HOTEL ROOM RESERVATION SYSTEM

Software Requirements Specification

Version 1.1

10th June,2025.

GROUP MEMBERS

NAMES	ROLL NO
Hadia Naveed	F23-0093
Syed Sibtul Hassan	F23-0080
Amna Gul	F23-0078

Prepared for:
Software Engineering
Instructor: Sir Mehmood
BSCS 4TH “B”

Revision History

Date	Description	Author	Comments
3rd June,25	Created initial draft	Hadia Naveed	Added Introduction, Purpose, and Scope
3rd June,25	Added functional requirements section	Syed Sibtul Hassan	Covered Use Cases and User Roles
3rd June,25	Completed non-functional requirements	Amna Gul	Included performance, security, and availability details

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Hadia Naveed	Documentation Specialist	4th June 2025
	Syed Sibtul Hassan	Lead Software Engineer	4th June 2025
	Amna Gul	System Analyst	4th June 2025

Table of Contents

CHAPTER 4:

CHAPTER 5:

CHAPTER#1

1. Introduction

The Hotel Room Reservation System is a tool for booking the rooms of Hotel through online by the Customer. It provides the proper management tools and easy access to the customer information.

1.1 Purpose

The main objective of this Software Requirements Specification (SRS) document for the Hotel Room Reservation System (HRRS) is to serve as a foundational reference for the development of the project. It captures and analyzes the client's expectations and requirements to produce clear, specific, and unambiguous functional and non-functional requirements. These requirements are intended to guide the development team in building a system that meets the needs of the end users effectively. Additionally, this SRS document can serve as a reference for future development, providing insight into how the project was initiated. It acts as a blueprint for new developers and maintenance teams, aiding in the ongoing support and future modifications of the system.

1.2 Scope

The Hotel Management System (HMS) is designed to enable online room reservations and automate core hotel operations. The system will support three types of users: Customer, Receptionist, and Hotel Manager. It will include key components such as a Booking Management System, Database Server, and Report Generator.

Customers can check room availability, select rooms, and make payments online. Receptionists will be able to update and modify booking details, while Managers can view financial reports and update room information such as pricing and categories.

The main objective of this automated system is to streamline daily hotel operations and enhance customer convenience. It replaces traditional manual record-keeping, improving efficiency, accuracy, and ease of access. Features such as secure transactions, quick data retrieval, user-friendly interface, error recovery, and fault tolerance will be implemented to ensure high performance and user satisfaction.

1.3 References

This section lists all the documents and sources referenced throughout the Software Requirements Specification (SRS) for the **Hotel Room Reservation System**. Some of these are internal documents, accessible via the appendices.

1. **IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications**
   Report No.: IEEE Std 830-1998
   Date: 1998
   Publishing Organization: IEEE
   Source/Location: IEEE Xplore
2. **Project Proposal: Hotel Room Reservation System**
   Version: 1.0
   Date: May 2025
   Publishing Organization: BSCS – Group B (Amna, Hadia, Hassan)
3. **Hotels and Guest Houses Survey Report**
   Date: 27 May 2025
   Publishing Organization: Project Team
   Source/Location: *Appendix 1 – Hotels and Guest Houses Surveyed*
4. **UI Design Guidelines for Responsive Systems**
   Report No.: UI-GUIDE-21
   Date: 2021
   Publishing Organization: ISO/IEC
   Source/Location: www.iso.org

1.4 Overview

The remaining parts of this Software Requirements Specification (SRS) document provide a complete overview of the system. It includes a general description of the Hotel Management System, outlining its background, core functions, user characteristics, assumptions, and constraints.

The document also specifies all system requirements, including functional and non-functional requirements, external interfaces, performance expectations, capacity, availability, safety considerations, and a requirement traceability matrix.

The SRS is organized to offer a clear and logical flow, making it easy for developers, stakeholders, and maintenance teams to understand the system's design, implementation goals, and future development needs

CHAPTER#2

2. General Description

The Hotel Management System is intended to automate and streamline the booking and management processes of a hotel, improving efficiency for both staff and customers. The system must accommodate multiple user roles, including customers making reservations online, receptionists managing bookings, and managers overseeing operations.

