Hotel Management System

Problem statement :- Manual hotel booking is inefficient & error prone, an automated system is needed.

1. Introduction

1.1 Purpose

Defines requirements for the Hotel Management System (HMS) to automate hotel room booking, customer management & reporting. Add food

1.2 Scope

* Customers : Register, log in, book/cancel rooms, pay online.
* Receptionist : Manage bookings, respond to inquiries.
* Manager : View reports, update room & financial details.

1.3 Overview

System ensures secure transactions, fast booking & simple web based access. Meeting hall, best facilities gym, Spa pool

2. General Description

* Customers : Book, cancel, and pay for rooms.
* Receptionists : Handle booking updates & customer queries.
* Managers : Generate reports & update room details.

UI

3. Functional Requirements

1. Customer registration & login
2. Room booking / cancellation
3. Online payment

4. Receptionist manages bookings

5. Manager generates reports

4. Interface Requirements

* UI : Simple web browser interface

* Software : Java (J2EE), MongoDB, Apache Tomcat, HTML/JS

5- Performance Requirements

* Data update : ≤ 2 sec

* Query response : ≤ 5 sec

* Login validation : ≤ 3sec

6. Design Constraints

* Memory : 10 GB server

* Language : English only

* Tech : Java - based, Windows OS

7. Non Functional Requirements

* Security : HTTPS for payments

* Reliability : Hourly backup, recovery within 1hr

* Capacity : 10,000 users, 20 real transactions

* Usability : Easy UI, user manual provided.

8. Schedule & Budget

* Development time : 4-6 months

* Budget : Limited

Credit Card Processing

Problem Statement :

Manual credit card processing is slow & insecure, an automated system is needed for fast & safe transactions.

1. Introduction

1.1 Purpose

This document defines the requirements for the credit card processing system (CCPS).
It aims to provide secure, fast & reliable credit card transactions for customers & merchants.

1.2 Scope

* Customers : Make payments securly.
* Merchants : Accept payment & verify transactions.
* System : Validate cards, process payments & generate transaction reports.

1.3 Overview

Manual credit card is slow & insecure, an automated system is needed for fast & safe transactions.

2. General Description

* Customers : Use credit cards for payments
* Merchants : Receive confirmation of transactions.
* Admin / Bank : Manage, monitor & audit transactions.

3. Functional Requirements

1. Validate credit card details
2. Authorize transactions
3. Process payments securly
4. Generate recepits & transaction history
5. Admin reporting & monitoring

4. Interface Requirements.

* UI : Simple & secure web interface
* Software : Java / Python backend, MySQL.

5. Performance Requirements

* Transaction approval ≤ 5 sec
* Data validation ≤ 2 sec
* System uptime 99.9%.

6. Design Constraints

* Must comply with PCI-DSS security standards.
* Encryption required
* Limited budget

7. Non Functional Attributes

* Security : SSL/TLS encryption
* Reliability : Backup every hour

8. Schedule & Budget

* Development time : 6 months => 26 weeks.
* Budget : Limited.

Cost Estimation

1) Development - ₹3 - ₹4 lakhs

2) Software/Tools - ₹50,000

3) Hosting & servers - ₹50,000 - ₹70,000

4) Compliance & security - ₹1 lakh

5) Maintenance - ₹50000

Total ₹5 - ₹6 lakhs

Library Management System

Problem Statement :

Most libraries still rely on manual systems for managing books, members, and transactions. This leads to frequent errors, data loss, time consuming & difficulty in tracking issued/returned books and fines.

1. Introduction

1.1 Purpose :

To define the software requirements for automating library operations like book management, user handling, issuing, returning & fine tracking.

1.2. Scope

The system manages books, members, transactions, fines & reporting with minimal manual work.

1.3 Overview

Provides a web based platform for librarians and members to manage & access library services efficiently.

2. General Description

LMS allows librarians to add/manage books & users; issue & return books & generate reports.

3- Functional Requirements

* Add / update / delete books & users
* Issue / return books
* Book search by title
* Generate reports.

4- Interface Requirement

* Web - based user interface
* Backend database
* Optional [barcode scanners]

5- Performance Requirement

* < 2 seconds response time
* Handle 100+ users
* Error rate < 1%.

6- Design Constraints.

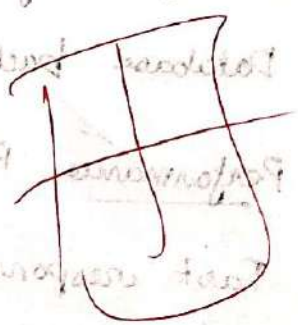
* Platform independent
* Web based only.

7- Non Functional Attributes

* Security
* Reliability
* Scalability
* Data integrity.

8) Preliminary Schedule

* Time : 3 monthes
* Cost : $10,000 - $15,000



Stock Maintenance System

Problem Statement

Manual inventory management causes stock losses and inefficiency; a stock Maintenance System ensures accurate tracking & better control

1) Introduction

* **Purpose :**

  Automate stock tracking, updates, & reporting.

* **Scope :**

  Manage inventory levels, track stock movement and notify low stock.

2) Functional Requirement

* Add / update / delete stock items.
* Track incoming & outgoing stock.
* Alert for low stock.
* Generate reports.

3) Interface Requirement

* User friendly web interface
* Database backend

4) Performance Requirement

* Fast response < 2sec
* Support 50+ concurrent users
* Handle 10,000+

5) Design Constraints

* Use open source tech
* Cross platform support
* Secure user authentication

6) Non Funtional Requirement

* Secure access
* Reliable & scalable
* Maintain data accuracy.

7) Schedule & Budget

* Time : 2 - 3 months
* Cost : $ 8000 - $ 12,000

Problem Statement

Manual passport processing is slow & error prone automation speeds up application & improves accuracy

1) Introduction

Purpose : Automate passport application, verification & issuance.

Scope : Manage application, track status, verify documents & generate reports.

2) Functional Requirements.

* Submit, update & track passport
* Verify applicant documents.
* Approve or reject applications.

3) Interface Requirements

* Web based portal
* Secure database
* Integration with govt ID

4) Performance Requirement

* Fast response < 2 sec.
* Handle applications
* Low error rate

5) Design Constraint

* use secure technology
* Accessible
* Role based user access.

6) Non - Functional Requirement

* High security
* Reliable
* Scalable

7) Schedule & Budget

* Time :- 3-4 months
* Cost : $12,000 - $18,000