

**Sir Syed University of Engineering and Technology Software Engineering Department Introduction to Software Engineering – Practical**



**Software Requirements Specification**

**Hotel Management System**

**Team Members**

**Name: Ahyan Aslam**

**Roll No: 2024F-BSE-042**

**Role: Project Manager**

A person wearing a blue head covering

AI-generated content may be incorrect.

**Name: Ifrah Pervaiz**

**Roll No: 2024F-BSE-037**

**Role: Managed Documentation**



**Name: Muhammad Ashar**

**Roll No: 2024F-BSE-041**

**Role: Researcher**



**Name: Batool Hamza**

**Roll No: 2024F-BSE-337**

**Role: System Analyst**

Table of Contents

**1. Introduction**

1.1 Purpose

Describe the purpose of this SRS document and the intended audience.

1.2 Scope

Briefly describe what the system will do and its main features.

1.3 Definitions, Acronyms, and Abbreviations

List all relevant terms and their definitions.

1.4 References

List any references used to create the SRS (books, articles, standards, etc.).

1.5 Overview

Briefly describe what each section of the document contains.

**2. Overall Description**

2.1 Product Perspective

How does this product fit into a larger system, if at all?

2.2 Product Functions

Summarize the major functions the system will perform.

2.3 User Classes and Characteristics

Describe the different types of users expected to use the system.

2.4 Operating Environment

Hardware, software, and other platforms required to run the system.

2.5 Design and Implementation Constraints

Any limitations on the design (e.g., regulatory policies, hardware

limitations).

2.6 User Documentation

User manuals, online help, or other user support documents.

2.7 Assumptions and Dependencies

Any assumptions or dependencies that the project relies on.

**3. Scheduling (Gantt Charts, Resources)**

2.1 Gantt Chart of your Whole Project

2.2 Pert Chart

2.3 Network Diagram

2.4 Resource Sheet

**4. Specific Requirements**

3.1 Functional Requirements

List all functional requirements with unique identifiers (e.g., FR1, FR2...)

3.2 Non-functional Requirements

Performance, reliability, security, scalability, etc.

3.3 Use Case Descriptions

Provide use case descriptions with actors, steps, and alternative flows.

3.4 System Features

Describe each system feature and its priority.3.5 Diagrams

➢ Use Case Diagram

➢ System Flow Diagram (Flow Chart)

➢ Class Diagram

➢ Sequence Diagram

➢ Mind Mapping (diagram)

**5. External Interface Requirements**

4.1 User Interfaces

Screens, menus, forms (use mockups or descriptions)

4.2 Hardware Interfaces

Interaction with hardware components

4.3 Software Interfaces

Interaction with other software components

4.4 Communication Interfaces

Protocols and methods for external communication

**6. Other Non-functional Requirements**

5.1 Performance Requirements

Speed, response time, etc.

5.2 Security Requirements

Login, data protection, etc.

5.3 Quality Attributes

Maintainability, usability, portability, etc.

**7. Appendices**

Additional information, forms, surveys, mock data, etc.

1. **Introduction:** 
   1. **Purpose:**

The goal of this document is to explicitly describe the functional and non-functional requirements of the Hotel Management System being designed as part of the ITSE course project. This Software Requirements Specification (SRS) is a formal contract between the development team, the instructor (as the client), and any future stakeholders who might review or extend the system.

This SRS assists in ensuring that the same requirements and functionalities are known to all the team members. It gives a foundation to the developers from where they can begin their work with clarity, ensures that the testers are aware of what should be tested, and keeps the project on track with its goals. The document will also prove useful in the future if any upgrades or modifications are needed because it has all the basic details.

This system is meant to automate and simplify major hotel functions It allows them to check room availability, book rooms, order food, request room service, and check out all in one place. For the admin or hotel staff, it provides tools to manage rooms, handle food menus, and view customer requests efficiently. The idea is to cut down on manual work, reduce errors, and create a smoother, more convenient experience for everyone involved in running or staying at the hotel. The document contains extensive use cases, diagrams, assumptions, and user interface descriptions. Through the creation of this SRS, we seek to avoid misunderstandings, handle expectations, and provide a seamless development process from beginning to end.

As our system is for human resources, the SRS ensures we keep our priorities in line with their speed, reliability, simplicity, and structured recordkeeping. Any feature we design will be directly based on what's outlined here, so this document is essentially the basement of the entire project.

* 1. **: Scope:**

Hotel Management System (HMS), created for the purpose of this project is a desktop application designed to assist hotel staff in performing daily activities in a systematic and effective manner. The overall aim is to transition regular hotel operations from paper-based or manual systems to a computerized system that is more efficient, reliable, and manageable.

This is a single-branch-only system intended to be used in a single hotel branch and is not targeted for chain hotels or multiple locations. The design is aimed at producing a simplified but working simulation of an actual hotel front desk system. It is being designed as a course project under time and academic limitations, so the scope is narrowly defined to encompass all significant operations without exceeding what can be reasonably developed and tested within the semester.

The primary features included are:

* + 1. **Customer Management:**

Staff can check in or out and edit or insert new customers. All relevant information, including ID, contact information, and room assignment, will be stored in the system.

* + 1. **Room Booking and Availability:**

Users will be able to display real-time room status (available, booked, under maintenance), assign rooms to customers, and cancel bookings if needed.

* + 1. **Staff Records Management:**

Administers also have access to a part where they can add new employees, set roles (e.g., receptionist, housekeeping), and view active workers.

* + 1. **Billing and Payments:**

Whenever a client checks out, the system will automatically compute their total billings based on their length of stay and services consumed. Payment status (e.g., Paid, Pending) will be manually input by employees.

* + 1. **Login System:**

Admin and Receptionist will have independent login access. Admin-level users can add rooms, maintain staff, and modify some settings.

* + 1. **Report Generation:**

The system will automatically prepare daily reports comprising number of customers served, rooms booked, and earnings. Admins can see the reports for improved decision-making.

* + 1. **View Menu and Food Ordering:**

Guests can browse the hotel food menu via the system interface. Receptionists can place orders for guests, and the charges will be billed to their room for payment at check-out time.

* + 1. **Service Requests:**

Guests may request extra services like laundry, housekeeping, or maintenance. These requests will be entered against their room record and forwarded to the staff concerned for follow-up.

The aim is to design a system that is practical and realistic, with emphasis on the primary processes that take place at a hotel's reception and admin desk. The features have been selected according to feasibility, complexity, and relevance from an academic perspective, in consultation with the team and instructor.

