

## Application-01

### Hotel Management System.

#### Problem Statement.

The existing manual process of hotel management is slow and error-prone, leading to poor customer experience. A Hotel Management System [HMS] is needed to automate booking, check-in/check-out, billing and customer records. This will improve efficiency, accuracy, and service quality for both staff and customers.

#### Software Requirements Specification [SRS]

##### 1. Introduction

###### 1.1 purpose

The purpose of this document is to specify the requirement for a Hotel Management System. It describes the functionalities, interfaces, and constraints to automate hotel operations. The system aims to improve efficiency, accuracy and customer satisfaction.

###### 1.2 Document Conventions.

This SRS follows the IEEE 830-1098 Standard for Software requirement documentation

###### 1.3 Intended Audience and Reading Suggestions

This document is intended for project developers, hotel administrators, testers.

and stakeholders. Developers should focus on functional requirements, while managers may refer to the overall description.

#### 1.4 Project Scope.

The System will manage reservations, customer details, room allocations, and billing. It reduces manual workload and provides quick access to information for hotel staff and customers.

#### 1.5 References

IEEE 830-1998 Standard for SRS

### 2. Overall Description.

#### 2.1 Product perspective

The Hotel Management System is a standalone application connected to a central database. It replaces manual registers and provides a faster, more reliable way to handle hotel services.

#### 2.2 Product Functions.

The System provides functionalities like room booking, customer management, check-in/check-out, and bill generation. It also tracks room availability and generates reports for management.

## 2.3 User classes and characteristics

There are three main user groups: hotel staff (who operate the system daily), administrators (who manage records and reports), and customers (who may use a limited booking interface).

## 2.4 Operating Environment

The system will run on Windows or Linux platforms with an SQL database. It requires a basic computer setup and internet access for online booking support.

## 2.5 Design and Implementation constraints.

The system must use a relational database for storing records. Security measures like authentication and encryption are mandatory.

## 2.6 User Documentation.

The system will include a user manual and quick-start guide. Training sessions may also be provided to hotel staff to ensure smooth adoption.

## 2.7 Assumptions and Dependencies:

It is assumed that internet services and power supply will be available.

## 3. Specific Requirements.

### 3.1 Functional Requirements.

- The system should allow room reservation and cancellation.

- check-in / check-out → customers must be able to check-in / check-out through the system.
- automatic bill → It should support with tax calculation.
- room availability status → The system must show in real time.

### 3.2 External Interface Requirements

- User Interfaces → login screen, booking forms, and billing screens should be simple and interactive.
- Hardware Interfaces → The system should support desktops and printers for invoice printing.
- Software Interface → Must integrate with SQL databases.

### 3.3 Non-functional Requirements.

- Reliability → The system must perform consistently without crashes.
- Availability → It should be available 24/7 for customers & staff.
- Security → Data must be encrypted, and user login should be password-protected.
- performance → Response time should be less than 2 seconds per transaction.

## 4. Appendices

## Application-02 Credit Card processing.

### problem Statement

Online transactions often face issues such as fraud, delays, and errors due to insecure or inefficient payment handling. A Credit Card processing System is needed to securely authorize, validate, and complete Credit card transactions in real-time, ensuring customer trust and business reliability.

### Software Requirements Specification (SRS)

#### 1. Introduction.

##### 1.1 purpose

The purpose of this System is to provide secure, reliable and fast Credit Card transaction processing. It ensures that card payments are validated and completed with fraud protection.

##### 1.2 Document Conventions.

This document follows the IEEE 830-1998 SRS Standard.

##### 1.3 Intended Audience and Reading Suggestions

The document is intended for developers, financial institutions, testers, and Payment gateway administrators.

## 1.4 Project Scope.

The system processes payment through credit cards, validates details, ensures secure authorization, and generates receipts.

## 1.5 References

- IEEE 830-1998 Standard.

## 2. Overall Description

### 2.1 Product Perspective

The system is a service that integrates with online application & to process credit card transaction

### 2.2 Product Function

The system validates card details, checks balance / limits, authorizes transactions, and confirms payment with receipts.

