

# Object Oriented Modelling

22/08/25

Lab - 1

## SRS Document

### 1. Hotel Management System

#### 1. Introduction

1.1 Purpose of this document: The purpose of this

document is to outline the requirements and specifications for the development of a Hotel Management System. It will provide a clear understanding of the project objectives, scope and deliverables.

1.2 Scope of this document: This document defines

the overall working and main objectives of the hotel Management system. It includes a description of the development cost and time required for the project.

1.3 Overview: The Hotel Management System is

a software solution designed to streamline hotel operations, including reservation management, guest check in/check out, room assignment, billing and reporting.

#### 2. General Description:

The Hotel Management System will cater to the needs of hotel staff and management, providing features such as room booking, guest profiles, inventory management and financial reporting. It will be accessible to users with varying levels of technical expertise.

### 3. Functional Requirements:

#### 3.1 Reservation Management:

- \* Allow users to make room reservations online or through front desk.
- \* Generate reservation confirmations and send notifications to guests.

#### 3.2 Room Management:

- \* Assign rooms to guests based on availability and preferences.
- \* Track room status (clean, occupied, vacant) in real-time.

#### 3.3 Guest Management:

- \* Maintain guest profiles with personal information, preferences and booking history.
- \* Facilitate guest check-in and check-out processes.

### 3.4 Billing and Invoicing:

- \* Generate accurate bills for room charges, additional services, and taxes.
- \* Accept various payment methods and generate invoices for corporate clients.

(Assignment 239)

### 4. Interface Requirements:

#### 4.1 User Interface:

- \* Intuitive and user-friendly interface for hotel staff and guests.
- \* Accessible via web browsers, mobile devices and desktop applications.

#### 4.2 Integration Interfaces:

- \* Integration with payment gateways for secure transactions.
- \* Integration with third-party booking platforms for seamless reservation management.

### 5. Performance Requirements:

#### 5.1 Response Time:

- \* The system should respond to user actions within 2 seconds.

#### 5.2 Scalability:

- \* Handle a minimum of 1000 concurrent users during peak hours.

#### 5.3 Data Integrity:

- \* Ensure data consistency and accuracy across all modules.

## 6. Design Constraints:

### 6.1 Hardware Limitations:

- \* The system should be compatible with standard hotel hardware (computers, printers, POS terminals).

### 6.2 Software Dependencies:

- \* Utilizes a relational database management system (e.g. MySQL) for data storage.
- \* Use programming languages and frameworks conducive to UML modeling (e.g. Java, Spring Boot).

## 7. Non-Functional Attributes:

### 7.1 Security:

- \* Implement robust authentication and authorization mechanisms to protect sensitive data.

### 7.2 Reliability:

- \* Ensure high availability and fault tolerance to minimize system downtime.

### 7.3 Scalability:

- \* Design the system to accommodate future growth and expansion.

### 7.4 Portability:

- \* Support multiple platforms and devices for user accessibility.

### 7.5 Usability:

\* The system shall have a user-friendly interface with clear navigation. (5)

#### 7.6 Reusability:

\* The system shall use modular code design to facilitate future enhancements and maintenance.

#### 7.7 Compatibility:

\* The system shall be compatible with common web browsers (Chrome, Firefox, Safari).

#### 7.8 Data Integrity:

\* The system shall ensure accurate and consistent data storage and retrieval.

### 8. Preliminary Schedule and Budget:

The development of the Hotel Management System is estimated to take 6 months with a budget of \$100,000. This includes project planning, development, testing and deployment phases.

## 2) Credit Card Processing

### 1. Introduction:

1.1. Purpose: This document describes the requirement for a Credit Card Processing system that allows secure authorization, authentication and settlement of transactions between cardholders, merchants and banks.

## 1.2 Scope :

The system will enable users to perform online and offline credit card transactions, validate details, detect fraud and process payments with minimal latency.

## 1.3 Definitions :

- \* Card holder: Customer using a credit card
- \* Merchant: Business accepting card payments.
- \* Issuer Bank: Bank that issued credit card.
- \* Acquirer Bank: Merchant's Bank.