* 1. **Definitions, Acronyms, and Abbreviations:**

|  |  |
| --- | --- |
| **Terms** | **Meanings/Abbreviations** |
| SRS | Software Requirement Specification is a document which defines the purpose, features and functionality of the system. |
| HMS | Hotel Management System |
| ID | Identification, it can be a String or Integer but always unique |
| UI | User Interface, the Interface which can be experienced by user to use the system |
| FR | Functional Requirements are those requirements which are required so the system can work properly |
| SMS | Short Message Service, it is a text messaging service used to send or receive short messages. |
| CI CD Pipelines | It an automated workflow followed to develop a system. |
| UC | Use Case describes what the systems works. |
| SQL | Structured Query Language is a programming language used to store information in database. |
| HTTPS | Hypertext Transfer Protocol Secure used to send data between a web browser and a website. |

* 1. **References:**
     1. **GeeksforGeeks - Software Requirements Specification (SRS):**

This website provides format, structures and examples of Software Requirements Specification (SRS) on a basic and advanced level. A conceptual guidance was taken from here in creating this document.

* + 1. **SRS Sample Document:**

Example SRS document shared by the instructor was used to create the base of the structure of this report. Formatting, section headings, and content flow is followed by the help of this.

* + 1. **TutorialsPoint – Software Requirements Specification Guide:**

From this online source additional clarity and explanation was taken, especially about functional vs non-functional requirements and use-cases.

* 1. **Overview:**

This Software Requirements Specification (SRS) document is detailed guide for developing the Hotel Management System (HMS). It is meant to present a clear and structured description and understanding of both functional and non-functional requirements of the software. This process helps for smooth communication among clients, developers, testers, and project stakeholders. The main goal of this SRS document is to minimize confusion, ensure that user expectations and needs align with system behavior, and highlight the development process. The proposed system aims to control the follow of daily hotel operations, which include room booking, customer check-in and check-out, food menu, ordering food, room request service and payment handling, and administrative tasks. The SRS details all expected user interactions, data flows, system limitations, and performance standards that the final product must meet. This document is divided into several organized sections, each section focuses on a specific task or aspect of the system. The Introduction section outlines the project goals, intended audience, and the system's scope. It sets clear expectations for what the system should achieve. The Overall Description provides an overview of the system environment, user roles such as Admin and Customer

## **Overall description**

* 1. **Product perspective:**

Hotel Management System is a software that automate daily activities of hotel.It is a computer based system that is made to help staff and management handle daily tasks in a simple and fast way. In many hotels, work is still done manually. Staff have to write down bookings in notebooks, prepare bills by hand, and store files in cabinets. This takes a lot of time and can cause mistakes. With HMS, most of these tasks become automatic.It assists in having all records in one spot, where it can be accessed when required. It is capable of being used independently or can be linked with other tools such as online booking sites or bookkeeping software.For instance, when a customer reserves a room, the system stores the information, updates the status of the room, and even generates a bill. In case it is required, it is capable of sending a confirmation message.. All of this is done in a few clicks. It reduces the load on staff and improves the customer experience. The system can also be updated later if the hotel adds more branches or services.It is made in such a way that it can work alone or be connected to other tools like online hotel booking websites, accounting software, or customer review systems in the future.The system replaces manual processes that were slow, error-prone, and required a lot of paperwork..By using this software, hotel staff will save time, reduce errors, and provide better service to customers.

The system will consist of:

* A login portal for Admin and Receptionist
* Room management interface
* Customer record handling
* Billing and payments
* Report generation

This software does not rely on any external system but can be integrated with online payment gateways or booking websites in future upgrades.

* 1. **Product Functions:**

The system offers many main features.It enables customer to reserves rooms,look up status and cancel reservations.Hotel staff can control room status and update status.The admin can handle staff accounts and change system settings. The billing system creates invoices, applies taxes or discounts, and prints receipts. Reports are generated to show daily and monthly performance.

HMS will perform the following main tasks:

**Room Booking Management:**

**The system enables the front desk to determine the availability of a room, book rooms for guests, and reschedule or cancel bookings if need be. Every booking record is stored with the customer's information and duration of stay.**

**Room Categories and Types:**

**HMS maintains data on various types of rooms, i.e., single, double, and suite, and their amenities such as air conditioning, bed number, attached bathroom, etc.**

**Cleaning and Maintenance Monitoring:**

**The software updates and monitors the cleaning status of rooms.**

**Billing and Receipt Printing:**

It determines the amount of money for the stay, including taxes and discounts if applicable. On payment, the system prints and keeps a record of the receipt which is stored in records.

**Storage of Customer Records:**

It keeps guest details and past booking history for future use, which is useful in handling regular customers.

**Daily and Monthly Reports:**

Automated reports of bookings, income, vacant rooms, and staff activities can be created by the hotel management. These are useful for performance assessment.

**User Role Control:**

Different roles and permissions can be allocated to every member of staff by the admin. The receptionist might view only bookings, whereas an admin might view all reports and settings.

**Confirmation Warnings:**

Upon reservation, the system can automatically send a confirmation e-mail or SMS to the customer.

**Room Status Summary:**

Gives a simple visual overview of all room status in real-time (booked, empty, in cleaning, etc.).

**System Settings and Configuration**:

Admin can revise tax rates, room rates, discount promotions, and seasonal rates via the settings panel.

**Integration Facilities:**

HMS can be integrated with accounting software, website booking sites, or customer review websites for seamless working.

The system has numerous core features. It can facilitate booking of rooms, check room availability, and cancellation of bookings. The room status can be managed by the hotel staff and information updated. Staff accounts are managed by the admin, and system settings are altered. The billing system generates invoices, applies to taxes or discounts, and prints receipts. It generates reports to display daily and monthly performance.

* 1. **User Classes and Characteristics:**

Various users will use the Hotel Management System, and each will have certain roles, responsibilities, and permissions to access. Knowing these classes of users enables the design of the correct interface and level of access for the given user category.

**2.3.1Receptionist/Front Desk Staff:**

They are tasked with carrying out day-to-day customer interactions like booking, check-in, and check-out processes.

They utilize the booking module to input customer information, allocate rooms, and verify payments.Usually, they have basic knowledge of computers and can navigate simple interfaces.

They are trained internally by the hotel and usually work in shifts.

* + 1. **Manager/Admin**:

The hotel manager or administrator has full control over the system.

They can create and manage staff accounts, view reports, manage pricing, and apply discounts.

They also have access to the system settings and can modify system-level configurations.

Managers generally have prior experience with such systems and require training only for specific functions.

* + 1. **Cleaning/Maintenance Staff**:

These users do not handle bookings but update the status of rooms.

