

Software System Requirements Specification

for configuration Duck's Hotel

d1 Hotel management system

1. Introduction

1.1 Purpose of the document

This SRS document defines the functional and non-functional requirements for a Hotel management system. It serves as the blueprint for the development system outlining the existing capabilities and behavior of the system.

1.2 Scope of this document

Guest reservation

Room management

Inventory management

Payment

1.3 Overview

It is a web-based application accessible to hotel staff. This application has the ability to manage the day-to-day operations.

2. General description

2.1 User characteristics

- Hotel staff
- guests

2.2 User interface

- Easy to use
- ~~navigation should be good~~ Concise navigation
- Responsive design for different devices

2.3 Features

- Guest detail management
- Room management
- Payment Handling
- Staff scheduling
- Online Reservations

2.4 Benefits

- Time saving : tasks will be completed faster and the resources will be available for other activities
- Reduction of manual tasks
- Better Resource management
- Better communication

3. Functional Requirements

• Reservation Management

- Allow user to create, modify and update reservations

• Room management [database that should keep track of rooms that are occupied]

• Inventory management

• Staff management

- Scheduling shifts and tasks
- Create profile and manage Attendance

• Billing and payments.

4. Interface requirements

- User interface : A web based interface for guests and staff (management team)
- System interface: integrate with payment gateways (API)

5. Performance Requirements

- Response time
- Data security
- Availability

6. Design constraints

- technology stack
 - framework limitations
 - compatibility

- security : All the ~~if~~ important information should be encrypted and then sent to database

7. Non-functional Attributes

- Reliability
- Scalability
- Portability

8. Preliminary schedule and Budget

- No. of weeks for the gathering + system design, development, ~~deployment~~ integration, testing.
- Budget includes cost of project manager, business analyst, developers, software tools, documentation etc.

8. Preliminary schedule and budgeting support

- requirement gathering (2-3 weeks) of system
- system design (4-7 weeks)
- development (13-15 weeks)

→ testing (3-5 weeks)

Budget

personnel costs (\$12000)

software tools \$20000

Documentation \$1500

testing \$25000

GR
and other additional resources from DA ; physical &
existing infrastructure for integration

estimated budgeting costs :
physical \$ -
hardware \$ -
software \$ -
personnel \$ -

total estimated budgeting cost :
Personnel \$120000 ; hardware \$20000 ; software \$20000
hardware required in terms of server, memory
storage, networking

total estimated budgeting cost :
Hardware \$20000 ; software \$20000

Credit card processing system.

Q2

1. Introduction

1.1 Purpose of this document

This SRS defines all the requirements 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 CPS will be integrated with a payment gateway to facilitate secure and efficient transactions. It will provide a user friendly interface for merchants to manage their transaction and generate separate reports.

2. General description

2.1 User characteristics

- Merchants who will use system to accept cards
- Customers who swipe/use card.

2.2 User interface

- Easy to use & intuitive
- Secure and encrypted data transmission: the payments will be processed through a secure gateway and the data stored will be encrypted
- Support various card types: the user interface can accept any kind of credit card irrespective of its type

2.3 Features

- verify validity of credit card and availability of funds
- customer's authorization is must for transaction
- process a refund for cancelled transactions
- handle chargeback issued by a bank
- generate report.

2.4 Benefits

- easy payment process
- Reduced fraud risk
- Improved customer satisfaction
- Enhanced business efficiency
- saves time

3. functional Requirements

- Authorization
 - verify credit card no., exp date & CVV
 - check for sufficient funds
 - response message
- Capture
 - complete a transaction based on authorization
 - update transaction status and generate receipt
- Refund
 - process a refund for a captured transaction
- void
 - cancel a transaction before it is settled
 - update transaction status and generate receipt
- charge back
 - handle chargebacks initiated by banks or customers
 - gathers evidence and respond to req

4. Interface requirements

- Integrate with secure and reliable payment gateway
- Adhere to gateway's API standard
- 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 least downtime

6. Design constraints

- security must be implemented for sensitive data
- Industry standard must be followed
- the preferred programming language, frameworks
- database should be specified

7. Non-functional requirements

- Reliability
- Portability: Application should be running on every device.
- Availability

8. preliminary schedule and Budget

• schedule for this project :

- Requirement gathering (3-4 weeks)
 - system design (2-3 weeks)
 - development (10-12 weeks)
 - testing and validation (4-6 weeks)
 - deployment and training (2-3 weeks)
- total estimated weeks \Rightarrow (21-35 weeks)

Estimated budget for this project is:

- requirement documentation - \$1000
- design and tools - \$2000
- development - \$5000
- testing - \$2000

Total estimated budget - \$10000.

