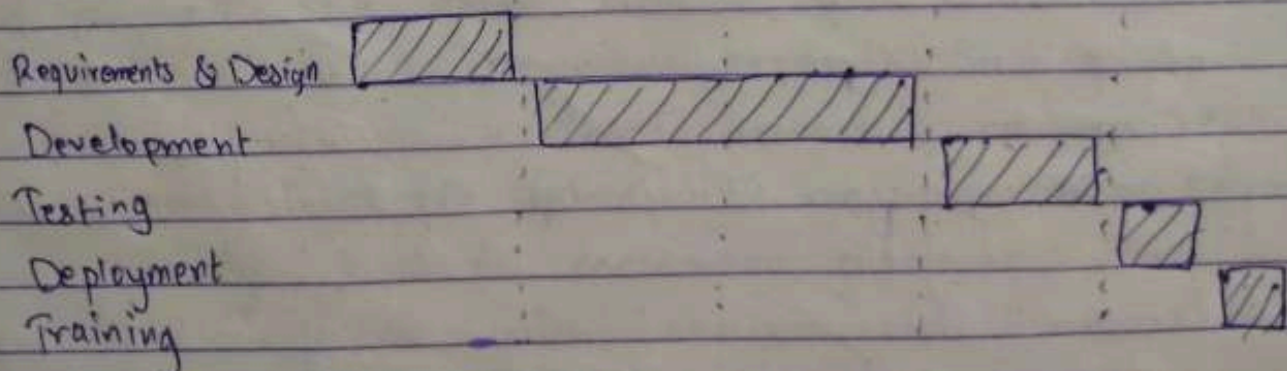
8. Preliminary Schedule and Budget

The project is estimated to take 5 months for full development.

- Requirements & Design: 1 month
 - Development & Testing: 3 month
 - Deployment & Training: 1 month
- Estimated budget: ~~\$25,000~~ - \$30,000

GIANTT CHART

Month:	1	2	3	4	5
Week:	1 2 3 4	5 6 7 8	9 10 11 12	13 14 15 16	17 18 19 20



CREDIT CARD PROCESSING SYSTEM

Problem Statement

With the growing use of online and in-store payments, there is a need for a secure and reliable credit card processing system. Manual handling of payments is slow and prone to errors, while existing systems often face issues like fraud, delays or poor user experience. A well-designed system is required to validate transactions, process payments securely and provide fast approvals while protecting sensitive cardholder data.

1. Introduction

1.1. Purpose of this Document

The purpose of this document is to define the requirements for the credit card processing system (CCPS) outlining its functionality, constraints, interface & performance expectations. It serves as a guideline for developers, testers & stakeholders.

1.2. Scope of this Document

The CCPS will allow merchants to process customer payments via credit cards. It will validate card details, authorize payments through issuing banks, transfer funds to merchant accounts, and maintain transaction logs. The system will integrate with payment gateways, banks & merchant platforms while ensuring compliance with security & financial regulations.

1.3 Overview

The system will act as a secure intermediary between merchants, customers & banks. It will handle transactions in real-time, provide reporting dashboards to merchants & include fraud detection.

2. General Description

The ccps is intended for use by:

- Merchants : To receive payments
- Customers : To pay
- Banks : To authorize & settle transactions.

3. Functional Requirements

- Card Validation : validate card no, cvv & expiry
- Authorization: Route trans. req. to issuing bank for approval
- Fraud detection: AI based monitoring
- Reporting: Generate transaction, settlement & fraud reports

4. Interface Requirements

- User Interface: Web based portal for merchants & admin
- System Interfaces: APIs for integration with bank, payment gateways etc.
- Data Interfaces: Secure database for storing logs

5. Performance Requirements

- Must process at least 1000 transaction / second
- Transaction approval response time must be under 2s
- System uptime should be 99.9% annually.

6. Design Constraints

- Must comply with PCI-DSS security standards
- Should support multiple currencies

- most operate on ~~high~~ redundant servers

7. Non-functional Attributes

- Security: End-to-end encryption, tokenization, multi-factor authentication for admin access
- Reliability: Redundant architecture with disaster recovery.
- Scalability: Able to handle peak loads.
- Portability: Should integrate easily with all device.