They use the system to check which rooms need cleaning or repairs.

They access the system through mobile devices or tablets if available.

No complex training is needed; only simple operational training is provided.

**2.3.4 Customers (Optional Web Interface Users)**:

* **Guests can interact with HMS either using an online portal or self-service.**
* **They can check room availability, do bookings, and get confirmation messages.**
* **The system is made extremely user-friendly to serve customers with limited technical capabilities.**
* **All types of users will be granted the appropriate level of access to secure and ensure free movement of processes in the hotel**

**2.4 Operating Environment:**

The system can run on any computer with Windows. It will also need a basic web browser like Chrome or Edge. A local database will be used to save all the records safely. The hotel’s computers must be connected through a local network if used by multiple staff members at the same time.

**The HMS requires a constant and reliable technical environment to function optimally. The operating environment comprises both hardware and software needs, along with other critical infrastructure necessary for the system to function normally.**

**2.4.1 Hardware Requirements**:

* Front desk and admin office desktop computers or laptops.
* At least 4GB RAM and a dual-core processor for smooth performance.
* A printer to generate customer bills and reports.
* Barcode scanners or card readers (optional) for faster check-in/check-out.
* UPS or backup power system to handle power failures.

**2.4.2 Software Requirements**:

* Operating System: Windows 10, Linux, or later versions.
* Web browser (Chrome, Firefox, etc.) if using a browser-based version.
* Database Management System: MySQL or PostgreSQL.
* HMS software installed either locally or accessible through the cloud.

**2.4.3 Network Requirements**:

A stable internet connection is necessary for cloud access or online booking integration.

Local LAN connection within the hotel for internal communication.

Basic firewall and antivirus protection for security.

**2.4.5 Other Dependencies**:

* Software must be compatible with local printers.
* Daily backup systems should be in place for data safety.
* IT technician or support staff should be present in the event of technical malfunctions.
* Such an environment enables efficient utilization of the system with minimum downtime and maximum ease of access to everyone.

**2.5 Design and Implementation Limitations:**

While designing and building the Hotel Management System, there are some limitations that the development team needs to consider. These are either due to customer requirements, technical limitations, or external regulations.

**2.5.1 Security Regulations:**

The system should ensure that there is no unauthorized use of personal or financial data. Only specific users with legitimate login details should be able to access specific activities.

**2.5.2 Limited Equipment:**

The system should be simple enough to operate on normal equipment because not all hotels possess high-end machines.

**2.5.3 Offline Availability:**

The system should function normally even when internet access is temporarily lost. All critical operations should persist in offline mode and then sync data when internet returns.

**2.5.4 Role-Based Access:**

Not all users may access all areas of the system. E.g., cleaning staff must not be able to modify guest information.

**2.5.5 Data Privacy Laws:**

The system must save customer information securely, complying with local data protection laws.

**2.5.6 Time Limitations:**

The system has to be implemented within a time frame, so all the features must be well planned and implemented in time.

**2.5.7 Budget Restrictions:**

The price of developing and keeping the system running within the budget of the hotel, so unnecessary or additional features should be omitted.

These limitations make the development team concentrate on what is actually necessary and prevent errors or delays.

**2.6 User Documentation:**

User manuals will be provided for both staff and admin. These will help them understand how to use the system. Help sections or tool tips may also be added inside the system itself for easier use.

To help users understand and work with the Hotel Management System easily, clear and simple documentation will be provided.This guidebook can be printed or can also be sent digitally.

**2.6.1 User Manual:**

Describes how every feature functions with simple steps and snapshots. It assists receptionists, cleaners, and admins in operating the system correctly.

**2.6.2 Quick Start Guide:**

A concise document with just the primary steps such as login, booking a room, and printing an invoice.

**2.6.3 Online Help Section:**

Available online, users simply click a Help button on every screen to receive brief descriptions of what that section of the system does.

**2.6.4 Troubleshooting Tips:**

Contains solutions to frequent problems such as login failures, slow loading, or printer misbehavior.

**2.6.5 FAQs:**

Responses to frequent questions by staff. For example: How to cancel a booking? or How to update a room's status?

**2.6.6 Training Videos**:

Optional short videos to show how to use important features of the system. Can be helpful for new employees.

**2.6.7 Support Contact Information:**

When users still experience problems, they can reach the IT support via phone or email.

This manual guarantees users are not perplexed and can utilize the system proficiently without any expert assistance.

**2.7 Assumptions and Dependencies:**

In every software project, some assumptions are made, and the system may depend on certain things being available or true. For the HMS, the following points are important:

**2.7.1 Assumptions**:

Hotel staff using the system have basic computer skills like using a mouse and typing.

The hotel management is willing to train the staff for 1-2 days before using the system.

Internet will be available for online bookings and email confirmations, at least during working hours.

**All hotel rooms are accurately numbered and classified in the system.**

**2.7.2 Dependencies**:

The system will be dependent on an active internet connection for functionalities such as email confirmation, web bookings, and distant support.

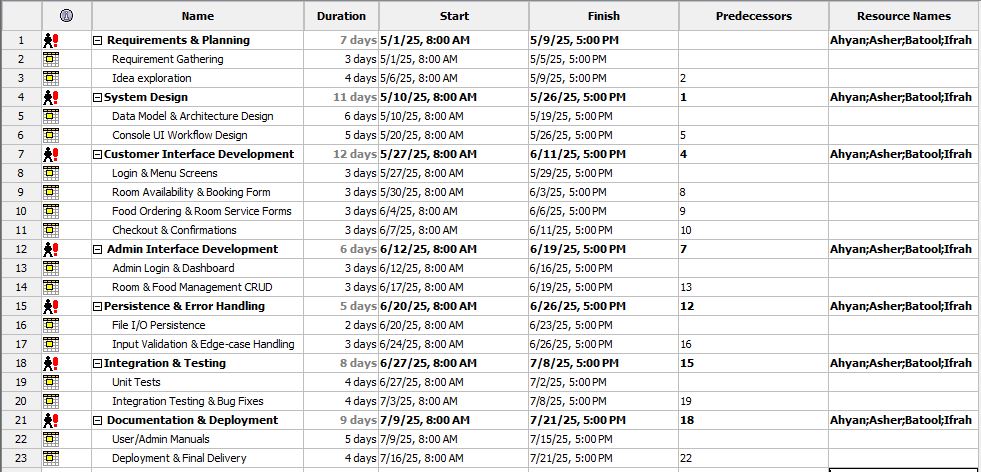
Printers and other hardware like barcode readers must be installed and configured.

