be considered

1) Hotel Management System.

Problem statement

The Hotel Industry requires efficient and streamlined processes for managing room reservations, customer check-ins / check-outs, billing, and reporting. Currently, many hotels rely on manual systems that lead to errors, delays, and inefficiencies. The goal of this is to develop an automated solution that integrates key operations, such as reservation management, customer data Handling, payment processing, and reporting into one user-friendly software system.

1 Introduction

1.1 Purpose of this document:

This document defines the requirements of the Hotel Management Systems, explaining its purpose to automate hotel operations such as booking, billing and reporting.

1.2 Scope of this document

The system improves efficiency, reduces manual errors, and provides value through integrated management of hotel activities.

1.3 Overview

The product is a centralized hotel management solution for reservations, customer management, billing and reporting.

2. General description

The system allows staff and customers to interact with hotel services, improving efficiency in reservations, billing and reporting while ensuring ease of use.

3. Functional Requirements

- The system allows room booking, cancellations, check-in/check-out, customer record management and automated Bill Generation.

4. Interface Requirements

The system provides a web interface for customers, an admin dashboard for staff, and integrates with payment gateways and databases.

5. Performance requirements

The system should support 100+ concurrent users, respond within 2 seconds, and maintain secure, accurate transactions.

6. Design Constraints

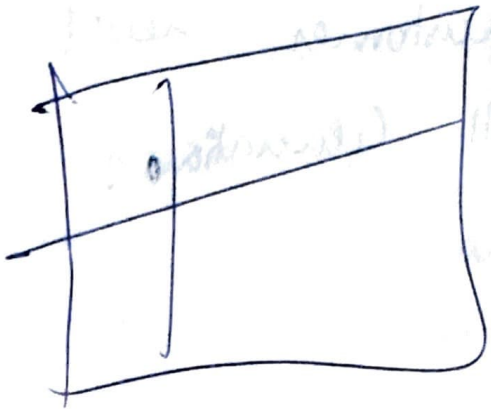
The software will be web-based, use relational databases, and comply with industry security standards for payments and data protection.

7. Non-Functional Attributes

The system must be reliable, secure, portable, and scalable, ensuring data integrity and easy maintainability.

8. Preliminary schedule and Budget

The project will be developed in about 5 months with an estimated budget of $20,000 including development, testing, and deployment.



Table

Cost Estimate

2) Credit Card Processing.

Problem statement

Credit card transactions require secure, efficient and real-time processing to prevent fraud and ensure smooth customer experiences. Manual / outdated systems often cause delays, errors.

1. Introduction

1.1 Purpose

→ Define requirements for a secure and efficient credit card transaction system.

1.2 Scope

→ Process payments, validate transactions, prevent fraud, and generate reports.

1.3 Overview

→ Provides real-time processing, integration with banks, and secure user authentication.

2. General description

→ Enables merchants, banks and customers to interact in a seamless payment cycle.

→ Ensures fast transaction approval, rejection of suspicious activities.

3. Functional requirements

→ Validate card details and authenticate transactions.

→ Process payments and update user/Bank records.

→ Generate invoices and transaction history.

→ Provide fraud detection alerts and reports.

4. Interface requirements

→ Merchant portal for payment requests.
→ Bank API integration for validation.
→ Customer interface for transaction confirmation.

5. Performance requirements

→ Handle 500+ transactions per second.
→ Transaction response within 2-3 seconds.
→ Maximum error rate ≤ 0.1%.

6. Design constraints

→ Must comply with PCI-DSS security standards.
→ Encryption for all sensitive data.
→ High availability

7. Non-FR

→ Security: strong encryption
→ Reliability: 99.9% up time
→ scalability: support increasing transaction loads.
→ Portability: Work accross platforms and devices.

8. Preliminary schedule & budget

Component	Cost	Time
1. Requirement Analysis	$3000	2-weeks
2. Design	$4000	3-weeks
3. Development	$12000	8-weeks
4. Security Integration	$5000	—
5. Testing	$4000	4 weeks
6. Deployment	$2000	2 weeks
7. Maintainence	$3000	—
	$33000	

3. Library Management System

Problem statement

Manual library often leads to misplaced records, delays in returning books, and difficulty in tracking availability.