8. Preliminary Schedule & Budget.

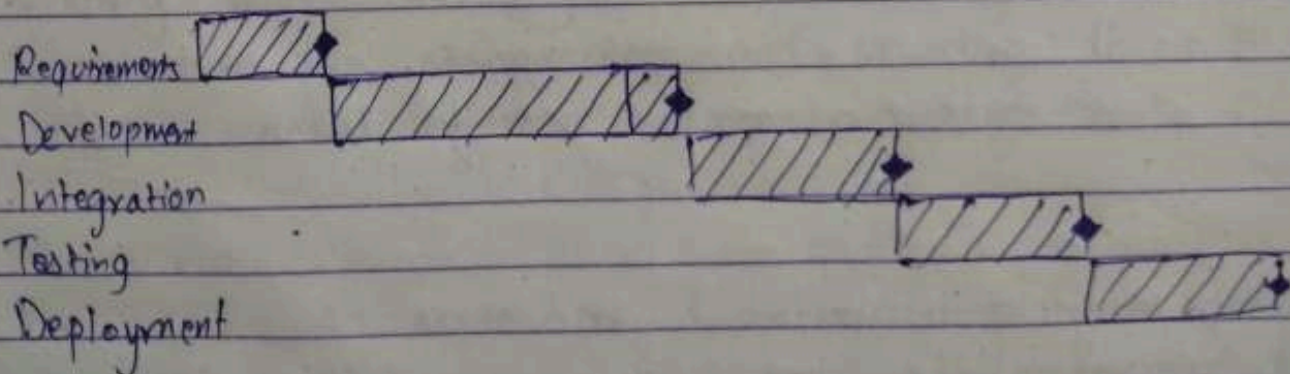
The project is estimated to take 6 months

- Requirements & architecture: 1 months
- Development & Integration: 3 months
- Testing: 1 months
- Deployment: 1 months

Estimated Budget: \$50,000 - \$70,000

GAANT CHART

Month:	1	2	3	4	5	6
Week:	1 2 3 4	5 6 7 8	9 10 11 12	13 14 15 16	17 18 19 20	21 22 23 24



Problem Statement

Traditional library management system relies heavily on manual processing such as maintaining physical records of books, manually issuing / returning books, and tracking due dates. This leads to inefficiencies like misplaced books, inaccurate inventory, delayed book returns & difficulty in generating reports. A Library Management System (LMS) is required to automate the core functions of a library.

1. Introduction

1.1. Purpose of this Document

The purpose of this document is to define the requirements for the Library Management System (LMS). It specifies the features, functional and non-functional requirements, constraints and performance expectations to guide development.

1.2. Scope of this Document

The LMS will automate all library operations including book cataloging, member management, book issue/return, file management and reporting. It will reduce manual effort, increase accuracy.

1.3. Overview

The LMS will serve as a centralized platform for librarians, library members and administrators. It will provide role-based access, secure data handling & support for both physical & digital resource management.

2. General Description

The system will be used by:

- Librarians: To manage books, members & generate reports.
- Library members: To search, reserve, borrow & return book.
- System: To track book inventory, calculate fines & send alerts.

3. Functional Requirements

- Book management: Add, update, Delete, categorize
- Member management: Register new member, manage
- Book Issue / Return
- Search & reservation
- Fine calculation

4. Interface Requirements

- User Interface: web based portal
- System Interface: Integration with barcode / RFID scanner
- Data Interface: Centralized Database

5. Performance Requirements

- Support upto 500 concurrent users
- Process search queries in under 2 secs.
- Handle upto 100,000 books
- Maintain system uptime of 98.5%.

6. Design Constraints

- Must support barcode / RFID
- must operate on commonly used browsers
- must comply with data protection

7. Non-Functional Attributes

- Security: Secure login, Data encryption
- Reliability: Backup & recovery mechanism
- Scalability: ability to handle growing no. of users & books
- Usability: Easy user interface.