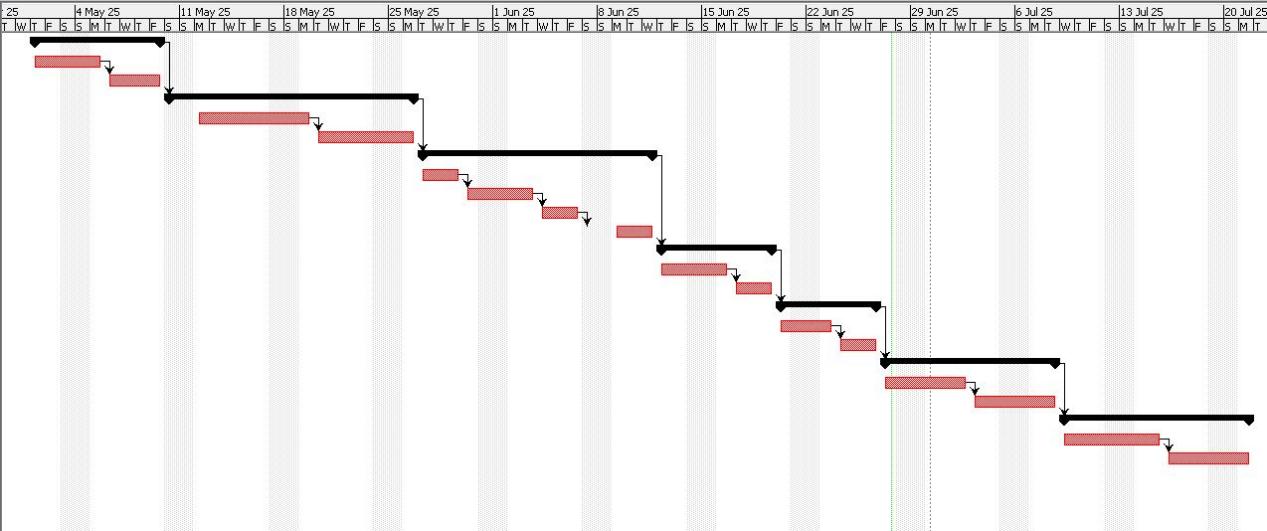
Regular backups need to be scheduled to avoid loss of customer or booking information.

At least one individual with admin access needs to be present at all times for system management and for emergency updates.

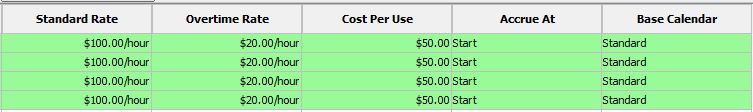
**3. Scheduling (Gantt Charts, Resources)**

**3.1 Gantt Chart:**



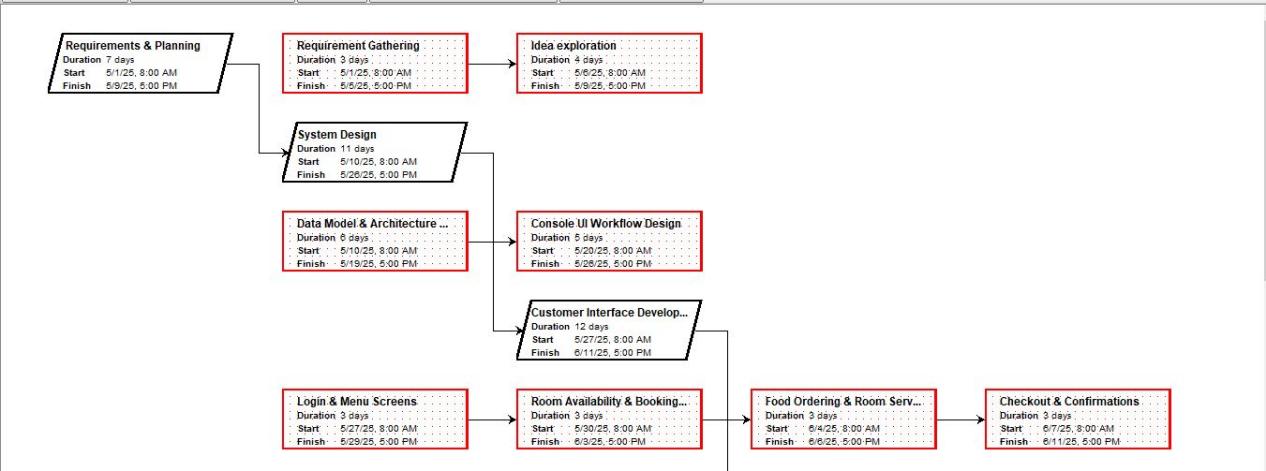


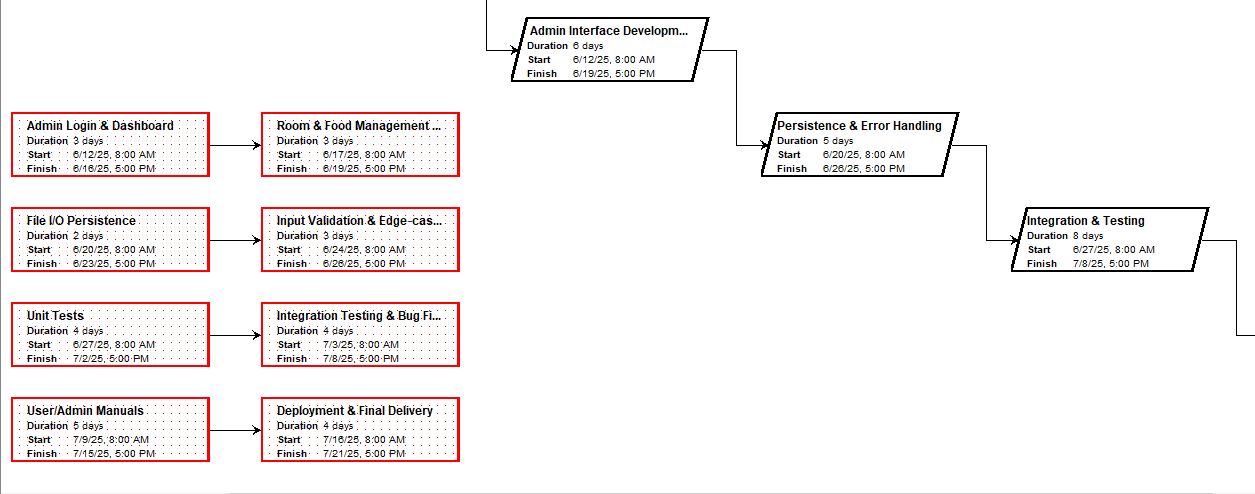
**3.2 Resource Sheet:**





**3.3 Pert Chart:**





**4. Specific Requirements**

**4.1 Functional Requirements:**

Functional Requirements are those specific actions or works which are performed by system. Some of the major functions of the hotel management system are given below:

**4.1.1 FR1: User Registration**

The system will provide the facility to create new accounts for the users. Users will have to enter their name, email address, phone number and password to complete the form for registration and then the information will be securely stored in the database.

