

1. Hotel Management System

1.1 Introduction:

1.1 Purpose of this Document: The purpose of this document is to outline the requirements & 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 & time required for the project.

1.3 Overview: The Hotel Management System will cater to the needs of ^{the} 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 tech expertise.

2. General Description: The hotel management system will cater to the needs of hotel staff and management, providing features such as room booking, 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
- Generate reservation confirmations and send notifications to guests.

3.2 Room Management:

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

3.3 Guest Management:

- Maintain guest profiles with personal information, preferences.
- Facilitate guest check-in & check out.

3.4 Billing & Invoicing:

- Generate accurate bills for charges & taxes.
- Accept various payment methods.

4. Interface Requirements:

4.1 User Interface:

- Intuitive & user friendly interface for hotel staff & guests.
- Accessible via all devices.

4.2 Integration Interfaces:

- Integration with payment gateways for secure transactions.
- Integration with 3rd-party booking platforms.

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.

5.3 Data Integrity:

Ensure data consistency & across all modules.

6. Design Constraints:

6.1 Hardware Limitations:

The system should be compatible with standard hotel hardware.

6.2 Software - Dependencies:

RDBMS for data storage, Java, SpringBoot.

7. Non-Functional Attributes:

7.1 Security:

Implement robust authentication & authorization for sensitive data.

7.2 Reliability:

Ensure high availability & fault tolerance to minimize system downtime.

7.3 Scalability:

Design the system to accommodate future growth & expansion.

7.4 Portability:

Support multiple platforms & devices for user accessibility.

7.5 Usability:

The system shall have a user-friendly interface with clear navigation.

7.6 Reusability:

The system shall use modular code design to facilitate future enhancements & maintenance.

7.7 Compatibility:

The system should be compatible with common web browsers.

7.8 Data Integrity:

The system shall ensure accurate & consistent data storage.

8. Preliminary Schedule & Budget:

The development of the Hotel Management System is estimated to take 6 months with a budget of \$100,000.

2. Credit Card Processing System

1. Introduction

1.1 Purpose

This document outlines the requirements for a Credit Card Processing System. It ensures secure, fast and reliable authorization, transaction processing, & settlement between merchants, banks & customers.

1.2 Scope

The CCPS will enable merchants to process payments, banks to authorize/settle transactions, and customers to make secure card payments. It supports both online and in-store transactions.

1.3 Overview

The system handles card authorization, fraud detection, payment settlement, reporting, and integration with banking/payment networks.

2. General Description

The CCPS provides merchants with payment processing capabilities, customers with secure transactions and financial institutions with settlement & reporting features.

3. Functional Requirement

3.1 Transaction Authorization

Verify cardholder details with issuing bank.
Approve/decline based on credit limit & fraud checks.

3.2 Payment Processing

Support credit, debit, and contactless payments.

Process refunds & chargebacks.

3.3 Security & Compliance

3.3 Security & Compliance

Encrypt sensitive data.

Implement 2-factor authentication.

4. Interface Requirement

4.1 User Interface

Merchant dashboard for transaction monitoring.

Customer-friendly payment pages (web/mobile).

4.2 Integration Interfaces

API's for merchant websites & POS terminals.
Integration with banking gateways.

5. Performance Requirements

Response time ≤ 2 seconds for transaction & authorization.
Handle at least 5000 concurrent transactions.
99.9% uptime.

6. Design Constraints

Must comply with PCI DSS standards.

Use RDBMS database.

Supported platforms - Web, mobiles.

7. Non-Functional Attributes

- 7.1 Security: End-to-end encryption and fraud detection.
- 7.2 Reliability: Fault tolerance & disaster recovery.
- 7.3 Scalability: Expandable to support global transaction.
- 7.4 Usability: Simple interfaces for customers.

8. Schedule & Budget

Estimated development: 8 months.