Factors such as user accessibility, security of financial transactions, system reliability, and data integrity significantly affect the design and implementation of the system. Additionally, the system will operate in an environment with variable network connectivity and must integrate with existing hotel infrastructure, such as payment gateways and database servers

2.1 Product Perspective

The Hotel Management System (HMS) is designed to function as an integrated software solution within the existing hotel environment. It is intended to replace manual and paper-based processes with automated workflows that improve efficiency and accuracy.

The system must accommodate interactions between various user roles—customers, receptionists, and managers—each with distinct responsibilities and access levels. It will also interface with external systems such as payment gateways and database servers, which impacts system design considerations including security, reliability, and data integrity.

Operational factors such as network connectivity, system scalability, and ease of maintenance influence the requirements for performance and availability. These general factors provide context to better understand the detailed functional and non-functional requirements that follow in this document.

2.2 Product Functions

Our Product General functions are:

1. Customer Registration
2. Check for Availability of Rooms
3. Display the Rate
4. Confirmation Of Booking
5. Email Notification
6. Payment
7. Set Room Details
8. Manage Booking Details
9. Generate Report
10. Customer Service.

2.3 User Characteristics

There are 3 user Levels in our Hotel Management System:

A. Hotel Manager

B. Receptionist

C. Customers

A. **Hotel Manager**
The Manager has full access to the system and is responsible for overseeing hotel resources and staff. The Manager can view and analyze various reports, including financial data, customer details, booking records, and room information, to make informed decisions. It is expected that the Manager has prior experience in hotel management and possesses basic knowledge of databases and application servers to effectively use the system.

B. **Receptionist**
The Receptionist's primary role is to provide quality customer service with limited system access compared to the Manager. Responsibilities include managing booking details, checking room availability, adding new customers, confirming bookings, and updating reservations. The ideal Receptionist should have good communication skills, proficiency in English, and basic IT knowledge to operate the system efficiently.

C. **Customer**
Customers are the primary users who interact with the system to view available rooms and prices, make bookings, and cancel reservations if needed. They can also submit inquiries via the customer service portal. Customers should have basic computer literacy and be able to navigate the web-based user interface independently.

2.4 General Constraints

1. Memory Constraint: The system will have access to only 10 GB of storage space on the data server, limiting the amount of data that can be stored and processed.Language
2. Requirement: The software interface and documentation will be developed exclusively in English.
3. Budget Constraint: Due to limited funding, the HMS will be designed to provide only basic functionalities. Consequently, the user interface will be kept simple and minimalistic.
4. Implementation Constraint: The application must be developed using Java as the primary programming language to ensure platform independence and compatibility.

5. Reliability Requirement: The system must frequently synchronize data with a backup server to prevent data loss in the event of system failure. This ensures that data can be recovered reliably.

2.5 Assumptions and Dependencies

It is assumed that the Hotel Management System will be developed to operate on the Windows operating system, utilizing Apache Server and MongoDB as the database. The system is expected to function correctly within this environment. However, if any challenges arise, the SRS will be updated and adjusted accordingly to accommodate necessary changes.

CHAPTER#3

3. Specific Requirements

3.1 External Interface Requirements

External Interface Requirements 3.1.

1. 1 User Interfaces The user interface for system shall be compatible with any type of web browser such as Mozilla Firefox, Google Chrome, and Internet Explorer. 3.1
2. 2 Software Interfaces Web Server

a. Apache Tomcat Server, OS (Windows)Database Server

b. Mongo DB, OS (Windows)Development End

c. J2EE, Java, JSP, Servlet, HTML, XML, JavaScript, OS (Window

3.1.1 User Interfaces

The user interface for system shall be compatible with any type of web browser such as Mozilla Firefox, Google Chrome, and Internet Explorer.