**4.1.2 FR2: User Login**

Users can get logged in by entering a username or email address and password to access the dashboard.

**4.1.3 FR3: View Available Rooms**After login, user will be able to see the real-time availability of rooms and it can be filtered to search according to the different room type, price rage and dates.

**4.1.4 FR4: Room Booking**Now the user can book available room by selecting the desired check-in and check-out dates. Booking will get confirmed through the SMS/ email according to the given personal information. Then all this information will be updated and stored in the database.

**4.1.5 FR5: Cancel Booking**

Users can cancel their existing booking any time before 24 hours of the check-in date. On cancellation, he would see the confirmation module and then the cancelled room will be updated in the available room’s list.

**4.1.6 FR6: View Booking History**

Users can see their paste and active booking details anytime including booking dates, room number, status and total charges. This is important for feature transparency for users to see their complete stay history.

**4.1.7 FR7: View Food Menu**

Users can access the complete details of food items on the menu with names and prices. The menu will be daily updated, and it will be connected from the backend from which the prices and availability can be modified.

**4.1.8 FR8: Order Food**

User can order food in their room. System will show the status if the order is pending, preparing or delivered. After order confirmation, system will notify the kitchen or food department and update the real-time order status.

**4.1.9 FR9: Request Room Service**

Room service can be requested like cleaning, towel replacement, laundry etc. Admin will get notified about the request and the history will be shown in the dashboard.

**4.1.10 FR10: Admin Login**

Admin will be logged in by using their secure credentials to access dashboard. He can only access authorized modules to access rooms, food, booking, and to manage users.

**4.1.11 FR11: Add/Edit/Delete Room Details**

Admin can add new rooms in the system and can also update or delete the existing room’s price, type, status.

**4.1.12 FR12: Manage Food Menu:**

Admin can add new food items to the menu and can also update or delete the existing items.

**4.1.13 FR13: View All Bookings (Admin)**

Admin can view all bookings in the system and can also access user’s details, room information, status, and booking times.

**4.1.14 FR14: Generate bill**

The system generates bills automatically whenever a user checkout contains room rent, food charges and room service charges.

**4.1.15 FR15: Logout**

User admin can log out from the system whenever they want.

**4.2 Non-Functional Requirements:**

During the development of the hotel management system, there are some non-functional requirements which do not only define the functionality of the system, but it also shapes its structure, design quality, compatibility and extensibility. There are son-functional requirements given below which must be done during the development of the system:

**4.2.1 Architecture and Design Requirements**

* The design of the system will be based on layered architecture in which presentation, logic and data layers will be separated.
* Modularity will be ensured so that every component (like room module, booking module, food module) can work independently.
* Losse coupling principles should be followed to make the system flexible.
* Common utility classes should be made to promote reusability.

**4.2.2 Compatibility Requirements**

* UI of the system must be compatible with multiple browsers (like Chrome, Firefox, Safari, Edge).
* Frontend framework should be developed in a manner that it can follow mobile-first responsive design.
* The backend of the system should be on the latest server so that it can be deployed easily.

**4.2.3 Data Modeling Requirements:**

* The design of the database should be in normalized form so that redundancy can be avoided and efficiency can be increased.
* Entity relationships should be defined clearly.
* Primary keys, foreign keys and indexing should be used to optimize performance.

**4.2.4 Development Environment Requirements**

* Standard IDEs should be used to develop the system (e.g. VS Code).
* Git should be used for code repository version control and GitHub will be used for collaboration.
* CI CD pipelines will be used so that bills generate bills can generate automatically.

**4.2.5 coding standards and documentation requirements**

* All source code files will follow naming conventions and file structure.
* Code comments top strings and function headers will be written in every module so that it can be easy to understand.
* The module must contain separate technical documentation in which it's classes, methods and data flow will be explained.
* UML diagrams like class diagrams and sequence diagrams will be included in project documentation

**4.2.6 testability requirements**

* Every module should be written in a manner so that unit testing can be done.
* Dummy objects will be used Show that back-end systems should also be used for isolate testing.
* Testing coverage should be maintained 80% on more than 80% so bugs can be identified in the first phase.

**4.2.7 Internationalization requirements**

* System should be developed in a manner so It can be shift to multi language support in future.
* All text elements should be separated from UI so language switch can be easy.

**4.3 Use Case Description**

There are some use case descriptions given below that tells us how does system actors (customer or admin) Can interact with the system.

**4.3.1 Use Case ID: UC1**

A diagram with a black and white text

AI-generated content may be incorrect.

**Preconditions:**

User Have not registered in the system

**Main flow:**

1. User clicks on the register button.
2. Register form opens
3. User enters name, e-mail, phone number and password.
4. The system validates the input if data is valid then system creates user account.
5. Confirmation message display on the screen.

**Alternate flow:**

If e-mail is already registered to the system, then an error will be showed on the screen “e-mail already registered”.

**Post condition**

A new user successfully registers in the system and can be able to log in.

**4.3.2 Use Case ID: UC2**

A diagram with text on it

AI-generated content may be incorrect.

**Preconditions:**

User should be Registered.

**Main flow:**

1. User opens the login page.
2. Users enter e-mail, username and password.
3. The system validates the credentials, if credentials are right then user’s dashboard opens.

**Alternate flow:**

Username is wrong then an error message shows on the screen “invalid credentials”.

If user inputs wrong details 4 three times, then the account will temporarily lock.

**Post condition**

User successfully log in to the system and ready to use it.

**4.3.3 Use Case ID: UC3**

A diagram with a black text

AI-generated content may be incorrect.

**Preconditions:**

Users must be registered and logged in the system

**Main flow:**

1. Users click the “Available Rooms” section
2. Users enter check-in & check-out date and type of the room.
3. Clicks on “Book Now”.
4. The system confirms the availability of the room.
5. After successful booking, confirmation messages are shown on the screen.

**Alternate Flow:**

If room is already booked than system will show an error message “Selected room is not available”.

**Post conditions**

room booking will be saved in database end users record will be upgraded.

**4.3.4 Use Case ID: UC4**

A diagram with text on it

AI-generated content may be incorrect.