Budget: \$250000

3. Library Management System

1. Introduction

1.1 Purpose: Manage books, users & borrowing efficiently.

1.2 Scope: Supports

1.2 Scope: Supports catalog management, member records, borrowing/returning, and fines.

1.3 Overview: Provides librarians and students with an easy-to-use system for book management & access.

Functional Description:

Library management system provide an interface to easily manage the account of books issued, returned with its date, fines etc.

Functional Requirements:

Catalog: Add, update, search and remove books.

Users: Maintain student / faculty membership based access.

Borrow / Return: Track due date, apply fine, renewals

Reports: Usage stats, fines collected, popularity

Non-Functional Requirements:

Admin Dashboard: Manage books, users & report generation.

Student Portal: Search, borrow, return books.

Hardware Integration: Barcode / RFID scanner support.

Performance:

Response time should be < 2 seconds.
Process 1000+ users at a time.
99.9% uptime.

Constraints:

Cloud DB.

24x7 Desktop Support.

Integrate with library handbook.

7.

Non-Functional

7.1 Security: Role-based access control.

7.2 Reliability: Backup and recovery.

7.3 Scalability: Multi-branch support.

7.4 Usability: Simple, intuitive interface.

8.

Schedule & Budget:

Development - 5 months

Budget - \$80000

4. Stock Maintenance System

1. Introduction:

1. Purpose: Manage stock levels, purchase efficiently.

1.2 Scope: Tracks inventory, alerts and generates reports.

1.3 Overview: Provides businesses stock updates, supplier info & sales.

2. Functional Requirements:

2.1 General Description:

This will track your stocks and generate business insights and information about your profit as well as manage

3. Functional Requirements:

3.1 Inventory Management: Add, update, and remove stock items.

3.2 Stock Tracking: Monitor quantities, set reorder levels.

3.3 Sales & Purchase: Record sales, purchases and supplier details.

3.4 Reports: Stock levels, profit/loss, & usage trends.

4. Interface Requirements:

4.1 Admin Dashboard: Manage stock, suppliers & reports.

4.2 User Interface: For sales entry operators to update stock.

4.3 Integration: Barcode scanner support for quick updates.

5. Performance:

Response Time ≤ 2 seconds

Supports 1000+ concurrent users

99.99% uptime

6. Components:

Relational DB

All applications

Compatible with warehouse hardware.

7. Non-Functional Requirements

7.1 Security: Role-based access and encrypted transactions.

7.2 Reliability: Backup and recovery system.

7.3 Scalability: Expandable for multi-branch warehouses.

7.4 Usability: Intuitive, easy to learn.

8. Schedule and Budget:

Schedule: 6 months

Budget: \$100,000

5. P

5. Passport Authentication System

1. Introduction

1.1 Purpose: Automate the process of applying, verifying and issuing passports.

1.2 Scope: Handles applications, document verification, appointment scheduling.

1.3 Overview: Provides citizens, passport officers and administrators with efficient, secure systems.

2. Functional Requirements:

3.1 Application Management: Online app submission and editing.

3.2 Document Verification: Upload and validate required documents.

3.3 Appointment Scheduling: Book slots for interviews in person verification.

3.4 Status tracking: Real-time updates on officer progress.

4. Interface Requirements

4.1 User Portal: Citizens apply, upload docs to track status.

4.2 Officer Dashboard: Verify documents, reject docs.

4.3 Integration: Payment gateways for all fees.

5 Performance:

Response time ≤ 3 seconds.

Handle 1000+ concurrent users.

Database capacity for 2 million records.

6. Contracts:

Government data security standards must comply
FIPS 140-2

7. Non-Functional Requirements:

7.1 Security: Strong authentication, data encryption.

3

7.2 Usability: Highly available & fast service.

7.3 Scalability: Multitenant deployment, multiple offices.

Usability: All citizens can use.

8. Schedule & Budget

Schedule: 12 months → 1 year

Budget: \$500 000

S
C
2018/2019

2018/2019