8. Preliminary Schedule and Budget

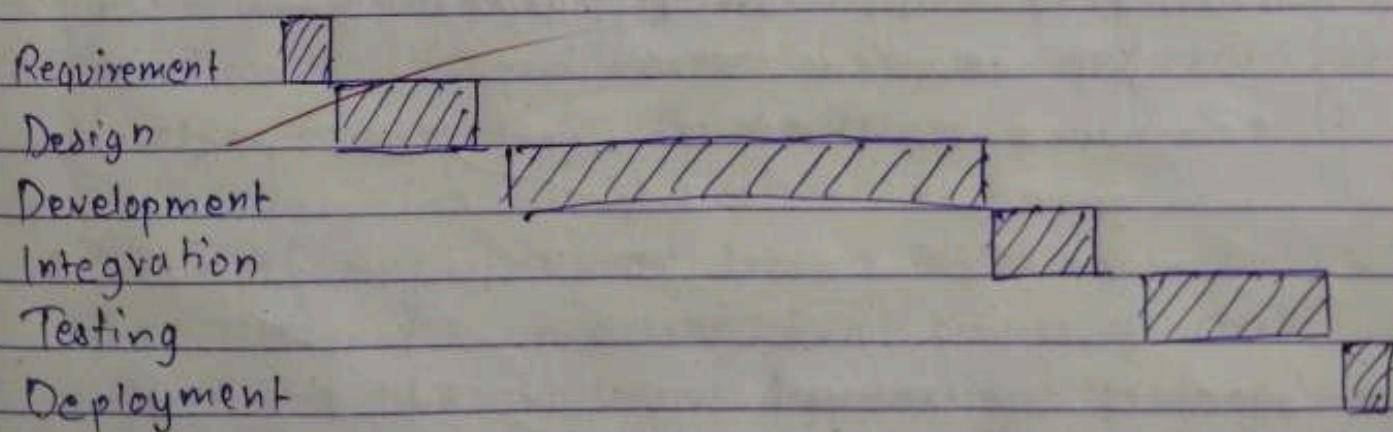
Estimated project duration of 4 months

- Requirements & design: 1 month
- Development & Integration: 2 month
- Testing & Deployment: 1 month

Estimated Budget: \$15,000 - \$20,000

GANIT CHART

Month:	1	2	3	4
Week:	1 2 3 4	5 6 7 8	9 10 11 12	13 14 15 16



STOCK MAINTAINANCE SYSTEM

Problem Statement

Business often face challenges in managing their inventory efficiently, resulting in overstocking, stockouts, mismanagement of resources and revenue loss. Traditional manual stock management methods - using spreadsheets or paper based logs - are time consuming, error-prone and lack real-time visibility into inventory levels. A stock maintainance system (SMS) is required to automate inventory tracking, provide real time stock updates, generate purchase and sales records and ensure optimal stock levels.

1. Introduction

1.1. Purpose of Document

The purpose of this document is to define the requirements for SMS including its features, scope, functional requirements, constraints, & performance benchmarks. It will serve as a reference for developers, testers & stakeholders.

1.2. Scope of this Document

The SMS will automate the process of stock management for businesses by tracking inventory levels, managing stock inflow/outflow & generating reports. It will enhance operational efficiency, reduce human errors, and optimize stock levels to prevent shortages or excess inventory.

1.3 Overview

The system will provide a centralized platform accessible by administrators, warehouse staff, and management. It will feature a dashboard displaying real-time stock data, automated alerts for low stock, and integration with external sales/purchase systems.

2. General Description

The SMS will create to:

- Admins / Managers: To monitor inventory, generate reports & configure system settings.
- Warehouse staffs: To update stock in/out transactions.
- System: To send alerts, maintain stock logs, & provide analytics.

3. Functional Requirements

- Stock Entry & Update: Add, edit, delete.
- Stock Transactions: Record stock inflow & outflow.
- Threshold Alert: Notify managers when stock levels fall below minimum limits.
- Reporting & Analytics: Generate inventory valuation, movement history & usage trends.
- Barcode / RFID support.

4. Interface Requirements

- User Interface: Web and mobile dashboards.
- System Interface: Integration with barcode/RFID scanner.
- Data Interfaces: Secure database to store product details, stock levels.

5. Performance Requirements

- Handle at least 60,000 stock items
- Support 100 concurrent users
- Maintain 99.7% system uptime

6. Design Constraints

- must ensure data accuracy & prevent duplicate stock entries
- must comply with business data protection standards
- should support offline mode.