**Preconditions:**

The user must be logged into the system and has booked a room.

**Main Flow:**

1. User clicks on “Cancel Booking” button.
2. User inputs its room ID, check-in and check-out date.
3. Booking cancellation process starts and user get a notification “Just wait for a while”

**Alternate Flow:**

If user Cancels booking after 24 hours then a message shows on the screen “booking cannot be cancelled after 24 hours of check-in”

**Post condition:**

booking cancelled successfully And room is now available for other customers.

**4.3.5 Use Case ID: UC5**

A diagram with a black and white text

AI-generated content may be incorrect.

**Preconditions:**

Users must be registered and logged in the system

**Main Flow:**

1. User opens food menu
2. User selects food items and the quantity
3. Click on “place order”
4. Order status shows on the screen “pending, preparing or delivered”

**Alternate Flow:**

If the item is out of stock, then the user gets notification.

**Postcondition:**

Order places successfully and order saves in the system.

**4.3.6 Use Case ID: UC6**

A diagram with a black line and a white oval with text

AI-generated content may be incorrect.

**Preconditions:**

The user must be logged into the system and has booked a room.

**Main Flow:**

1. User opens “room service” section
2. Selects cleaning, towel or laundry
3. Click on request button
4. System notifies admin about room service request
5. Status of request updates

**Alternate flow**

If service is not available in the system, then “not available” message shows on the screen.

**Post condition**

Room service request records in the system so admin can handle it

**4.3.7 Use Case ID: UC7**

A diagram with a black and white text

AI-generated content may be incorrect.

**Preconditions:**

The user must be logged into the system and has booked a room.

**Main Flow:**

1. User opens “booking” section
2. Clicks on “Check-out button”.
3. Enters room ID, check-in and check-out date.

**Alternate flow:**

If inputs are wrong, then system shows an error message on the screen “Invalid Credentials”.

**Post condition:**

The check-out process successful and the room is available for other customers.

**4.3.8 Use Case ID: UC8**

A white oval with black text

AI-generated content may be incorrect.

**Preconditions:**

Admin must be logged in the system.

**Main Flow:**

Admin can manage rooms, booking and food menu.

**Alternate Flow:**

If required fields are missing, then the system will produce an error.

**Post condition:**

Admin successfully updated changes

**4.4 System Features:**

The main features of the hotel management system are given below.

* + 1. **User Registration**

New users can create their account by entering their name, email, phone, and password.

**Priority:** high

* + 1. **User Login**

After registration, users can login by entering credentials to access other features.

Priority: high

* + 1. **Room Search & View**

Users can view available rooms with type and price.

**Priority:** high

* + 1. **Room Booking**

Users can book rooms by entering check-in & check-out date.

**Priority:** high

* + 1. **Cancel Booking**

User can cancel his existing bookin with in 24 hours of booking.

**Priority:** Medium

* + 1. **Food Menu Viewing**

Users can view food menu in which food items, prices and availability shows on the screen.

**Priority:** high

* + 1. **Food Order**

Users can select food items to order

Priority: high

* + 1. **Request Room Service**

Users can request room service (e.g. cleaning, laundrey)

Priority: Medium

* + 1. **Admin Dashboard**

Admin can login and overview the system.

Priority: high

* + 1. **Manage rooms**  
       Admin can manage rooms (like add new rooms, delete existing rooms)

**Priority:** High

* + 1. **Manage Food Menu (Admin)**  
       Admin can manage food items (like add new food items, delete existing food items)

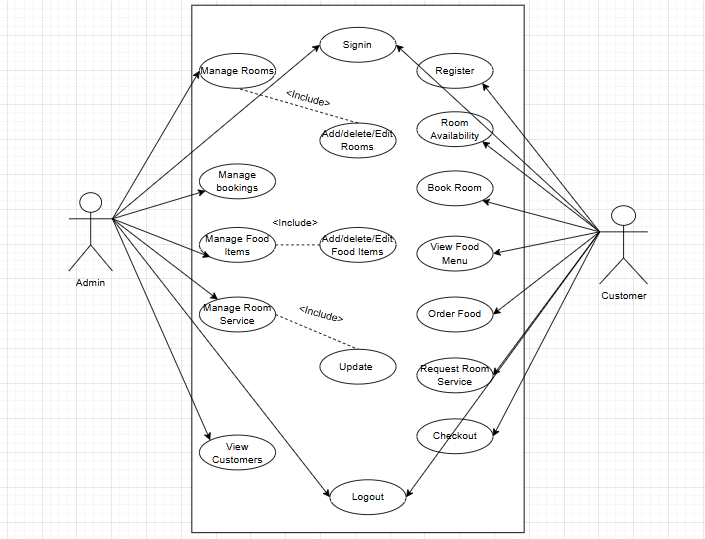
**Priority:** Medium

* + 1. **Generate Bill at Checkout**  
       When the user checks out then system automatically generates the bill.

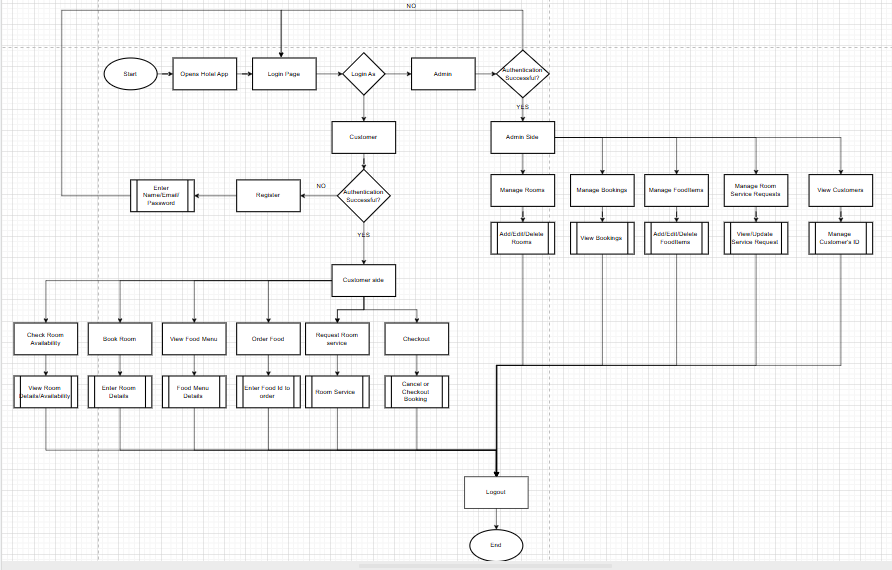
**Priority:** High

* 1. **Diagrams**

**4.5.1 Use Case Diagram:**



**4.5.2 System Flow Diagram**



**4.5.3 Class Diagram**

**A screenshot of a computer

AI-generated content may be incorrect.**

**4.5.4 Sequence Diagram**

**A screenshot of a computer

AI-generated content may be incorrect.**

**4.5.5 Mind MappingA black grid with white text

AI-generated content may be incorrect.**

1. **External Interface Requirements:**

**5.1 User Interfaces (Screens, Menus, Forms):**

Interface of this Hotel management system is based on simple and and user friendly environment. For two roles different interfaces are made: customer and admin. Both interfaces are console based but are easy to understand.