3.1.2 Hardware Interfaces

The Hotel Management System (HMS) is designed to operate on standard hardware components for both client devices and server infrastructure. The minimum hardware requirements to ensure efficient deployment and operation are as follows:

Client Devices:

1. **Supported Devices**: Android smartphones (version 9.0 or later), iPhones (iOS 13 or later)
2. **RAM:** Minimum 2 GB

3. **Browser Compatibility:** Latest versions of Chrome for Android, Safari for iOS, an Firefox Mobile
4. **Display:** Minimum resolution of 720p
5. **Input:** Touchscreen support required
6. **Network:** 4G/LTE or stable Wi-Fi connection

Server-Side Hardware (Hosting Application and Database):

1. **Processor:** Intel Xeon or equivalent multi-core processor
2. **RAM:** Minimum 16 GB (Recommended 32 GB for optimal performance)
3. **Storage:** SSD with at least 500 GB capacity, with regular backup support
4. **Network:** High-speed broadband with a static IP address, minimum 100 Mbps bandwidth
5. **Power Backup:** Uninterruptible Power Supply (UPS) or generator to prevent data loss during power outages

Additional Requirements:

i. Hardware must support Windows Operating System for both client and server environments

ii. Server hardware must be compatible with Apache Tomcat application server and MongoDB database

3.1.3 Software Interfaces

Web Server:

1. Server Software: Apache Tomcat (version X.X or later)
2. Operating System: Microsoft Windows (Windows Server 2016 or later)

Database Server:

1. Database System: MongoDB (version X.X or later)
2. Operating System: Microsoft Windows
3. Development Environment:
4. Technologies Used:
   a. Backend: J2EE, Java, JSP (JavaServer Pages), Servlets
   b. Frontend: HTML, XML, JavaScript
5. Operating System for Development: Windows 10 or later

3.1.4 Communications Interfaces

1. Serial Port

2. USB (Universal Serial Bus)
3. Ethernet ports
4. Wi-Fi adapters
5. TCP/IP stack
6. HTTP, FTP protocols
7. Operating System APIs for network communication

3.2 Functional Requirements

3.2.1 Registration

1. The Customer should be able to register with their details.
2. The system should record the following customer details into a member database. Name Email Password Address DOB.
3. The system shall send verification messages to email

3.2.2 Logging In

The system should verify the customer email & password against the member database when logging in FR5. After login, members should be directed to home screen

3.2.3 Reservation

1. The system should enable customers to check for the availability of rooms.
2. The system should display rates for all rooms.
3. The system should allow customers to confirm or cancel the booking. The system should record booking details into a database.

3.2.4 Receptionist Access

1. The system should allow Receptionist to update, add or delete booking information
2. The system should provide customer desk portal access to receptionists for providing responses to customer inquiries.

3.2.5 Manager Access

1. The system should generate financial and customer reports for managers.
2. The system should enable manager full modification access to customer, booking and room information

3.2.6 Payment Management System

The system should allow customers to pay bill via online using credit or debit card

3.2 Use Cases

the interactions between users and the system for performing specific functions.

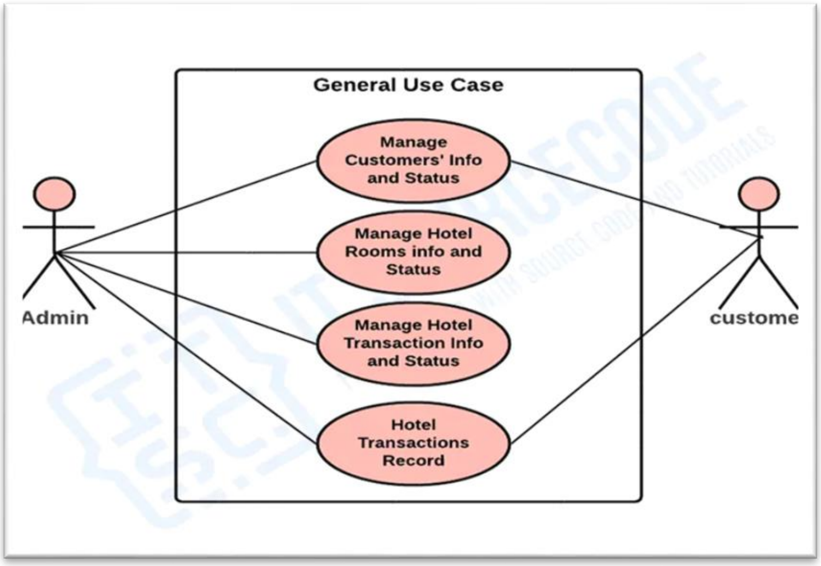
General Use Case
Manage Customers' Info and Status
Manage Hotel Rooms info and Status
Manage Hotel Transaction Info and Status
Hotel Transactions Record
Admin
custome

