

24-3-24

## 9.) Hotel Management System

### 1. Introduction

#### 1.1 Purpose of this document :

This SRS defines the functional & non-functional requirements for a Hotel Management System. It serves as a blueprint for the development team, outlining the expected capabilities & behaviours of the system.

#### 1.2 Scope of this Document :

- Guest Reservations
- Room Management
- Inventory Management
- Staff Management
- Billing & Payment

#### 1.3 Overview :

The Hotel Management System will be a web-based application accessible to hotel staff. It will provide a user-friendly interface for managing day-to-day operations efficiently and accurately.

### 2. General Description

#### 2.1 User Characteristics :

- Hotel Staff

- Guest

Hotel Staff : Managers, cleaners or security staff  
Guests : Visitors or clients staying at the hotel

## 2.2 User Interface

- Easy to use and Intuitive for all users.
- Responsive design for different devices.
- Concise Navigation

## 2.3 Features :

- Online Reservation & Cancellation
- Guest Details Management
- Room Management
- Check-in / Check-out
- Inventory Management
- Staff scheduling
- Billing and Payment

## 2.4 Benefits :

- Increased efficiency
- Improved Guest satisfaction
- Reduces Manual tasks
- Saves time

## 3 Functional Requirements

### • Reservation Management

- Allow user to create, modify & cancel reservations
- Validate Guests info by availability

### • Room Management

- Main Database of different rooms
- Track and update room status
- Room Allocation

## • Inventory Management

- Track availability of items like tissues, toiletries etc.
- Set reorder points & replenishment

## • Staff Management

- Create staff profiles with roles
- Schedule shifts & tasks for staff
- Attendance

## • Billing & Payments

- Calculate guest bills

(Guest bill) - process payment via all methods

(Guest bill) - generate invoices

## 4. Interface Requirements :

### • User Interface:

- Adhere to industry std. guidelines.
- (Provides) clear messages

### • Safety Requirement System Interface:

- Integrate payment gateway.

## 5. Performance Requirements

- Response time should be reduced
- It should be scalable.
- It should be secure.

## 6. Design Constraints

### • Standalone system

- It should be a stand alone system.

- Preferred programming language Java and an Access or Oracle database.

- Standards Compliance

## 7. Non Functional Attributes.

- Highly reliable & easy to use for everyone.
- Easy to maintain and must be secure.
- Portability should be kept in mind.

## 8. Preliminary Schedule And Budget

- Schedule:
- Requirement gathering and Analysis : (3-4 weeks)
  - System Design : (2-3 weeks)
  - Development : (10-12 weeks)
  - Testing and validation : (4-6 weeks)
  - Deployment & Training : (2-3 weeks)
- Estimated weeks : ( 21-25 weeks).

## Budget :

- Requirement Documentation : (₹ 100000)

- Design by tools : (₹ 200000)

- Development : (₹ 500000)

- Testing : (₹ 200000)

Total estimated Budget : (₹ 1600000).

## Q1. Credit Card Processing System

### 1. Introduction

#### 1.1 Purpose of this document:

This ORS defines all the requirement for a credit card processing system. It serves as a blueprint for the dev. team, outlining the behaviour of the system along with its capabilities.

#### 1.2 Scope of this document:

- Authorization

- Capture

- Refund

- Charge back

- Void

#### 1.3 Overview:

The CCPS will be integrated with a payment with a payment gateway to facilitate secure and efficient transactions. It will provide a user-friendly interface for merchants to manage their transaction by generating reports.

### 2. General Description

#### 2.1 User Requirement Characteristics

- Merchants who will use system for a company

- Customers who will provide card.

## 2.2 User Interface

- easy to use & intuitive
- secure & encrypted data transaction
- support various card types

## 2.3 Features

- verify validity of C.C. & availability of funds.
- Customer's authorization is must for transactions.
- Process a refund for a previous transaction.
- Cancel a transaction before it is captured.
- Handle chargeback issued by banks.
- Process charges for subscription.
- Generate report.

## 2.4 Benefits

- Easy payment process
- Reduced fraud risk
- Improved customer satisfaction
- Enhanced business efficiency
- Save time

## 3. Functional Requirements

### Authorization

- verify C.C no, expiry date, e.g. CVV.
- check for sufficient funds
- response message

### Refund

- process a refund for captured transactions.
- Up

#### • Void

- Cancel a transaction before it is captured.
- Update transaction status.

#### • Charge back

- Handle chargeback initiated by banks or customers.
- Gather evidence e.g. respond to reqs.

#### • Recurring Billing

- Store customer billing info.
- process charges on a scheduled basis.

#### • Reporting

- generate report on transaction volume, revenue, funds etc.
- provides data visualization tools.

### 4. Interface Requirements:

- Integrate with secure & reliable payment gateway.
- Adhere to gateway's API std.
- User-friendly interface.
- Secure data transmission.

### 5. Performance Requirements:

- Response time should be reduced for better experience.
- It should be scalable.
- It should be secure.
- It should be reliable and have less downtime.

### 6. Design Constraints:

- Security must be implemented for sensitive data.
- Industry std. must be followed.