System architecture:

- client side: Java Swing application
- server side: MySQL database
- internet connection
- hardware: Dell OptiPlex 7010
- operating system: Windows 7

Functional aspects:

- User registration and login
- Book management (add, remove, update)
- Member management (add, remove, update)
- Transaction management (borrowing, returning, overdues)
- Reporting and statistics

Non-functional requirements:

- User friendly interface
- Database integrity
- Data security
- Scalability
- Reliability

Tools & technologies:

- Java (language)
- MySQL (database)
- Eclipse (IDE)
- MySQL Workbench (management tool)
- JUnit (testing framework)
- Git (version control system)
- Jenkins (CI/CD pipeline)



Library Management System

1. Introduction

1.1 Purpose of this document

The SRS defines all the requirements for a library management 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

The LMS will handle tasks such as book inventory management, member registrations, book borrowing / returning.

1.3 Overview

- Books: Inventory, availability and categorization
- Members: Registration, history of borrowed books
- Transactions: Borrowing, returning and overdue management

2. General Description

2.1 User characteristics

- Librarian: Manage books, members and transactions
- Members: Borrow books, return them and manage their own accounts

2.2 User interface

- Ease of use & Intuitive design: Easy navigation through the system, clear labels, buttons and minimal clicks will be implemented

- support for various actions
 - librarian interface
 - member interface
- consistency in design: same kind of design lang.
- eg layout across all pages.

2.3 Features

- ① Book management - Add, update, remove and categorization
- ② Member management
- ③ Borrow/Return system: Process book borrowing and returning with automatic due date and fine calculation
- ④ Search functionality : Based on category, title, book number

2.4 Benefits

- ① Efficiency - reducing human errors of tracking, remembering/borrowing/writing/returning, and calculation
- ② time-saving - automated transaction processing, inventory management and report generation

3. Functional Requirements

- Authentication and Authorization - Admin and librarian logins, members sign up etc
- Book management - add, edit, delete, borrow, return
- member management : - Register new members
- Edit member information
- Transaction management ; - Borrow books
- Return books
- transaction reports

4. Interface Requirements

- userfriendly interface
- Integrate with secure and reliable database

5. Performance Requirements

- Response time : Should be near about 5 seconds
- concurrent users: system should support atleast 150 concurrent users
- data processing : report generation, updates should be generated in real-time

6. Design Constraints

- technology stack: Preferred programming language, frameworks, database should be specified
- data integration and ACID (Atomicity, consistency, isolation, durability) properties must be ensured

7. Non-functional requirements

- security

- scalability

- Usability

8. Preliminary Schedule and Budget

8.1 Development Schedule

- Requirement gathering : 2 weeks
- Design : 3 weeks
- Implementation : 4 weeks

SB2 Estimated Budget

• Hardware : \$1000
• Software tools and licensing : \$1000
• developer cost : \$5000

Stock maintenance system

1. Introduction

1.1 Purpose of this document:

This SRS defines the functional and non-functional requirements for a Hotel management system. It serves as a blueprint for the development team outlining the expected major features.

1.2 Scope of this document

The system will handle authorization, stock management, inventory tracking (monitoring stock level) in real time, report generation.

1.3 Overview

The system offers functionality to help business maintain accurate stock levels, streamline procurement and avoid stockouts or excess inventory.

2. General description

2.1 User characteristics

- Stock Managers: Responsible for adding, updating and tracking stocks

- Employees: can view stock details, perform stock checks and report low stock levels

- Admin

2.2 User interface

- Ease of use

- Search and filter

- Dashboard: shows current stock levels

2.3 Features

- Real-time Inventory updates; stock levels are updated automatically as items are received or sold
- Report generation: Detailed reports on inventory movement, stock valuation and supplier performance
- Supplier Integration: track supplier details and manage purchase order