Fig 1.0

3.3 Classes / Objects

classes and objects are concepts from object-oriented design and programming.
In a Hotel Management System, classes might include:

1. Guest
2. Room
3. Reservation
4. Payment

3.4 Objects:

An object is an instance of a class, with its own set of attributes (data) and methods (functions). In the SRS document, objects might be used to illustrate how the system will work.

Using the classes mentioned earlier, objects might include:

Guest object: John Doe, with attributes like name, email, and phone number.

3.5- Reservation object:

Reservation #123, with attributes like guest, room, and dates.

3.4.1 Class / Object #1

Class: Room
object: Room 101 attributes like room number, type, and rate

Attributes: room number, type, and rate

3.4.2 Class / Object #2

Class: Reservation
object: Reservation #123
attributes: guest, room, and dates.
…

3.5 Non-Functional Requirements

3.5.1 Performance

1. Data in the database should be updated within 2 seconds.
This means whenever a user adds or updates something—like booking a room, changing personal info, or canceling a reservation—the system should save that change to the database in less than 2 seconds.
This ensures that all changes are quickly stored and the data stays accurate and up-to-date without delay.

2. Query results must be returned within 5 seconds.
When the system is asked a question—like "show me all available rooms" or "list today's bookings"—it should fetch and show the results within 5 seconds.
This keeps the system responsive and avoids frustrating users with long waiting times.

3. The user interface (UI) should load in 2 seconds or less.
Whenever someone opens the website or app, the main screen or any page (like the homepage or dashboard) should load in 2 seconds or faster.
A fast-loading UI makes the system feel smooth and professional, and users are more likely to stay.

4. Login validation must be completed within 3 seconds.
After a user types their username and password, the system should check the credentials and either log them in or show an error message within 3 seconds.
This makes the login process fast and efficient, giving users immediate access without unnecessary delays.

5. Customer inquiries must be responded to within 5 minutes.

If a customer sends a message (like a help request or question), they should get a reply—either automatic or from support staff—within 5 minutes.
This helps build trust and shows that the service cares about customers' problems and is quick to respond.

6. The system must handle at least 20 transactions per minute during peak hours.

During busy times (like holidays or weekends), the system should be able to process at least 20 bookings, updates, or payments per minute without slowing down or crashing.
This ensures that the system stays reliable and performs well even when many users are active at the same time.

3.5.2 Reliability

1. The system should recover from failure and return to normal operation within 1 hour.
2. The system should be tested to ensure it functions correctly according to business rules and user requirements.

3.5.3 Availability

1. The system must be operational 24/7.
2. The database should be backed up every hour to ensure data can be recovered quickly after failure.

3.5.4 Security

1. All communications between the server and clients must be encrypted.
2. Sensitive data such as personal and financial information must be securely stored and protected.
3. Payment processing must be conducted over a secure HTTPS connection.

3.5.5 Maintainability

1. The system should support easy updates and bug fixes with minimal time.
2. An admin panel or tools should be provided to support system diagnostics and maintenance activities.

3.5.6 Portability

1. The system must run on any Microsoft Windows environment.
2. The web application should be compatible with all major browsers (Chrome, Firefox, Edge, etc.).

3.6 Inverse Requirements

1. The system must not allow access to user or payment data without proper authentication.
2. The system must not take longer than 10 seconds to load under normal usage.
3. The system must not store sensitive information like passwords in plain text.
4. The number of registered users must not exceed 10,000, to ensure system performance.