7. Non-functional attributes

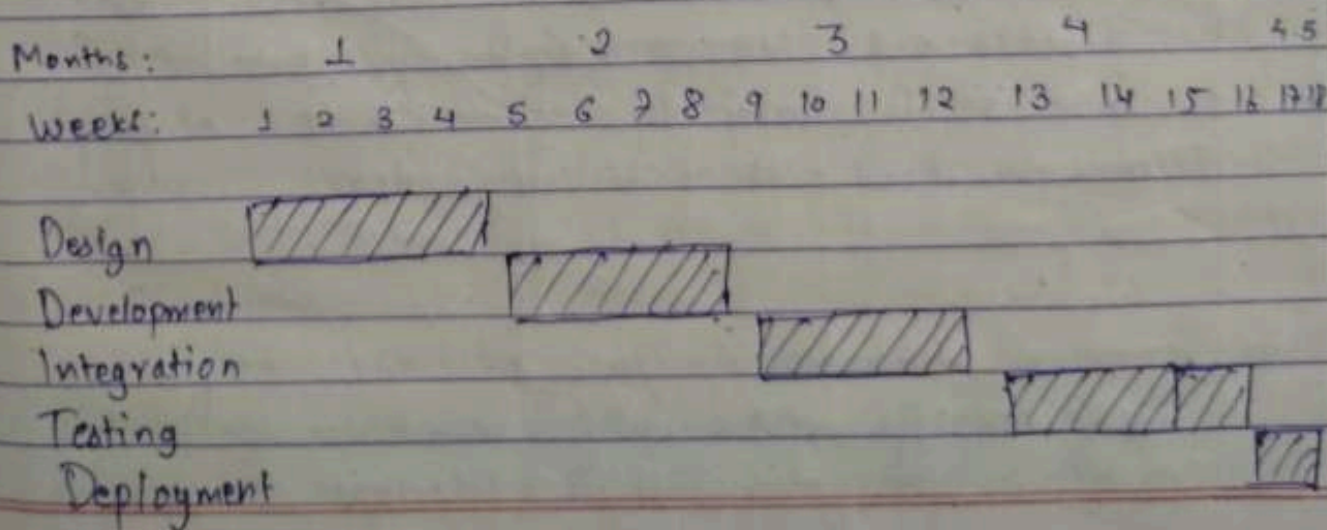
- Security: Role-based auth., encrypt data
- Reliability: Regular backups & failover support
- Scalability: ability to support multiple warehouse
- Usability: Intuitive UI with minimal training

8. Preliminary Schedule & Budget

Estimated timeline: 4.5 months

Estimated Budget: \$18000 - \$20000, including development, hardware integration & support.

GANTT CHART



PASSPORT AUTOMATION SYSTEM

Problem Statement:

Traditional passport application & issuance processes involve excessive paperwork, manual verification, long processing times, & inefficiencies in tracking application status. Applicants often face delays due to missing documents, lack of transparency and manual errors in data entry. A passport management system (PMS) is required to automate the entire passport lifecycle - from application, verification, delivery tracking etc.

1. Introduction

1.1. Purpose of this Document

This document defines the requirements for the passport management system (PMS), specifying its functionalities, constraints, performance standards and interfaces. It will serve as guide for developers, testers & stakeholders.

1.2. Scope of this Document

The PMS will streamline passport application & issuance by digitizing all processes. It will provide an online platform for citizens to apply, track applications & book appointments.

1.3. Overview

The system will be web-based and mobile-friendly supporting applicants, government officials & law enforcement agencies. It will integrate with external

systems for background verification.

2. General Description

The PMS will be used by:

- Applicants: submit form, book appointment, track
- Government officials: To review application, verify documents, approve/reject
- Law Enforcement Agencies: To conduct police verification

3. Functional Requirements

- Online application submission
- Appointment Scheduling
- Document & Identity verification
- Police Verification Integration
- Reports & analytics
- Status notifications

4. Interface Requirements

- User Interface: Web portal & mobile app
- System Interface: Integration with national ID database, police verification systems.
- Data Interface: Secure centralized database

5. Performance Requirements

- Support upto 10,000 concurrent users
- Application submission & status updates must process within 2 seconds.
- Uptime of 99.9% annually.

6. Design Constraints

- must comply with data protection laws
- must include biometric authentication
- must support multilingual interface

7. Non-functional Attributes

- Security: End to End encryption, biometrics
- Reliability: Redundant servers, audit trails
- Scalability: Support future expansion
- Usability: Simple user interface

8. Preliminary Schedule and Budget

Estimated Duration: 7 months

- Requirements & system architecture: 1.5 months
- Development & Integration: 3.5 months
- Testing: 1 month
- Deployment & Training: 1 month

Estimated Budget: \$80,000 - \$100,000, including infrastructure & biometric integration.

GANTT CHART:

Months: 1 2 3 4 5 6 7
Weeks: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