- The preferred prog. language, frameworks & db must be specified.

## 7. Non functional Requirements:

- Easy to use for merchants
- Maintainability
- Portability & security

## 8. Preliminary Schedule & Budget:

- estimate of dev & deployment timeline.

- estimate of overall budget must be done.

- Estimated schedule for this project is :

- Requirement Gathering & analysis - (4-5 weeks)
- System Design - (2-3 weeks)
- Development - (8-10 weeks)
- Testing and validation - (4-6 weeks)
- Deployment & Training - (2-3 weeks)

Total estimated weeks - (19-23 weeks)

- Estimated budget for this project is :

- Requirement documents - (₹ 100000)

- Design eq. tools - (₹ 200000)

- Development - (₹ 400000)

- Testing - (₹ 200000)

Total Estimated Budget - (₹ 900000)

30-09-24

## Q. Library Management System

### 1. Introduction

#### 1.1 Purpose of the document

The system will manage library operations such as book borrowing, returns, user registration and catalog management.

#### 1.2 Scope of the document

This provides detailed overview of system's objectives, features and user interactions. CMS will automate management of library resources, including cataloging, borrowing, returning and backing books.

#### 1.3 Overview

The system will facilitate the operations of physical library by allowing librarians to track inventory, borrowers to borrow books, returned books, generate reports. It will support multiple user roles, administration, user members, efficient platform to manage library digitally.

### 2. General Description

#### 2.1 User characteristics

- Library staff
- General public

#### 2.2 User Interface

- Easy to use & intuitive.

- Secure & encrypted data transactions

- Support various functionalities.

## 2.3 Features

The system will function as a centralized platform for managing library resources. It will enable users to search books etc. Librarians, administrators will have ability to add, update, remove books for library management. Easy tracking of book inventory.

## 3. Functional Requirements

- User registration, authentication new user to register and existing user to log in using credentials.
- Book catalog - administrator can add, update and remove books.
- Book searching and Borrowing
- Book Return, fine Calculation.
- Notification System
- Report Generation.

## 4. Interface Requirements

User Interface: System provides a easy web-based, interface for both librarians & users.

API: System will have API's for integration system will send reminders for users to return books on time.

## 5. Performance Requirements

- Have least response time.
- Should be able to handle load.
- Should recover easily e.g. quickly after a failure to reduce down time.

## 6. Design Constraints

- Security: Prioritize security measure to protect sys.
- Scalability: Future growth & increasing demand should be kept in mind.
- Integration: Must consider other IIS system by comms.

## 7. Non-functional attributes:

- Easy to use
- Maintainability
- Interoperability
- Portability

## B. Preliminary Schedule & Budget

Schedule for this project is estimated as:

- Requirement gathering & Analysis - (4-5 weeks)
- System Design - (2-3 weeks)
- Development - (8-10 weeks)
- Testing & Validation - (4-6 weeks)
- Deployment & Training - (2-3 weeks)

Total Estimated weeks - (19-23 weeks)

Estimated project for this project is:

- Requirement documentation - (£100000)
- Design & tools - (£50000)

Development - (£100000)

- Testing - (£50000)

Total Estimated Budget - (£300000)

# 9 Stock Maintenance System

## 1. Introduction

### 1.1 Purpose of this document

This document outlines software reqs for the system. It serves as guide for developers, stakeholders and also end users, detailing the system's functionalities, performance criteria and design constraints.

### 1.2 Scope of this document

The document covers objectives, functionalities, expected outcomes of stock maintenance system. It includes an estimation of developments cost and time required, providing valuable insights for customers.

### 1.3 Overview

The system is designed to automate inventory tracking & managing stock levels, & facilitates order processing. It ensures real-time data accuracy, reduces manual efforts, supports efficient stock management practices.

## 2. General Description

The system aims to assist users in maintaining optimal stock levels, preventing overstocking or stockouts, users will benefit from automated alerts, detailed inventory reports and user-friendly interface. The primary users include inventory managers, warehouse staff & procurement teams.

### 3. Functional Requirements

- Real-time inventory tracking and update.
- Automated reorder alerts when stock level fall below predefined threshold.
- Detailed reporting on stock levels, reorder history and stock movements.
- User authentication and role based access controls.
- Integration with existing ERP system for seamless data flow.
- Search and filter capabilities for quick access to stock information.

### 4. Interface Requirements

- A user-friendly graphical interface for easy navigation and operation.
- API's for integration with external systems
- Data import/export capabilities in standard formats
- Notification interfaces for email & SMS alerts

### 5. Performance Requirements

- System should handle upto 10,000 stocks entries.
- Real time processing with response time with 2sec for user actions
- Minimal memory usage to ensure efficient performance.
- Ensure error rate should be very less.

### 6. Design Constraints

- The system must be compatible with all devices.
- MySQL can be used for database
- Compliance with industry standards for data security & privacy
- Implementation of Restful API for external integration.

## J. Non-Functional Attributes.

- Security should be provided for data, login, sign up.
- Should accessible via browser & mobile.
- It should be reliable.
- Design must be modular for easy updates & changes.
- It should be scalable.