3.7 Design Constraints

1. The system must follow company encryption policies, such as implementing AES-256 encryption for all sensitive data including user credentials and personal information.
2. All payment features must comply with PCI DSS (Payment Card Industry Data Security Standards) to ensure secure handling of credit/debit card transactions. Direct storage of cardholder data is prohibited.
3. The system must be fully web-based, requiring no client-side installation. It should be accessible via standard web browsers like Google Chrome, Mozilla Firefox, and Microsoft Edge.
4. The system must be optimized to operate efficiently in standard server environments, including but not limited to:
    - **a.** 8 GB RAM
    - **b.** 4-core CPU
    - **c.** 100–200 GB SSD storage
    - **d.** Compatible with both Windows Server and Linux-based hosting platforms.

3.8 Logical Database Requirements

1. A relational database will be used to store all users, rooms, and booking information.

2. Data must follow proper structure and integrity, using foreign keys and normalization.
3. Booking and customer data must be retained for at least 2 years.
4. The system must support hourly backups and allow recovery within the last 24 hours.

3.9 Other Requirements

1. **Usability**: The system must provide user manuals or help documentation for all user types.
2. **Correctness:** The system must implement all hotel booking processes accurately (e.g., check-in/out, availability, payments).
3. **Flexibility**: The design should allow new features to be added easily.
4. **Efficiency:** The system must use resources effectively, especially during high traffic.
5. **Integrity:** The system must ensure no duplicate bookings or data loss.
6. **Testability**: The system must include automated tests and allow for quality assurance before release.