## 1.4 Overview:

The system will integrate with merchants, banks and payment gateways, ensuring secure, fast and reliable processing.

## 2. General Description:

The system will cater to merchants, banks and customers by ensuring fast and secure transaction processing. It will provide real-time fraud detection, encrypted communications and transaction settlement.

## 3. Functional Requirements:

### 3.1 Transaction Authorization:

- \* Validate card details (Card number, expiry etc)
- \* Authenticate customers using OTP/biometric

### 3.2 Payment Settlement

- \* Facilitate fund transfers between issuer and acquirer banks.
- \* Generate daily settlement reports.

### 3.3 Fraud Detection:

- \* Identify unusual spending patterns.
- \* Notify banks and customers of suspicious activity.

### 3.4 Reporting and Logs:

- \* Maintain transaction history.
- \* Generate monthly financial summaries.

## 4. Interface Requirements:

### 4.1 User Interface:

- \* Simple and secure interface for customers, merchants and administrators.
- \* Accessible via web and mobile platforms.

### 4.2 Integration Interfaces:

- \* Integration with merchant POS system.
- \* API support for third-party applications.

## 5. Performance Requirements:

### 5.1 Response Time:

- \* Transactions must be processed within 3 seconds.

### 5.2 Scalability:

- \* Handle at least 10,000 concurrent transactions.

### 5.3 Data Integrity:

- \* Ensure accuracy and consistency of financial data.

## 6. Non Functional Attributes:

(8)

### 6.1 Security:

- \* End-to-End encryption and secure authentication.

### 6.2 Reliability:

- \* 99.9% uptime guarantee.

### 6.3 Scalability:

- \* Support growing merchant and customer base.

### 6.4 Portability:

- \* Support mobile, web and POS platforms.

### 6.5 Reusability:

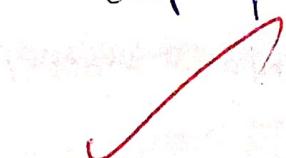
- \* Modular architecture for future enhancements.

### 6.6 Data Integrity:

- \* Accurate and consistent transaction records.

## 7. Preliminary Schedule and Budget

The development is estimated to take 9 months with a budget of \$500,000, covering planning, development, security audits and deployment.



✓  
✓  
✓

## 3) Library Management System

### 1. Introduction:

#### 1.1. Purpose of this Document:

The purpose of this document is to specify the requirements of a Library Management System that automates books cataloging, user registration, borrowing, returning and fine management.

#### 1.2 Scope of this Document:

This document defines the objectives of automating library operations to save time, improve efficiency and ensure accurate record keeping.

#### 1.3 Overview:

The Library Management System enables librarians and members to manage books, track borrowings, generate fines and access reports.

### 2. General Description:

The system will serve librarians, staff and library members, ensuring streamlined operations. It will support searching, issuing, returning books and tracking overdue items.

### 3. Functional Requirements:

#### 3.1 Book Management:

\* Add, update or remove details.

\* Track availability status.

### 3.2 Member management:

- \* Register and Authenticate users.
- \* Maintain borrowing history.

### 3.3 Borrowing and Returning:

- \* Record issue and return dates.
- \* Automatically calculate fines for late returns.

### 3.4. Search and Reporting:

- \* Allow searching by title, author or ISBN.
- \* Generate reports on book circulation.

## 4. Interface Requirements:

### 4.1. User Interface:

- \* User-friendly interfaces for librarians and members.
- \* Accessible through web browsers and kiosk.

### 4.2. Integration interfaces

- \* Barcode scanner integration.
- \* Payment gateway for fine collection.

## 5. Performance Requirements:

### 5.1 Response Time:

- \* System should respond within 2 seconds.

### 5.2 Scalability:

- \* Support at least 10,000 books.

### 5.3 Data Integrity:

- \* Ensure consistency of records across all user.