## 8. Preliminary Schedule and Budget

Schedule: Requirements Gathering : 2 weeks

System Design : 3 weeks

Development : 8 weeks

Testing : 4 weeks

Deployment : 2 weeks

Total time for completion : 19 weeks

## Budget:

- Requirements documentation - £ 500 00

- Design & tools - £ 1500 00

- Development - £ 400 000

- Testing - £ 50 000

Total estimated budget : £ 750 000

Q5.

## PASSPORT AUTOMATION SYSTEM

### 1. Introduction

#### 1.1 Purpose of this document

This document outlines the software req. for the passport automation system. It serves as a comprehensive guide for developers, stakeholders & end-users, detailing system's functionalities, performance criteria & design criteria.

#### 1.2 Scope of this document

Document covers the objectives, functionalities, expected outcomes of P.A.S., It includes an estimation of development costs & the time required, providing valuable insights for customers.

### 2. General Description

System aims to assist user in submitting passport application tracking application status and receiving passports efficiently. users will benefit from automated notifications, user friendly interface. The primary users include application, passport officers administrative staff.

### 3. Functional Requirements

- Online application submission & document upload.
- Automated verification of application details and documents.
- Real time tracking of application status.
- User authentication and role based access control.
- Integration with government database for data verification.
- Notification system for updates via SMS & email.

- Appointment scheduling for biomarker & document verification.

#### 4. Interface Requirements

- A user friendly graphical interface for easy navigation & operations.
- API's for integration with external systems.
- Data import/export capabilities in std. formats.
- Notification interfaces for email and SMS alerts.

#### 5. Performance Requirements

- The system should handle 50000 conventional users.
- Real-time processing with response times within few seconds for every action.
- Minimal memory usage to ensure efficient performance.
- Error rate & downtime should be minimized.

#### 6. Design Constraints

- The system must be compatible with all types of devices & systems.
- PostgreSQL to be used for database.
- Compliance with gov. regulations for data security & privacy.
- Implementation of RESTful API's for external integration.

#### 7. Non-Functional Attributes

- Security: Role-based access control, data encryption & secure login.
- System should be accessible via web browsers & mobile devices.
- Reliability: Reduces downtime with backup.

- Modular design for easy update & changes.
- It should be scalable.

## B. Preliminary Schedule & Budget

Schedule : Requirements Gathering	: 3 weeks
System Design	: 6 weeks
Development	: 10 weeks
Testing	: 5 weeks
Deployment	: 3 weeks

Estimated schedule for project is : 27 weeks.

## Budget

Requirements documentation, £ 100000

Design & tools : £ 50000

Development : £ 100000

Testing : £ 40000

Estimated budget for project is : £ 2000000.

7/30/24

## Class Identification

### 1. Hotel Management System:

Hotel: Represents a hotel entity, including details like name, location, rating.

Room: Represents rooms in the hotel, with attributes like room type, availability, and price.

Booking: Handles room reservation, booking status and related data.

Guest: Contains guest details such as name, contact info, and booking history.

Staff: Represents hotel staff members, including roles like housekeeping, receptionist, etc.

Invoice: Manages billing and payment details for a stay.

Service: Includes services offered by the hotel (e.g. laundry, room service).

### 2. Credit Card Processing System:

Cardholder: Stores details of the credit card owner (e.g. name, address).

Credit Card: Represents the actual credit card, including number, expiration date and limits.

Transaction: Handles individual card transactions, such as purchase or refunds.

Payment Gateway: Facilitates bank-merchant transactions.

Merchant: Represents merchants accepting payments.

Authorizations: Manages authorization checks for card transactions.

Fraud Detection: Detects potential fraud.

Account statements: Generates monthly usage statements.

### 3. Library Management System:

- Book: Represents books in library, including title, author, availability.
- Member: Stores detail about library members (e.g. name, membership ID).
- Liberation: Manages library operations (adding/removing books, managing members).
- BorrowTransaction: Tracks borrowing, due dates, returns.
- Fine: Calculates and stores overdue fines for late book returns.
- Category: Organizes books by genre or category.
- Reservation: Handles the reservation of books by members.
- Author: Manages info of books' authors.

### 4. Stock Maintenance System:

- Product: Represents each product in the stock, including name, description and price.
- Stock: Tracks stock level and details like quantity available, reorder levels.
- Supplier: Contains details of supplier who provides the products.
- ~~Order: Manages orders and inventory updates.~~
- Customer: Stores customer information and purchase history.
- Invoice: Handles billing and payment for stock purchase.
- Warehouse: Manages details related to storage locations and inventory tracking.
- Report: Generates report on stock level, sales and reordering needs.

## 5. Passport Automation System

Applicant : Stores details about the passport applicant (e.g. name, address, ID proof).

Passport : Represents the passport issued to applicants with validity and passport number.

Application : Manages the application process, including submission, processing and status.

Verification Officer : Handles the verification of applicant documents.

Appointment : Manages appointment scheduling for interviews or verification.

Fee Payment : Handles payment details for the passport application.

Notification : Sends notifications to applicants regarding application status.

Renewal : Handles passport renewals and associated procedures.