**5.1.1 Customer interface:**

**Login Screen:**

Customer enters email and password to login into the system if credentials are not matched error message is shown.

**Customer Menu:**

After login a menu appears for the user where customer enter option of own desire.

1. Room availability
2. Book room
3. Order food
4. Request room service
5. Checking out

**Booking Form:**

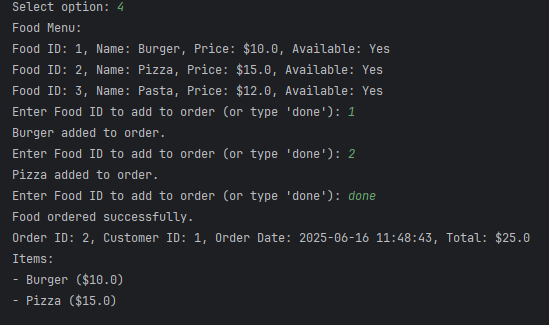
When customer books room a form type menu shows up where customer have to enter room type, check in and check out date.

A screen shot of a computer

AI-generated content may be incorrect.

**Food Menu:**

When customer orders food, system shows available food items and prices, and customer places its order.



**Room Service Form:**

For room service request simple input is taken from the customer like “cleaning required” or “need extra towels”.

A screenshot of a computer program

AI-generated content may be incorrect.

**Confirmation Screens:**

After every action a message is shown on the screen that the work has been done successfully.

**5.11.2 Admin interface:**

**Login Screen:**

Admin also enters email and password to login and if wrong credentials are entered system denies the access.

**Admin Menu:**

After login following options are shown to admin:

1. Add rooms
2. Delete/Update existing rooms
3. Add food items
4. Delete/Update food items
5. View room bookings
6. View room service record

**Login as Admin:**

A screenshot of a computer

AI-generated content may be incorrect.

**Menu:**

A screenshot of a computer

AI-generated content may be incorrect.

**Manage room:**

A screenshot of a computer program

AI-generated content may be incorrect.

**Add Room:**

A screenshot of a computer program

AI-generated content may be incorrect.

**Edit Room:**

A screen shot of a computer code

AI-generated content may be incorrect.

**Delete Room:**

A screenshot of a computer

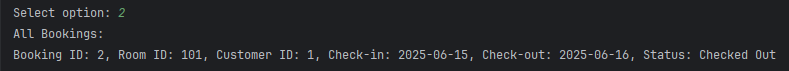
AI-generated content may be incorrect.

**View Rooms:**

A screen shot of a computer screen

AI-generated content may be incorrect.

**Manage Bookings:**

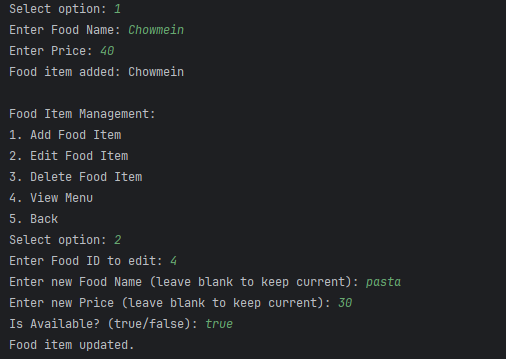


**Manage FoodItems:**

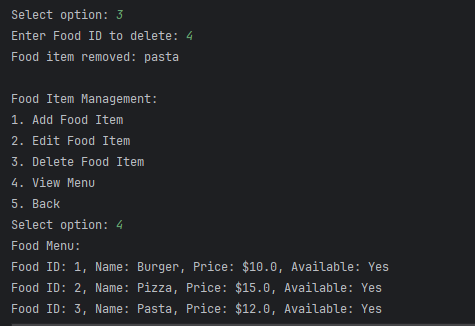
A screenshot of a computer program

AI-generated content may be incorrect.

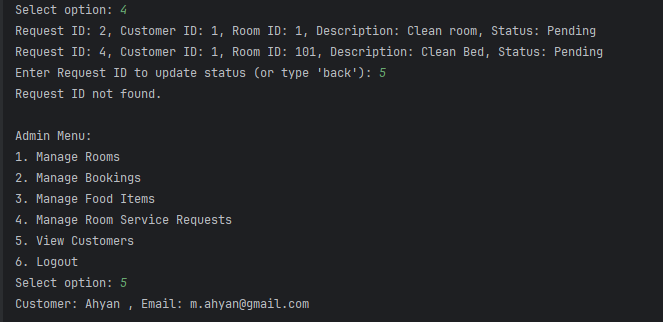
**Add And Edit:**



**Delete and View Food items:**



**Manage room service and View customers:**



**5.2 Hardware Interfaces:**

Specialized hardware is not needed to run this system. Only basic hardware components are used in this system.

**5.2.1 Computer/Laptop:**

System is made on java so it can be run on any Windows, Mac or Linux PC.

**5.2.2 Keyboard and mouse:**

For user input and navigation keyboard and mouse is used.

**5.2.3 Display Screen:**

System output(menus, forms, confirmations) is shown on screen.

This system does not directly interact with physical servers, biometric devices or hardware sensors.

**5.3 Software Interfaces:**

Hotel management system interact with some software components.This interaction is done on basic level.

* + 1. **Java runtime environment (JRE):**

To run the system JRE must be install in pc or laptop.

* + 1. **Java development:**

During development and testing JDK is used.

* + 1. **Operating system:**

The system can be run on any modern OS like Windows, Linux, MacOS.

* + 1. **Optional: File I/O Libraries:**

**I**f you want to store data in future java file handling can be used to store data in local files

The present system does not use any external software or third party APIs, but in future database can .be connected to it like (MySQL, SQLite).

