

1. Hotel Management System :

• Problem Statement :

Hotels often face challenges in managing bookings, check-ins, check-outs, & billing when done manually. This results in errors like double booking, loss of records, and customer dissatisfaction.

The problem is the absence of an efficient, automated system that can handle hotel operations smoothly.

• Introduction :

> Purpose of this Document :

The purpose of this SRS document is to describe the Hotel Management in detail. It defines the system's objective, requirements, and functionality, which will help the development team and also act as a reference for testing.

> Scope of this Document :

The Hotel Management System will automate tasks such as reservation, customer check-in / check-out, staff management, billing and ensures customer satisfaction.

> Overview :

The HMS is a user-friendly application that handles hotel activities. It consists of modules for room booking, billing, customer records, and reporting. The system reduces errors and increases efficiency for hotel management.

• General Description :

The HMS is meant for receptionists and administration. It provides features such as viewing available rooms, managing staff, handling payments and increases efficiency for hotel management.

- Functional Requirement
 - 1. Customer Registration
 - 2. Room Booking and cancellation
 - 3. Check-in and check-out Processing
 - 4. Billing and Payment
 - 5. Staff Management.
 - 6. Report Generation.

- Interface Requirements

The system will provide a GUI for users with options like booking, billing, and staff management. It will also include a backend database [MySQL] for storing records.

- Performance Requirement

The HMS should process bookings quickly and allow multiple users at the same time. It should handle upto 5000 records efficiently and responses should be under 2 seconds.

- Design Constraints

1. Developed using Java / Python and MySQL
2. Should meet data security standards for customer details.
3. Should run on Windows and Linux OS

- Non-functional Attributes

1. Security : Protect customer data with encryption
2. Reliability : Should be available 24/7 with 99% uptime
3. Portability : Must support hotel expansions.
4. Usability : Easy for staff to use.
5. Scalability : Must support expansion

* Preliminary Schedule and Budget

Development is estimated to take 4 months with a budget of \$ 20,000. This includes design, coding, testing and deployment.

2. Credit Card System

* Problem Statement

In today's world, most payments causes delays, errors, and sometimes fraud. Customers face problem problems like declined transaction even when they have sufficient balance, while banks and merchants face the risk of fraudulent activities. There is need of a software system that can process credit card transactions automatically.

* Introduction

> Purpose this Document

The purpose of this document is to describe the credit card Processing System and tasks holden.

> Scope of this Document:

The system automates credit card transactions, improving speed, reliability and security while reducing manual work

> Overview:

The system involves customers, merchant and banks. Customer module for authentication and reporting.

* General Description:

The system involves customers, merchant and banks. Customer use credit cards for payment, merchant initiate transaction and banks authorize and settle them. It ensures quick transactions.

- Functional Requirements:

1. Customer Authentication
2. Transaction Authorization
3. Fraud Detection and Prevention
4. Billing and Settlement
5. Transaction History Management.
6. Report Generation for banks and merchants

- Interface Requirement

A secure web interface for merchants and admins. Customers authenticate and integration with banking servers occurs via secure APIs

- Performance Requirements:

1. handle thousands of transactions per second
2. authorization response time under 1 second
3. 99.9% uptime

- Design Constraints:

1. compliance with PCI-DSS security standards
2. use of AES and RSA encryption
3. integration with various banks' existing servers

- Non-Functional Attributes:

1. Security
2. Reliability
3. Scalability
4. Usability
5. Compatibility

- Preliminary Schedule and Budget.

- > Development duration of 6 months with a budget of \$50,000 including requirement gathering, design, implementation, security testing and deployment.

- 3. Library Management System.

- Introduction

- > Purpose of this Document

Detail the LMS objectives, requirements and features to guide development

- > Scope of this Document

Automate book cataloging, member registration, issue / platform, fine calculation and reporting for students, librarians, and administration.

- > Overview:

maintain a central database of books, members, and transaction with search features for availability and account access enhancing speed and reliability.

- General Description:

Users include students, librarians and administrators

- Functional Requirement

1. Member registration and management.

2. Book Catalog management

3. Book Issue and Return Management.

4. Fine Calculation for late returns.

5. Search Facility by title, author, or category

6. Report Generation

- Interface Requirements

Graphical User Interface for librarians and students, student login capability and database backend (e.g. MySQL) for record storage

- Performance Requirement

1. handle upto to 10000 book records efficiently
2. process transaction under 2 seconds
3. Support multiple concurrent user

- Design constraints

1. Developed using Java or Python with relational database
2. Basic security rules for student and book data
3. compatible with windows and Linux system.

- Non-functional Attribute

1. Security
2. Reliability
3. Scalability
4. Usability
5. Portability

- Preliminary Schedule and Budget.

- > Development estimated at 3 months with a budget of \$15,000 including all development and deployment activities.

4. Stock Maintenance System

- Introduction

- Purpose of this Document

Define req. and functionalities for effective inventory management.

> scope of this document

automate stock management for shops, warehouses and organizations

> Overview

Maintain accurate, real-time stock information, supporting purchases, sales, alerts and reporting for managers, employees, and administration.

• General Description

Users include manager, employee, and administration.

• Functional Requirement

1. Item Registration
2. Purchase Entry and stock update
3. Sales Entry and stock reduction
4. Low Stock Alert System
5. Supplier and customer management
6. Report Generation.

• Interface Requirement

GUI for stock management, MySQL database backend, separate access levels for managers and employees

• ~~Performance Requirements~~

1. Support up to 201000 stock items
2. process updates with 2 secs
3. Multi-user simultaneous access

• Design constraints

1. Developed using Java/Python with MySQL
2. Security measures for inventory protection
3. Runs on Windows / Linux desktop computers

- Non-Functional Attributes

1. Security
2. Reliability
3. Scalability
4. Usability
5. Portability

- Preliminary Schedule and Budget

Development planned for 4 months with \$18,000 budget including analysis, coding, testing and deployment phases.

- 5. Passport Automation System

- Introduction

- > Purpose of this Document

Describe requirements to digitize passport application and approval process.

- > Scope of this Document

Online submission, document upload, payment, processing, verification, status tracking, appointment scheduling, and reporting for application officers and administrators.

- > Overview

Modules include applicants, passport officers & administration and administrative reports to increase speed, security and transparency

- General Description

User includes applicants, passport officers, and administrators.

- Functional Requirements

1. Online Application Submissions with document upload
2. Payment Gateway for fees
3. Verification Module
4. Status Tracking and Notifications
5. Appointment Scheduling
6. Report Generation

- Interface Requirements

Web interface for applicants, secure portal for officers, database backend for data and documents

- Performance Requirements

1. Handle thousands of applicants daily
2. Response within 2 seconds
3. 24/7 availability

- Design Constraints

1. Compliance with Government policies
2. Data encryption for security
3. Integration with government ID.

- Non-functional Attributes

1. Security: Encrypt sensitive data
2. Reliability: 99.9% uptime
3. Scalability: support increasing users
4. Usability: Easy for all applicants
5. Portability: compatible across browsers and platforms

- Preliminary Schedule and Budget

6 months development period with a \$60,000 budget covering all stages including testing and deployment.



~~OK~~
298