

SRS Document for

1) Hotel Management System

1. Introduction

1.1 purpose of this document

The purpose of this document is to outline the requirements for the Hotel management system (HMS). It serves as a guide for stakeholders, developers, and project managers to ensure a clear understanding of the system's objectives, functionalities and constraints.

1.2 scope of this document

This document covers the overall objectives of the HMS, including guest management, room reservations, billing and reporting features. It will provide value to customers by streamlining operations, enhancing guest experiences, and improving management efficiency. The estimated development cost and timeline are also included.

1.3 overview

The Hotel management system is designed to automate and manage the day-to-day operations of a hotel. It will facilitate online booking, check-in/check-out processes, and maintain guest records, enhancing operational efficiency and customer satisfaction.

2. General Description

The HMS will serve hotel staff and guests by offering features such as online reservations, room management, invoicing, and reporting. Users include hotel management, front desk staff, and guests. The system will provide easy navigation, quick access to information, and integration with existing software.

3) Functional Requirements

User registration: user can create and manage their accounts

Catalog management: operators can

Room management: staff can add, update and delete room information, including status

Booking system: to book rooms

Billing and invoicing: generate and manage invoices for guests, including additional services

4) Interface Requirements

User Interface: web-based and mobile interfaces for guests and hotel staff

Database Interface: interaction with a backend database for room and guest data management

API Interfaces: Integration with external services, such as payment gateways and travel agencies

5) performance requirements

• Response Time: system should respond to user request within 2 seconds.

concurrent user: support up to 200 simultaneous users.

Data Retrieval: Database queries should return results within 1 second.

6) design constraints

Technology: must utilize specified programming languages and frameworks (e.g. Python, Django, PostgreSQL)

7) non-functional Attributes

security

scalability

reliability

usability

maintainability

8) preliminary schedule and budget

• Timeline: Estimated development duration is 8 months

Budget: Initial budget allocation is \$75,000, covering development, testing and implementation costs.

Split up?

Sgt
2024

credit card system

1. Introduction

1.1 purpose of this document

The purpose of this document is to define the requirements and specifications for the development of credit card management system. This document will provide a comprehensive description of the system features, performance and constraints.

1.2 scope of the Document

The scope of this document covers the design and functionality of a credit card management system that allows both users and Administrators to manage credit card-related activities such as transactions.

1.3 overview

The credit card management system is a secure and efficient system aimed at providing a comprehensive platform for credit cardholders and financial institutions.

- ions to manage and monitor credit card-related services

2. General Description

This system offers secure, transparent credit card management with features like payment processing, transaction monitoring and fraud detection.

Thus this system is designed to help both cardholders and Administrators and this can give real-time access to accounts, fraud alerts, simplified payment process and credit limit management.

3) Functional requirements

- Authentication: secure login with multi-factor auth
- Account Management: Access to balances, credit-limits, and transaction history
- Payment: multiple payment options, real-time processing
- Fraud Detection:
- Support: for any other queries

4) Interface Requirements

- User Interface: web and mobile app with an intuitive dashboard for cardholders
- Admin Interface:
 - Database interface: to communicate easily with database

5) Performance Requirements

• Response Time: ≤ 3 seconds for transaction and data retrieval

• Scalability: support up to 1 million users simultaneously

• Error Rate: transaction error rate < 0.01%

6) Design constraints

- Security Compliance: must comply with PCI-DSS for credit card data protection
- Hardware: scalable cloud infrastructure (e.g. AWS or Azure)

- 7) Non-functional attributes
- security
 - portability
 - reliability
 - data integrity
- 8) preliminary schedule and budget
- schedule: Estimated 6-month timeline (design, development, testing)
 - budget: Approximately \$2,80,000 covering development, infrastructure, and testing

3) Library management system

1. Introduction

1.1 purpose of this document

The purpose of this document is to outline the requirements for library management system (~~process~~). It is mainly to give clear understanding of the ~~the~~ project to stakeholders, developers etc.

1.2 scope of this document

The document covers the overall objectives of the library management system. It includes catalogue management, billing and also about cost and timing of the document and it will provide value to customers by streamlining operations, enhancing user experience and improving management efficiency.

1.3 overview

The library management system is designed to automate and manage the day-to-day operations of a library. It will facilitate ~~any~~ features like displaying catalogue, maintaining database and also details about customers who are taking the books.

2. General description

The system provides features like displaying the catalogue, keep track of the activities like buying, and also it maintains the data, it helps the user to easily access the book and also the user can see where the book is present ~~each~~ ~~area~~ ~~area~~ ~~area~~ ~~area~~.

the different users like, manager, staff, user, etc.

3 Functional requirements

login: it should have one login page to login both user and staff and etc.

catalogue: it should have catalogues to display about the books.

customer details: it ~~not~~ should keep track of the user details.

Databases: it should have database to store all the activities and also book details and track details about the book.