2.4 Benefits

- Increased Efficiency
- Improved Accountability

3. Functional Requirements

① Stock Management

Add new stock items

- update stocks, delete stocks, Add stocks

- Low Stock Alerts

② Stock Transactions

- Receive stock

- Dispatch stock

③ Report Generation

4. Interface Requirements

- User login interface: provide secure access to different roles with a unique username and password

- Stock search interface: allowing users to search for products using SKU, product name, category and supplier

- ~~Report generation~~

⑤ performance Requirements

- Response time

within 1 second as per user requirements

- concurrent users

support up to 1000 concurrent users

- Data processing

support up to 1000 passport records per hour

⑥ design constraints

- technology stack

- security

- usability standards

⑦ Non-functional Requirements

- maintainability

- portability

- Reliability

⑧ preliminary project schedule and budget

8.1 Development Schedule

- Requirement gathering : 2 weeks

- System design : 3-4 weeks

- development and testing : 6-8 weeks

- deployment : 1 week

8.2 Estimated budget

- Hardware : \$1000

- Software tools and licensing : \$1500

- developer cost : \$5000

- travel and hospitality

- other project costs

Passport Authentication system

1. Introduction

1.1. Purpose of this document

This SRS document defines the functional and non-functional requirements for a passport Authentication system. It serves as a blueprint for the dev team, outlining the expected capabilities and behaviour of the system.

1.2 Scope

① User Authentication

② passport verification

③ data validation : cross-check passport data against a central database of passport records

④ fraud detection : identify anomalies such as forged documents

The system will integrate with national & international databases to ensure passport data is up-to-date and accurate.

2. General description

2.1 User characteristics :-

• Government officials : Responsible for using the system to verify passport details

• Border control Agents : Use the system to validate passports at entry/exit points

• Airline staff

2.2 User Interface

- Ease of use
- search and verification
- dashboard: displays the latest security rules, update rules, and transaction history

2.3 Features:

- Realtime verification
- Multilanguage support

2.4 Benefit

- Increased security
- fraud prevention
- Global integration

3. Functional Requirements

- Validation Against database: the system data will check the passport data against national & international databases to verify its authenticity
- Alert system:
 - Expired passport Alert
 - forged passport Alert

4. Interface Requirements

- user-login Interface
- passport verification Interface

5. Performance Requirements

- Response time
- concurrent users
- data base queries

6. Design Constraints

- technology stack
- security (global security standards)
- Hardware Requirements

7. Non-functional Requirements

- maintainability
- portability
- Reliability

8. Preliminary Schedule and Budget

- Requirement gathering: 2-3 weeks, \$500
- System design: 3-4 weeks, \$1000
- Development time and testing: 6-8 weeks, \$3500
- deployment: 2 weeks, \$500
- Hardware Requirements: \$500
- Software tools and licensing: \$1000

Class Identification

① Hotel management system

- Hotel: Represents a hotel entity including details like name, location, rating
- Room: Represents room 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.
- Service: Includes service provided by the hotel (e.g.: laundry, room service).

② Credit card processing system;

Cardholder: Stores details of the credit card owner

credit card: It includes number, expiration date and limits

Transactions: Handles individual card transactions, such as purchase or refunds

Payment gateway: facilitates the processing of transactions b/w merchant and bank

Merchant: Represents merchants that accept credit card payment

Authorization: Manages authorization check for card transaction

* Account statement: generates monthly statements showing card usage

Statement: tracks the total amount spent by the user

③ Library management system:

- Book: Represent book, title and availability
- Member: stores details about library members
- Librarian: manages library operations, including adding /removing book and members
- Borrow transaction: tracks book borrowing transactions due dates and returns
- Fine: calculate and stores overdue fines for late book returns
- Category: organize books by category or genre
- Reservation: Handles the reservation of books by members

④ Stock Maintenance System

→ Product: Represents each product in the stock, includes name, description and price

→ Stock: tracks stock levels and details like quantity available, delete, upgrade

→ Supplier: contains details of supplier who provide products

→ Customer: stores customer information and purchase history

→ Invoice: Handles billing and payment for stock purchases

→ Report: generates reports on stock levels, sales and reordering needs.

- ③ passport Automation system:
- Applicant: stores details about the passport applicant (name, address, ID etc)
 - passport: Includes details of passport issued i.e. validity and passport number
 - Application: manages the application process, including submission, processing etc
 - fee payment: Handles payment details for application
 - Renewal: Handles passport renewals and associated processes