## 6. Non-functional Requirements:

(ii)

6.1. Security: Role-based authentication

6.2 Reliability: Ensure 99.9% uptime

6.3 Scalability: Support large libraries

6.4 Portability: Support Windows / Linux.

6.5 Reusability: Modular design

6.6 Data Integrity: Accurate and updated records

## 7. Preliminary Schedule and Budget:

Estimated development time is 5 months with  
~~Q2 2018 budget of \$80,000.~~

## 8. Stock Maintenance System:

### 1. Introduction:

#### 1.1 Purpose of this Document:

The purpose of this document is to outline requirements for a Stock Maintenance System that manages inventory, sales, purchases and stock levels.

#### 1.2 Scope of this Document

This system aims to automate stock tracking prevent shortages, and provide business intelligence through reports.

#### 1.3 Overview:

The system will provide real-time monitoring of inventory, vendor management, low stock alert and automated reporting.

## 2. General Description:

The system will serve store managers, vendors and staff. It will help in minimizing losses and improving efficiency in stock handling.

## 3. Functional Requirements:

### 3.1 Inventory Management:

- \* Record purchase entries.
- \* Deduct stocks on sales.

### 3.2. Alerts and Notifications:

- \* Notify managers of low stock.
- \* Generate Expiry alerts.

### 3.3. Reporting:

- \* Daily, monthly and yearly stock reports.
- \* Vendor-wise purchase records.

### 3.4. User Management:

- \* Support Multiple Roles.
- \* Tracks user activities.

## 4. Interface Requirements:

### 4.1 User Interface:

- \* Dashboard for inventory overview.
- \* Graphic analytics for managers.

### 4.2. Integration Interface:

- \* Integration with POS systems.
- \* Barcode scanner report.

## 5. Non-Functional Attributes:

(13)

5.1 Security: Role-based authentication

5.2 Reliability: Ensure 24/7 availability

5.3 Scalability: Expandable for large businesses

5.4 Portability: Support desktop & web.

5.5 Compatibility: Compatible with POS hardware.

5.6 Data Integrity: Maintain real-time accuracy.

## 6. Preliminary Schedule and Budget:

Estimated development time is 6 months with  
a budget of \$120,000.

## 5) Passport Automation System:

### 1. Introduction:

#### 1.1. Purpose of this document:

The purpose of this document is to outline requirements for a Passport Automation system that enables online applications, verifications and passport issuance.

#### 1.2 Scope of this Document

The system will automate application submission, appointment scheduling, document verification, police verification and passport delivery.

#### 1.3 Overview:

The Passport Automation System integrates the government databases and verification services to reduce delays and minimize manual work.

## 2. General Description:

The system will serve applicants, verification officers and administrators, providing a secure platform for passport processing.

## 3. Functional Requirements:

### 3.1. Application Management:

- \* Online form submission with document uploads.
- \* Edit or withdraw applications.

### 3.2. Appointment Scheduling:

- \* Allow participants to book appointment slots.
- \* Notify applicants via SMS/email.

### 3.3. Verification:

- \* Integration with police verification databases.
- \* Document authenticity checks.

### 3.4. Passport Issuance:

- \* Generate unique passport numbers.
- \* Track application status.

## 4. Interface Requirements:

### 4.1 User Interface:

- \* Web portal for applicants and officers.
- \* Mobile support for notifications.

### 4.2 Integration Interfaces:

- \* Integration with police and government databases.
- \* Payment gateway for fees.

## 5. Performance Requirements:

### 5.1. Response Time:

- \* Page loads within 3 seconds.

### 5.2. Scalability:

- \* Handle millions of applications.

### 5.3. Data Integrity:

- \* Maintains consistency across all nodes.

## 6. Non-Functional Attributes:

### 6.1. Security: Strong encryption and multi-factor authentication.

### 6.2 Reliability: High availability and disaster recovery.

### 6.3. Scalability: Designed for national-level deployment.

### 6.4 Portability: Support web and mobile.

### 6.5. Usability: Simple forms for applicants.

### 6.6. Data Integrity: Ensure authenticity & accuracy.

## 7. Preliminary Schedule and Budget:

The deployment will take 12 months with a budget of \$2,000,000 due to large-scale deployment and security audits.

✓ 2/28