### 2.3 User Classes and Characteristics

- Customers (who make payments)
- Merchants (who receive payments)
- Administrators (who monitor transactions)

### 2.4 Operating Environment

The system works on web-based application, requires internet, and uses SSL/TLS for secure communication

2.5 Design and Implementation Constraints.  
Must follow PCI DSS compliance.

### 2.6 User Documentation

User manuals and API integration guides will be provided.

### 2.7 Assumptions and Dependencies.

Assumes stable internet connection and active bank / payment gateway services.

## 3. Specific Requirements

### 3.1 Key Functional Requirements.

- Credit card payment authorization
- Debit - Prepaid Card → must perform card validation
- Transaction logging → should support for auditing.
- Must generate digital receipts → for every transaction
- Refunds and cancellation → should support

### 3.2 External Interface Requirements.

- User Interfaces → Payment form for entering card details, error messages, and receipt screen.
- Hardware Interfaces → optional POS terminal and receipt printers
- Software Interfaces → Integration with payment gateways and banking APIs.

### 3.3 Non-Functional Requirement

- Security → Must use encryption for card details
- Reliability → Transaction should not be lost or duplicated.
- Performance → Each transaction must complete within 5 seconds

## 4. Appendices.

## Applications - 03

### Library Management System

#### problem statement

Traditional library systems depend on manual book-keeping, which leads to inefficiency, misplaced records, and difficulties in tracking borrowed or returned books.

A library Management System [LMS] is required to automate book management, borrowing, returns, and user registrations to improve efficiency and accessibility.

#### Software Requirements Specification [SRS].

##### 1. Introduction

###### 1.1 purpose

The purpose of this system is to automate library operations including book issue, return, catalog management, and user records.

###### 1.2 Document Conventions

This SRS follows IEEE 830-1998 standard.

Requirements are classified into functional.

###### 1.3 Intended Audience and Reading Suggestion.

The document is meant for developers, librarians, system administrators, and students.

## 1.4 project Scope

The LMS manages books, users, borrowing and return operations. It will help librarians organize book efficiently.

## 1.5 References

- IEEE 830-1998 SRS Standard.

## 2. Overall Description

### 2.1 product perspective

The system will act as a central database application for storing library records, connected to a user-friendly interface for both Staff and Students.

### 2.2. product Functions.

The LMS will provide functions such as book search, borrowing, return, catalog update & fine calculation.

### 2.3 User Classes and Characteristics

- Librarians → Manage books
- Students/Users → Search and borrow books
- Administrators → System monitoring and reports

### 2.4 operating Environment

The System will run on desktops with database support (SQL) and may extend to web or mobile access.

### 2.5 Design and Implementation constraints

The System must use a relational database.

Authentication and role-based access are required.

### 2.6 User Documentation.

User manuals and quick guides will be provided for staff and students.

### 2.7 Assumptions & Dependencies

It is assumed that a stable internet or local network connection will be available.

## 3. Specific Requirements

### 3.1 Functional Requirements.

- Add, update, and delete book records
- Borrow and return books with due dates.
- Calculate and apply fines for late returns
- Generate library usage reports

### 3.2 External Interface Requirements.

- User Interfaces → login screen, book search form, borrowing / return panel.
- Hardware Interfaces → Desktop computers, barcode scanner
- Software Interface → SQL Database, Student record system.
- Communication Interface : LAN / Internet for remote access.

### 3.3 Non-Functional Requirements

- User Interfaces → logic screen, book search forms, etc.
- Reliability → System must not lose records during update
- Availability → Accessible during library hours (24/7 for online portal)
- Security → User authentication and role-based access control.

## 4 Appendices

## Application - 04

### Stock Maintenance System.

#### problem statement

Manual Stock management often leads to errors, misplaced ~~or~~ records, and difficulties in tracking inventory levels. A Stock Maintenance System is needed to automate stock-in, stock-out, and reporting processes, ensuring accuracy and efficiency in inventory control.

#### Software Requirements Specification (SRS)

##### 1. Introduction

###### 1.1 purpose

The purpose of the System is to automate Stock records, track item availability, and generate reports for efficient inventory control.