4) Interface requirements:

User Interface: web-based and mobile interfaces for users ~~and~~ and staff

data base interface: interface to retrieve data from database

API interface: to communicate with ~~the~~ backend

5) Performance requirements

Response time: the response time must be less than 2 sec

Concurrent users: it must support over 200 users concurrently

Data retrieval: data should be retrieved efficiently

6) Design Constraints:
It must use programming languages
and frameworks (e.g. python, Django,
PostgreSQL)

7) Non-functional Attributes

Security

Scalability

Reliability

Usability

Maintainability

8) Preliminary

Timeline: Estimated development duration
is 8 months

Budget: Initial budget allocation is
\$75,000, covering development, testing and
implementation costs

credit card system

1> introduction:

1.1 purpose of this document:

This document outlines the requirements for credit card Management system (ccms).

1.2 scope of this document

The ccms is designed to facilitate the management of credit card applications, transactions, billing, and customer support.

1.3 overview

The ~~ccms~~ ccms will automate processes related to credit card issuance, transaction monitoring, and customer account management.

2> General description

The ccms aims to provide financial institutions with an efficient platform to manage credit card operations while enhancing customer experience. Key features include a user-friendly interface for cardholder and bank staff, transactional and comprehensive reporting tools.

3> Functional Requirements

User registration: customers can apply for credit cards and create accounts

Application processing

Transaction monitoring

Billing statement: automatic generation and distribution of monthly billing statements

A) Interface Requirements

- User Interface: Intuitive web and mobile interface for customers and bank staff.
- Database Interface: Integration with secure database for transaction and user data management.
- API Interfaces: connectivity with external payment processors

5) Performance Requirements

- Response time: System should respond to user actions within 2 seconds
- concurrent users: support up to 500 simultaneous users
- Data Retrieval: Database queries should return results within 1 second

6) Design constraints

must use:

java, spring, mysql

7) Non-Function Attributes

- security
- scalability
- reliability
- maintainability
- usability

8) Preliminary schedule and Budget

- Timeline: Estimated development duration is 10 months

- Budget: initial budget allocation is \$10,000 covering development, testing and deployment cost.

Stock maintenance system

Introduction

1.1 purpose

The purpose of this document is to define the requirements for developing a stock maintenance system.

1.2 scope

The stock maintenance system is designed to help business manage inventory levels, track stock movements, generate reports, and provide alerts.

1.3 overview

The stock maintenance system will track product quantities, manage supplier information, and generate real-time reports.

2) General Description

Provide businesses with an automated solution to manage their stock, track inventory movements, and ensure efficient ~~replenishment~~ services.

users

- warehouse managers

- storekeepers

- Administrators

The benefits of this system are:

- Real-time inventory tracking

- Automated reorder alerts

- Simplified reporting and analysis of stock levels

3. Functional Requirements

- User Authentication: Secure login for all users with role-based access control.

- Stock Tracking

- Inventory Updates

- Stock Reports: Generate daily, or monthly reports on inventory status including stock levels, usage trends

4. Interface Requirements

- User Interface:
Dashboard showing stock levels, pending orders and alerts

- Admin Interface

- Tools for managing user roles, stock items and supplier details

5. Performance Requirements

- Response Time: System should update stock levels and process orders within 2 seconds.

- Scalability: System must handle multiple warehouses and up to 500,000 stock entries

- Error Rate: Error rate for stock mismatch or duplication should not exceed 0.01%

6. Design Constraints

- Compliance: Must comply with industry standards for stock and inventory management.

- Must use ~~language~~ tools like Java, Spring Boot or Node.js, frontend React or Angular.

7) non functional attributes

- security
- portability
- Reliability
- data integrity
- scalability

8) preliminary schedule and Budget

Schedule

- Requirements specification : 2 weeks
- System design : 3 weeks
- Development : 10 weeks
- Testing : 3 weeks
- TOTAL : 5 months

Budget

Software Development : \$150,000

Infrastructure / Hard ware : \$40,000

Testing : \$20,000

Total : \$210,000

Passport automation

Introduction

1.1 purpose

The purpose of this document is to outline the functional and non-functional requirements for the passport Automation system.

1.2 scope

This system will Automate the entire process of applying for and issuing passports.

1.3 overview

The system will provide a user-friendly platform for citizens to apply for passports, track their application status, schedule appointments for interviews or document submission.

2) General Description

To automate and streamline passport application process, reducing the manual workload for passport officers and improving the user experience for applicants.

• users

Applicants

passport officers

3) Functional requirements

- user Authentication: secure login system
- Document verification
- Application tracking
- passport issuance

7) Interface requirements

- user interface: web or mobile interface
- database interface
- office interface: for verification of documents

5) performance requirements

- response time: Application submission should occur within 3 seconds
- scalability: support up to 1 million users

6) design constraints

- technology: Backend (Java/Spring Boot)
Frontend (React), Database (MySQL)

7) non-functional requirements

- security
- portability
- reliability
- data integrity

8) preliminary schedule and budget

Schedule: Requirements specification: 1 week

Development: 4 weeks

Testing: 4 weeks

Total: 6 months

Budget

Software Development: \$250,000

Infrastructure and Hosting: \$60,000

Testing QA: \$40,000

Total Budget: \$350,000