**5.4 Communication Interfaces:**

* Current version of the system is standalone and is not linked with any network or web-based communication features, may be added in future.
* **Email Notification (Future Plan):When customer book any room or checksout system can send email automatically. (By using SMTP protocol)**
* **Web-Based Version (Future Plan):If this project is made web based then HTTP/HTTPS protocols will be used.**
* **Multi-user Networking:TCP/IP protocols can be used to connect multiple users in a networked environment.**
* **Currently communication is internal, means its between user and system through input output commands. External API or web services are not used in this version.**

**6. Other non-functional requirements**

**6.1 Performance Requirements:**

The performance of the hotel management system is very important because it directly affects user experience. some of the following important performance requirements are given below:

* The system should be able to response to the user’s action (such as login, room booking, food order) within 2 seconds to make the user experience better.
* The system should be able to work in a multi-user environment smoothly. If there are 50 users using the system at a time, then it should not work slowly. The system should be able to support a minimum of 100 concurrent users during load testing.
* The communication between System’s back end and front-end should be fast and efficient. Whenever a user uses search or filter then system must show the results of real-time.
* There should not be any delays during checking room availability and the booking process, especially in pick season when there are so many bookings.
* The performance of the system should be acceptable on even low bandwidth internet. If a user is accessing the system from any remote area, then at least basic functionality should work.
* Periodically performance should be tested by using automated tools like JMeter, so that is should be ensured that the system is ready to handle future traffic.
* Back-end processing (such as invoice generation, payment confirmation) should be in background threads so that UI could not be clow.
* A performance monitoring dashboard should be applied in the system where admin can view CPU usage, memory load and real-time status of the system.
* Important tasks like room booking and food ordering should be prioritized so it can always work fast.
* The system should be supported by an asynchronous system so multiple tasks can work parallel, especially when there is heavy user traffic.

**6.2 Security Requirements:5.**

In the hotel management system, security is very important because it keeps users’ personal data, payment information and confidential records. There are some security requirements given below:

* The user authentication system should be strong which must have secure login page, and sensitive data should be stored in encrypted form. Usernames and passwords must not be stored in plain text.
* Password hashing algorithms should be used so that if database will leak then password will not be exposed.
* There should be rate limiting and account lockout policies at the time of login so that brute force attack should be stopped.
* There should be an option of Two-Factor Authentication for advanced security, in which the user will get a code via mobile or email.
* The rights of admin and general users must be defined clearly. Admin panels will be only accessible by authorized people, and every user can access their data only.
* Input validation and sanitization should be used so that SQL injections, cross-site scripting and other injections should be secured by any attack.
* The communication of the system should be encrypted through HTTPS (SSL certificate) so that data should be secured in transit.
* Security audit logs should be maintained so that any suspicious activity should be tracked.
* The system should go through penetrating testing regularly so that new vulnerabilities should be detected and fixed.
* Whenever the system is logged into by a new device, then user should get notification via email or SMS.
* A secure password policy should be implemented in which minimum length, special characters will be used.
* Records of failed login attempts will be maintained (like IP address, username) and it will be shown on the admin dashboard.
* The system should have maintained every configuration change (especially user roles) so unauthorized access can be paused.

**6.3 Quality Attributes:**

The system’s quality is not only its functionality, but it also matters that the system is reliable, usable, maintainable and future proof. There are some important quality attributes given below:

**6.3.1 Maintainability**

* System architecture should be modular so that it will be easy to update or replace any module in future.
* Code should be written with proper comments and documentation to make it easy to understand for new developers.
* Version control should be used for the system to keep the record of changes.
* An issue tracking system should be implemented to fix bugs.
* Unit and integration testing should be implemented on every module so that future updates can be done easily.
* Coding standards and naming conventions should be implemented, and every developer must follow it.

**6.3.2 Usability**

* The user interface should be intuitive and simple. There should be no need for any technical training for users to use the system.
* The system must contain tooltips, help sections, and feedback messages to guide users.
* Fonts, colors, and layout should be used according to accessibility standards so the user can use the system easily.
* Onboard screens or tooltips should be available for the users who are using the system for the first time so that they can understand the usage of the system.
* Real-time validation feedback should be given in form fields so that users can get guidance on time.

**6.3.3 Portability**

* The system must have a responsive design so it will be easy to use it on a mobile phone, tablet and even on desktops.
* Web app should be compatible with different browsers (such as Chrome, Firefox, Safari, Edge)
* The system will migrate easily from local to cloud environments like AWS or Azure.
* Application should be containerized by tools like Docker so that it can be deployed in any environment.
* Sensitive environments like API keys and database credentials should be separated from code in environment variables.

**6.3.4 Reliability**

* The system must be available all the time, with expected uptime 99.9%.
* A backup mechanism should be implemented so data will not be lost.
* Fault tolerance should be implemented, if any service fails so others act automatically.
* The system must have automated health check endpoints so monitoring tools can verify whether services are active or not.
* System should automatically retry failed operations like payment failures or API errors but with safe retry policy.

**6.3.5 Scalability**

* System should be designed in a manner that in future if number of user or branches will increase then it will be scaled easily.
* Cloud-based deployment should be considered so the system will allocate resources automatically based on traffic.
* The system’s architecture should be horizontally scalable so new servers can be added when in need.
* Catching tools (like Redis) should be used to maintain load on databases during high traffic.

**6.3.6 Interoperability**

* System should be able to integrate with third party APIs like payment gateways, email / SMS services or external hotel platforms.
* Data exchange will happen in standardized format with every external system.
* OAuth 2.0 or API tokens should be used to secure communication with external systems.
* The system should have a feature of importing and exporting data like CSV or Excel files so that system will remain compatible with old and external systems.

**7. Appendices**

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **Field Name** | | |  | | --- | | **Description** | |
| |  |  | | --- | --- | |  |  | | |  | | --- | | Guest Name | |  | | |  | | --- | | Full name of customer | |
| |  | | --- | | Check-in Date | | |  | | --- | | Date of arrival | |
| |  | | --- | | Check-out Date | | |  | | --- | | Date of leaving | |
| |  | | --- | | Room Type | | |  | | --- | | Single, Double, Suite etc. | |
| |  | | --- | | Contact Number | | Mobile Number |

**Appendix B: Room Status Table**

|  |  |  |
| --- | --- | --- |
| **Room Number** | **Type** | **Status** |
| **804** | **Single** | **Booked** |
| **999** | **Double** | **Available** |
| **333** | **Suit** | **Cleaning** |

**Appendix C: Sample Bill Format**

* Customer Name: Muhammad Ashar
* Room Type: Double
* Nights Stayed: 2
* Room Rate: Rs. 5000/night
* Total: Rs. 10000

**Appendix D: Survey Questions for Staff**

* Is the new system easy to use?
* Are the bookings and payments faster now?
* Do you face any issues while using the system?
* Any suggestions for improvement?