

Software Requirement Specification (SRS)

1. Introduction

1.1 Purpose of this Document

The SRS document outlines the detailed requirements for Hotel Management System. It serves as blue print for the development team, ensuring final product meets the specified specification and fulfills the customer's requirements.

1.2 Scope of this document

It aims to provide clear specifications for the system's development, including overview of costs and timelines involved. This HMS (Hotel Management System) is designed to provide streamline hotel operations and enhance customer experience.

1.3 Overview

The HMS will provide solution for managing hotel operations including reservations, customer management, billings and reportings. This will enable staff to handle daily tasks and offer user friendly interface for bookings.

2. General Description

2.1 User characteristics - The general users for HMS will be hotel staff, guests and administrators.

3. Features and Benefits

3.1 User-friendly Interface

3.2 Real-time availability

3.3 Automated Reporting

Mobile compatibility

Real-time information

2.3 Importance

The HMS is crucial for optimising hotel operations, improving customer satisfaction and increasing revenue through efficient management of resources.

3. Functional Requirements

3.1 Reservation Management

- Users can search for available rooms
- Users can create, modify, or cancel reservations.

3.2 Customer Management

- System can store guest info, like contact details, preferences.
- Staff can view guest history, handle complaints.

3.3 Billings and Payments

- Generate invoices and process payments via various methods.
- Send automated payment confirmation emails.

3.4 Reporting

- Generate occupancy reports, service reports.
- Schedule regular reports to be sent to admin.

4. Interface Requirements

4.01 User Interfaces

- Web and Mobile interfaces for staff and guest to manage bookings.

4.02 System Interfaces

- Database connection - Interface for the HMS to communicate with database for data storage.

Payment Gateway - Integration with third party payment processors for handling transactions

5. Performance Requirements

- System should respond to users within 10 seconds.
- Support simultaneous access for 1000 users.
- Maximum error acceptable rate for payments should be less than 1%.

6. Design Constraints

- Must utilize specific set of technologies like Java for backend and React for frontend.
- Use of relational database (e.g. MySQL) for data management.

7. Non-functional

- Security - Implementing user authentication mechanisms.
- Portability - System should operate on various devices.
- Reliability - The system should have 99.9% uptime.
- Scalability - Ability to scale resources on demand.

8. Preliminary Schedule and Budget

8.1 Schedule

- Project Duration - 6 months
 - Requirements Gathering - 1 month
 - Design phase - 1 month
 - Development phase - 3 months
 - Testing phase - 1 month

8.2 Budget

- | |
|-----------------------------------|
| ◦ Estimated Total cost - ₹ 55,000 |
| ◦ Development - ₹ 30,000 |
| ◦ Software license - ₹ 10,000 |
| ◦ Hardware - ₹ 15,000 |
| ◦ Miscellaneous - ₹ 10,000 |

Credit Card Processing System

1. Introduction

1.01 Purpose of this Document

The purpose of this document is to define the software requirements for the development of credit card processing system. This system will facilitate secure, efficient and reliable processing of credit card transactions.

1.02 Scope of the Document

This document covers the software specifications for a system that handles credit card authorization, transaction processing. It will be used by developers, testers and stakeholders to ensure the system meets both business and technical objectives.

1.03 Overview

This processing system will allow users to authorize, process and settle payments using credit cards. This system will be compatible with various payment gateways.

2. General Description

This system will provide users with secure and efficient way to complete transactions. Its key feature includes real-time authorization, encrypted communication, fraud detection mechanisms.

3. Functional Requirements

Authorization - The system will verify the availability of funds and validity of credit card details.

Transaction Processing - It will ensure purchases are processed accurately.

Settlement - The system will transfer funds from customer's account to the receiver account.

4. Interface requirements

The system will offer an API for integration with e-commerce platforms.

5. Performance requirement

The system must process transaction within 2 sec under normal condition with failure rate of less than 0.1%.

6. Design constraints

- Must use secure encryption methods.
- Should be compatible with major credit card networks.

7. Non-functional

→ Uptime should be at least 99.99%.

→ All data must be encrypted during transmission.

8. Preliminary Schedule & Budget

The project is estimated to be completed within 6 months and total development budget is 1 lac.

Development - 85,000, Software License - 10,000, Hardware - 25,000.
Miscellaneous - 20,000.

Library Management System

1. Introduction

1.1 Purpose of this document

The SRS document outlines the detailed requirement for Library Management system. It serves as a blue print for the development team, ensuring final product meets the specified specification.