###### 1.2 Document Conventions.

This document follows IEEE 830-1998 SRS guidelines.

###### 1.3 Intended Audience

For use by store managers, warehouse staff and developers

###### 1.4 project Scope

The System manages stock-in, stock-out, supplier details, and alerts for low inventory

## 1.5 References

IEEE 830-1998 standard

## 2 Overall Description

### 2.1 Product perspective

The Stock Maintenance System is a standalone application with a database to track and manage inventory.

### 2.2 Product Functions

The system maintains stock records, generates reports and provides alerts when items are running low.

### 2.3 User classes

- Store Manager → oversees stocks
- Staff Members → update stocks entries
- Administrator → manages system settings

### 2.4 Operating Environment

Runs on desktop / mobile with support programming language

### 2.5 Constraints

Requires secure login, a stable database, and compliance with data accuracy.

### 2.6 User Documentation

User manual & quick-start guide will

be provided

## 2.7 Assumptions & Dependencies

Assumes continuous internet connection and functional hardware

## 3. Specific Requirements.

### 3.1 Functional Requirements.

- Add, update, and delete stock items
- Record stock-in and stock-out transaction.
- Generate daily / weekly / monthly
- Send alerts for low stock levels.

### 3.2 External Interface Requirements

- User Interface → Dashboard for stock
- Hardware → Desktop / mobile device, barcode scanner
- Software → SQL database integration
- Communication → LAN / Internet

### 3.3 Non functional Requirements

- Reliability → System must ensure data accuracy & consistency.
- Security → only authorized users can modify stock records
- Performance → stock updates should process within 2 seconds.
- Maintainability → easy to add new items, categories, and reports.

4

## Appendices

→ SQL

→ Stock in → Adding

→ Stock out → Removing.

## Application-05

### Passport Automation System

#### Problem Statement

The Manual process of passport application and verification often results in delays, errors and lack of transparency. A Passport Automation System is required to automate application submission, verification, and tracking ensuring faster processing & improved accuracy.

#### Software Requirements Specification [SRS]

##### 1. Introduction

###### 1.1 Purpose

The passport Automation System aims to automate the process of applying, verifying, and issuing passports. It will reduce manual errors and improve efficiency.

###### 1.2 Document Conventions

This document follows IEEE 830-1998 Standard for Software requirement Specifications.

###### 1.3 Intended Audience

This document is intended for applicants, passport officials, developers & system administrators.

## 1.4 Project Scope

The system allows online passport application, verification of details, appointment scheduling and status tracking.

## 1.5 References

IEEE 830 - 1998 Standard.

## 2. Overall Description

### 2.1 Product perspective

The system will act as an online platform integrated with a government database for passport processing.

### 2.2 Product Functions

It will handle new applications, renewals, verification, appointments, and issue status updates.

### 2.3 User classes

- Applicant → Applies for passport and tracks status
- Passport Office → Verifies applications and documents
- Administrator → Manages the system and reports

### 2.4 Operating Environment

The system will run as a web-based application with database connectivity and internet access.

## 2.5 Constraints

Requires secure authentication, compliance with government policies, and reliable servers.

## 2.6 User Documentation

User manual and online help will be provided for applicants and staff.

## 2.7 Assumptions & Dependencies

Depends on government ID verification

## 3. Specific Requirements

### 3.1 Functional Requirements

- Allow applicants to register and submit applications online
- Provide document upload & verification feature
- Enable appointment scheduling for verification/interview.
- Enable app status tracking for applicants.

### 3.2 External Interface Requirements

- User Interface → Online forms for application and status tracking
- Hardware → Desktop/mobile access, biometric scanners
- Software → Database, government verification APIs
- Communication → Secure internet connection (SSL/TLS)

### 3.3 Non Functional Requirements

- Reliability → The System must ensure data accuracy and integrity
- Security → Must encrypt personal data and follow government security standards.
- Performance → Application processing with 3-5 seconds per request
- Availability → System should be accessible 24/7

### 4 Appendices

PAS → Passport Automation System

SSL → Secure Socket Layer.