CHAPTER#4

4. Analysis Models

4.1 Sequence Diagrams

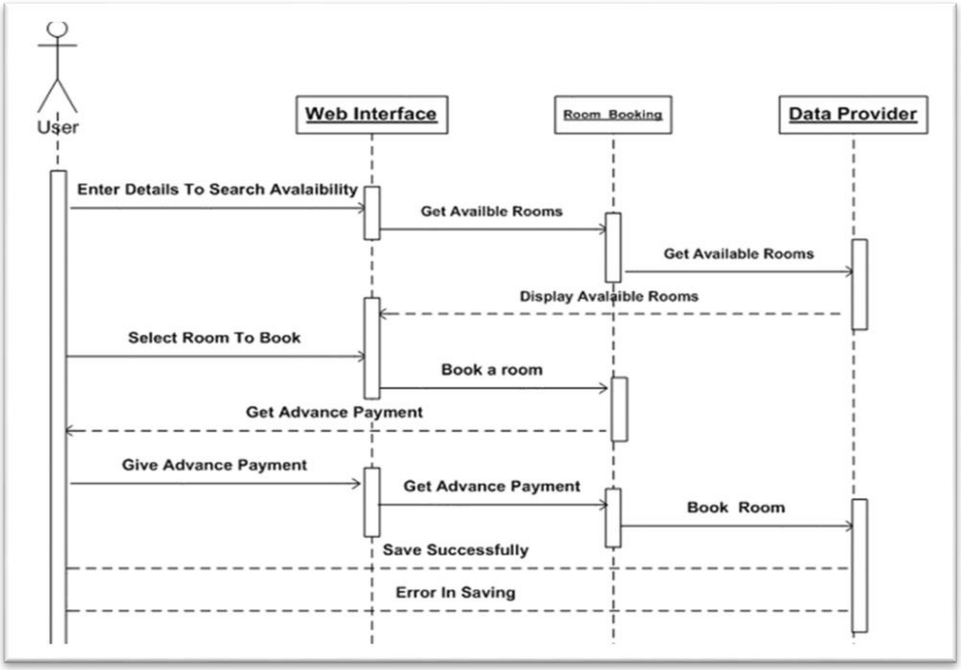

Fig 2.0

4.2 State-Transition Diagrams (STD):

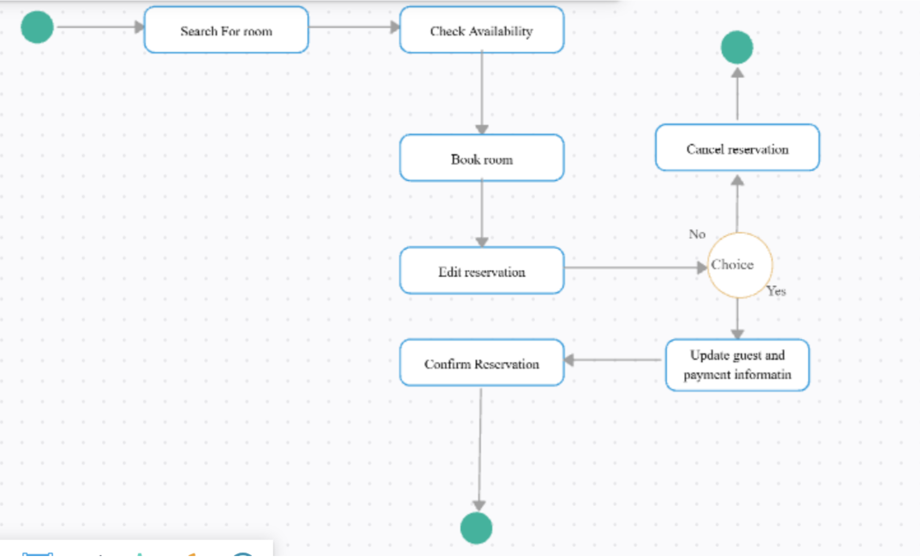

Fig 3.0

4.3 Data Flow Diagrams (DFD)

Level 0

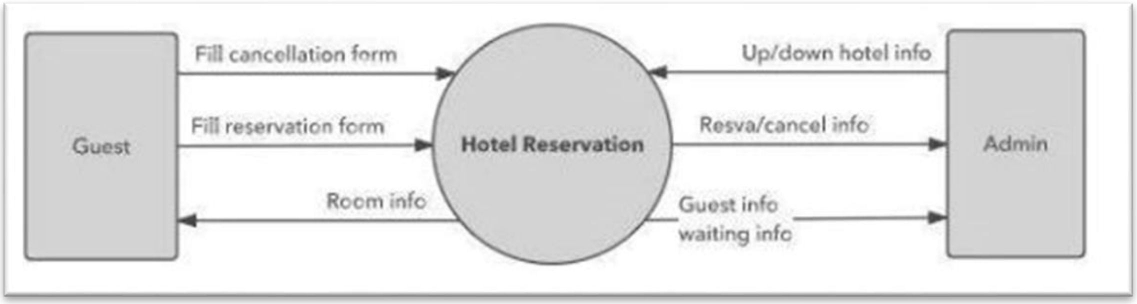

FIG 4.1

Level 1

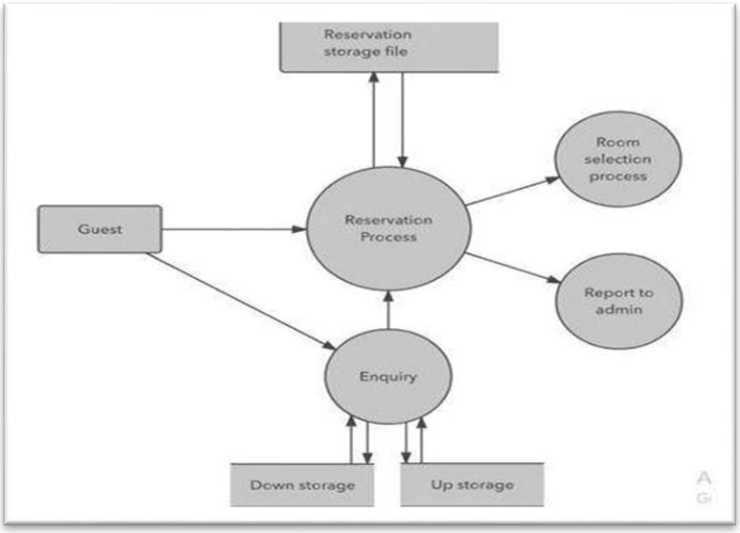

Fig 4.2

Level 2

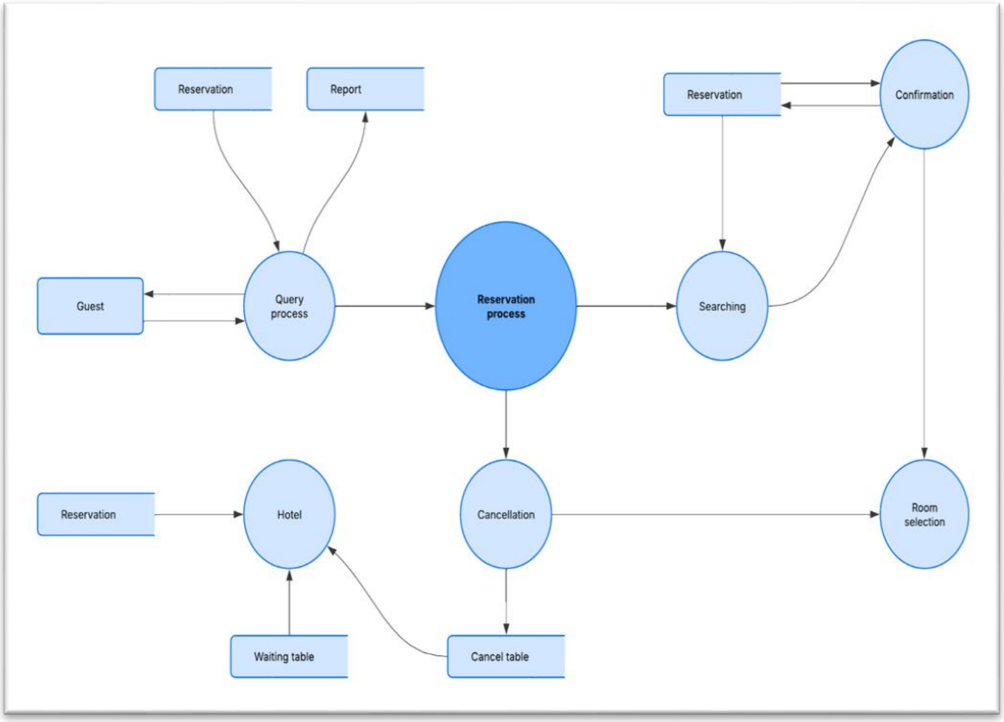

FIG 4.3

CHAPTER#5

5 Change Management Process

Changes to this Software Requirements Specification (SRS) document or project scope will be managed carefully to ensure the system stays aligned with client needs and project goals. The process below will be followed whenever updates are needed:

5.1 Change Submission

1. Change requests can be submitted by:
   - **a.** Project stakeholders (including hotel management and clients)
   - b. Development team members
   - **c.** Quality assurance/testers
   - d. End users (through official support channels)

2. Requests must be submitted in writing via email or through the project management tool (e.g., Jira, Trello). Each request should include:

   a. Detailed description of the proposed change

   b. Reason and justification for the change

   c. Potential impact on project timeline, budget, and functionality (if known)

   d. Priority level (urgent, high, medium, low)

5.2 Change Implementation

1. All team members will be informed about the approved changes.
2. Changes will be scheduled and implemented according to project priorities and timelines

3. Approved changes will be documented in an updated version of this SRS.