1.2 Scope of the Document

This document aims to include book details, video lectures, magazine, previous year question papers. It will be used by students, staff, faculty, indirect stakeholders like technician.

1.3 Overview

The Library Management System will provide solution for managing operations like issuing books, catalogues, fine management, member management, registration functioning. This will enable staff and other users to efficiently use and offer userfriendly interface.

2. General Description

The LMS will provide a platform for managing library resources efficiently. This system will provide registering students, cataloguing books, issuing/returning books.

3. ~~Functional Requirements~~

~~Registration Management - Users should be able to register to the system.~~

List of

- Users should be able to search for available books overdues.
- System should be able to store registration details like contact information, etc.

Book Management:

- The system will allow for addition, modification of book details.

Search facility - users should be able to search for books based on title, ISBN.

Interface Requirements:

The system should be able to graphical user interface accessible through web browser.

The system will integrate existing database to store book and user information.

Performance Requirements:

- o The system must be able to handle 1,00,000 books and 10,000 users.
- o Book search result should be available within 3 sec

Design Constraints

- o The database should support MySQL for large data.
- o Must be compatible with modern web browsers (Google chrome)

Non-functional

- o Uptime should be atleast 99.99%.
- o The system must protect user data from unauthorised access.

Preliminary Schedule & Budget:

Hardware - 25,000
Miscellaneous - 30,000

classmate

Date 20/9/24
Page

Stock Maintenance System

1.0 Introduction

1.1 Purpose of this Document

The SRS document outlines the detailed requirement for Stock Maintenance System. This system will help business to manage and track stock levels accurately.

1.2 Scope of this Document

It aims to manage stocks of various products. It helps to monitor and record purchases of the various stocks. The system is intended for ~~a~~ business and retail stores.

1.3 Overview

This SMS will help in tracking realtime stock levels and generate reports based on these stock levels, and purchases.

2. General Description

This Stock Maintenance System will help in the process of managing stocks. The primary users will be business managers, store employees and owners of various business.

3. Functional Requirements

- The system should be able to keep the quantity of stock available for each item.

- Automatically generate purchase recorder if stock quantity goes below the minimum level.

→ The system should be able to generate reports based on current stocks & sales trends.

→ The system should display alerts if it goes below minimum level.

4. Interface Requirements

The system will feature user friendly interface for stock management. It also will provide APIs to integrate with other systems like sales platform.

5. Performance Requirements

- The system should handle real-time stock updates for 1000 products and 10,000 individuals.
- Reports should be generated within 5 sec.

6. Design Constraints

- The system should be compatible with modern web browsers like Chrome.
- The system must support barcode scanning.

7. Non-functional requirements

Security - The system must have implemented access control to protect sensitive information.

Uptime should be atleast 99.99%.

8. Preliminary Schedule

The project is estimated to be complete within 4 months and total development budget of 1.60.

Requirements - 1 month	{ Development - 35,000; Miscellaneous - 10,000; Software license - 10,000; Hardware - 25,000 }
Design Phase - 0.5 month	
Development Phase - 2 months	
Testing Phase - 0.5 months	

Passport Automation System

1. Introduction

1.1 Purpose of this document:

The document outlines the objectives and benefits of creating a passport automation system, which simplifies passport application and processing procedures.

1.2 Scope of the Document - It covers entire passport process, including application system submission, verification and issue, while detailing cost required for development.

1.3 Overview: The system automates passport-related tasks, improving efficiency and reducing manual intervention.

2. General Description -

This system facilitates users, in applying for passport online, benefit applicants by saving time.

3. Functional Requirements:

- Users can submit applications online with required documents.
- Allow users to schedule appointments for document verification.
- Users can track the status of their application in real time.
- Users can register using a unique ID such as Aadhar card.

4. Interface Requirements

- The system will feature user-friendly, web-based interface for passport automation system.
- The system will interface with the national identity database for verification purpose.

5.

Performance Requirements

- the system should process user actions within 3 seconds.
- The system should maintain a rate of less than 1000 users, while maintaining tasks of a certain complexity.

6.

Design Constraints

- The system should be able to manage modern web browsers.
- The system must adhere to government regulations and laws.

7. Non-functional Requirements

- The system must use secure login methods for data and transmission.
- The system should be suitable to scale with increasing user numbers and structures.

8.

Preliminary Schedule and Budget

The development is estimated to take 6 months with total budget of 1,00,000.