1. Introduction

1.1 Purpose
→ Define requirements for automating book management, user records, and transactions.

1.2 Scope
→ Enable book cataloging, issue tracking, fine calculation, and reporting.

1.3 Overview
→ System ensures efficient and accurate library operations with easy access to information.

2. General Description
→ Manage books, users, borrowing history.
→ Provides search, reservation, and fine tracking.

3. Functional Requirements
→ User portal for search.
→ Add, update, delete books.
→ Issue, books and calculate fines.
→ Generate reports on usage and inventory.

4. Interface requirements
→ User portal for search and book requests.
→ Librarian dashboard for transactions.
→ Database integration for record storage.

5. Performance Requirements
→ Handle 50+ concurrent users.

6. Design Constraints
→ Must work on local network and web.
→ Use relational database

7. Non Functional Attributes
Security: Role-based access.
Reliability: 99% uptime
Usability: Simple and intuitive UI.
Scalability: Support large book databases.

8. Preliminary schedule and budget.

Component	Cost	weeks
1). Requirement Analysis	$1000	1
2) Design	$2000	2
3) Development	$7,000	5
4) Testing	$2,000	2
5) Deployment	$1,000	1
6) Maintenance	$2000	
	$15,000	

4. Stock Maintainence System

Problem Statement

Manual stock management often results in errors, delays and difficulties in tracking inventory levels. A Stock Maintainance system is required to automate stock entry, monitoring and updating, ensuring accuracy, real-time updates, and efficient control of goods.

SRS

1. Introduction
   - 1.1 Purpose - Automate stock tracking, reduce errors, and improve efficiency in inventory management.
   - 1.2 Scope - Manage stock entries, updates, sales, purchases, and generate real-time reports.
   - 1.3 Overview - Provides centralized control over inventory with alerts for shortages and overstock.

2. General Description
   - → Maintains records of stock inflow, outflow, and balances.
   - → Helps businesses monitor inventory levels, reorder needs and trends.

3. Functional Requirements
   - → Add, update, and delete stock items.
   - → Track incoming and outgoing stock.
   - → Generate low-stock alerts and reorder notifications.
   - → Provide inventory and financial reports.

4. Interface Requirements
   - → Admin dashboard for stock management.
   - → User interface for sales/purchase entry.
   - → Database for storing inventory records.

5. Performance Requirements
   - → Handle 200+ transactions per day.
   - → System should update inventory in real-time within 2 secs.

6. Design Constraints
→ Relational database required.
→ Should integrate with accounting and billing software.

7. NFR
→ Security - Role-based access.
→ Reliability - High availability with back-up system.
→ Scalability - Support for multiple warehouse.
→ Usability - Simple UI for non-technical staff.

8. Preliminary schedule and Budget

Component	Cost (USD)	week
1. Requirement Analysis	$1,500	1
2. Design	$2,500	2
3. Development	$8,000	6
4. Testing	$2,000	2
5. Deployment	$1,500	1
6. Maintainence	$2,000	
	$16,000	12

5. Passport Automation System

Problem statement

Manual passport processing is often slow, error-prone and inconvenient for applicants. A passport Automation System is needed to streamline applications, verifications and issue processes, ensuring faster approvals, transparency and secure data handling.

SRS

1. Introduction
1.1- Purpose : Automate passport application, verification and issue processes.
1.2 Scope : Handle applications, track status, integrate with verification authorities, and generate passports.
1.3 Overview: Provides online access to applications and an admin panel for authorities.

2. General Description
→ Manages applicant records, documents, and verification steps.
→ Improves efficiency by reducing manual paperwork and delays.

3. FR
→ Register new passport applications online.
→ Upload and verify documents.
→ Track application status in real-time

4. Interface Requirements
→ Handle 1000+ concurrent applications.
→ Response time for transactions ≤ 3 seconds.

5. Performance Requirements
→ Applicant portal for online applications and status tracking.
→ Admin portal for verification and approvals.

6. Design Constraints
→ Must comply with government security standards.
→ Biometric data storage requires encryption.

7. NFR
→ Security: High-level encryption, secure authentication.
→ Reliability: 99.99% uptime for continous service.
→ Usability: Simple web interface for applicants.
→ Scalability: Support nationwide applications.

8. Preliminary schedule & Budget.

Component	Cost.	weeks
1. Requirement Analysis	$3,000	2
2. Design	$5,000	3
3. Development	$15,000	8
4. Security Integration	$6,000	1
5. Testing	$4,000	4
6. Deployment	$3,000	2
7. Maintainence	$5,000	
	$41,000	20