Change Review and Approval

1. A Change Review Board (CRB), consisting of the Lead Software Engineer, Project Manager, and Client Representative, will review all change requests.

2. The CRB evaluates each change for feasibility, impact, cost, and priority.

3. The CRB will decide to:

   a. Approve the change for implementation

   b. Reject the change

   c. Request further information or defer the change

5.4 Change Tracking and Communication

1. All change requests and their status (submitted, approved, rejected, implemented) will be recorded in a Change Log.

2. The Change Log will be regularly reviewed and made available to all stakeholders.

3. Communication about approved changes will be sent promptly to ensure everyone is aware and aligned.

Appendix 1 — Hotels and Guest Houses Surveyed

This appendix summarizes the visits conducted at three hotels: **Paradise Guest House**, **Afaq Hotel,** and **Horizon Guest House,** all located in **Haripur** , on 27-may-2025. The purpose of these visits was to observe current hotel operations, understand customer booking processes, and gather detailed requirements for the Hotel Management System (HMS).
During the visits, the following activities were performed:

1. Observed the front desk and booking workflows at each hotel.
2. Interviewed hotel staff, including receptionists and managers, to identify pain points and essential features.
3. Collected information on room categories, booking challenges, and customer service procedures.
4. Noted existing tools and technologies used by each hotel for managing bookings and customer data.