Development - 60,000

Software license - 30,000

Testing and Maintenance - 20,000

Class Diagram

Hotel Management System:

Housekeeper
keeporderId: String
name: String
cleans()

Employee
employeeID: String
name: String
salary: Double
assignRoom()

classmate

4/10/24

Date

Page

Manager
managerID: String
name: String
manages()

Hotel
name: String
location: String
rooms: List<Room>
staff: List<Staff>
getAvailRooms()
getStaffDetails()
* serves

Manages

Room
roomNumber: String
roomType: String
price: Double
isAvailable: Boolean
checkAvailability()
bookRoom()
releaseRoom()

Customer
customerID: String
name: String
phoneNumber: Integer
makeBooking()
cancelBooking()
makePayment()

occupies

Payment
paymentID: Integer
amount: Double
* paymentMethod: String
status: String
processPayment()

<enum>
PaymentMethod
(from Payment)
Card
Cash
Online

Credit Card Processing

Customer
customerID: String
name: String
phoneNumber: Integer
register()
updateProfile()
Delete Account()

Account
accountNo: String
balance
openAccount()

has

Credit Card
cardNumber: String
expiryDate: Date
CVV: String
isActive: Boolean
activateCard()
deactivateCard()
validateCard()

BankAccount
accountNumber: String
accountDetails()

Transaction

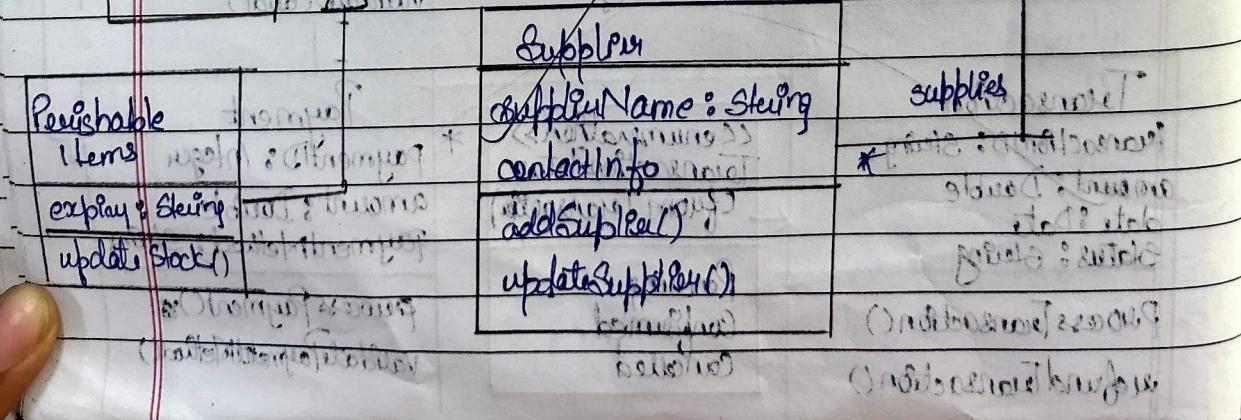
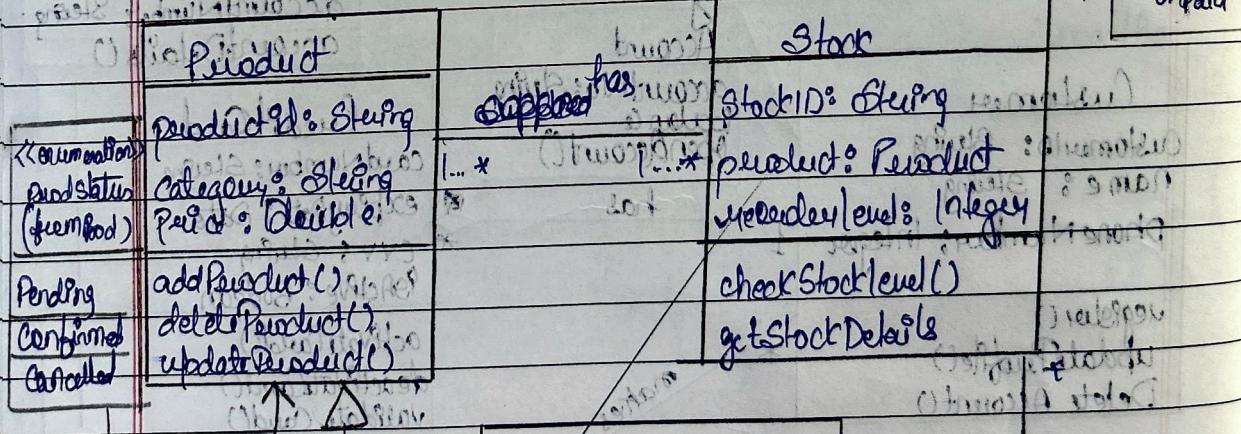
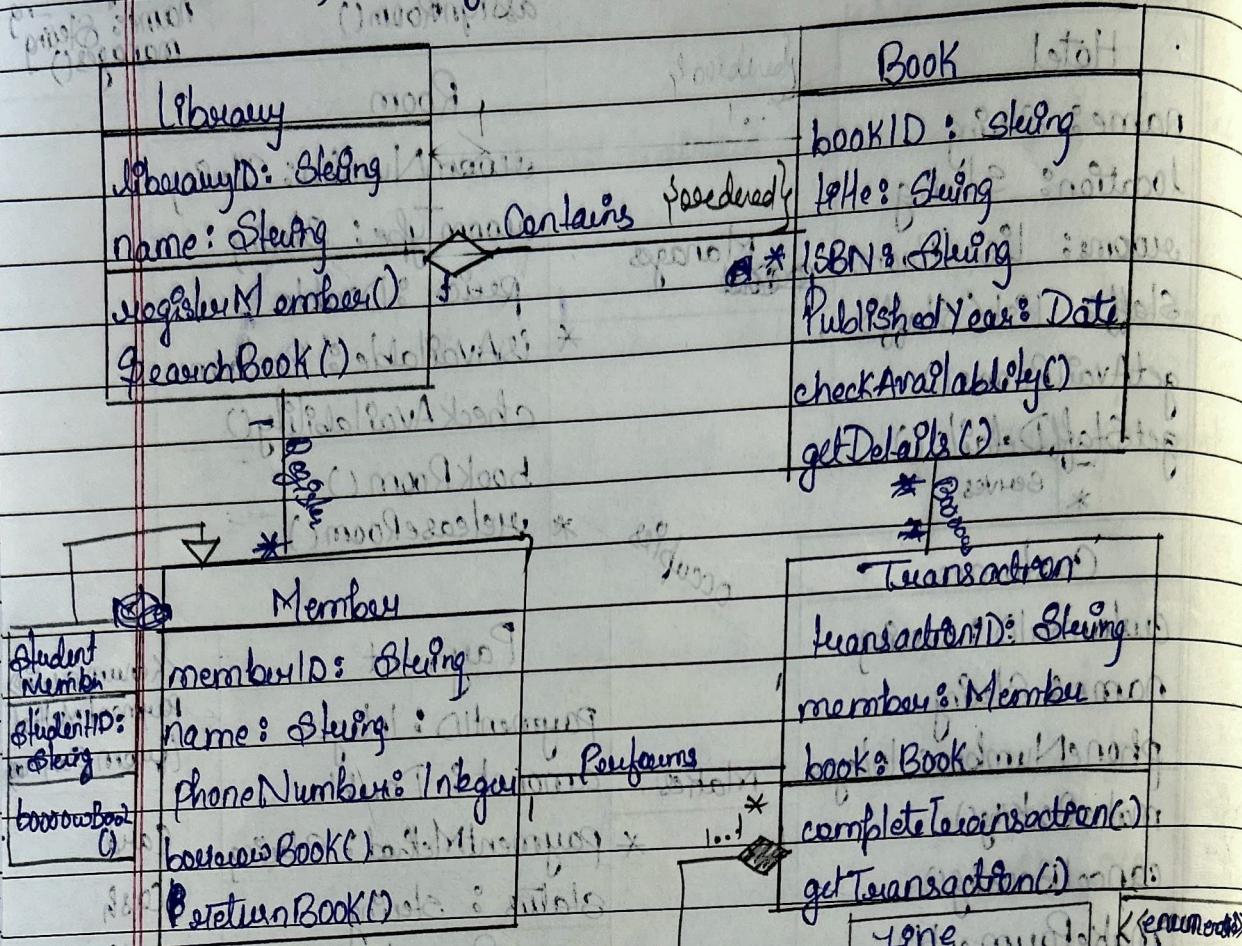
transactionID: String
amount: Double
date: Date
status: String
processTransaction()
refundTransaction()

<enum>
TransactionStatus
(from Transaction)
Pending
Confirmed
Cancelled

Payment

paymentID: Integer
amount: Double
paymentMethod: String
processPayment()
validatePaymentMethod()

Library Management System



Passport Automation System

