

assignment 3

20f-0185 , 20f-1083 , 20f-0115



Contents

[1. INTRODUCTION 1](#_Toc148756464)

[**1.1** **Purpose** 1](#_Toc148756465)

[1.2 This SRS outlines the requirements for a Hotel Reservation System, aimed at streamlining booking processes for both hotels and guests. . 1](#_Toc148756466)

[1.3 Document Conventions 1](#_Toc148756467)

[1.4 Intended Audience and Reading Suggestions 1](#_Toc148756468)

[**1.5** **Product Scope** 2](#_Toc148756469)

[**1.6** **References** 2](#_Toc148756470)

[2. OVERALL DESCRIPTION 3](#_Toc148756471)

[2.1 Product Perspective 3](#_Toc148756472)

[**2.2** **Product Functions** 3](#_Toc148756473)

[2.3 User Classes and Characteristics 4](#_Toc148756474)

[2.4 Operating Environment 6](#_Toc148756475)

[2.5 Design and Implementation Constraints 8](#_Toc148756476)

[2.6 User Documentation 9](#_Toc148756477)

[2.7 Assumptions and Dependencies 9](#_Toc148756478)

[**2.8** **Software Interfaces** 14](#_Toc148756479)

[**2.9** **Communications Interfaces** 15](#_Toc148756480)

[3. System Features 17](#_Toc148756481)

[4. Other Nonfunctional Requirements 19](#_Toc148756482)

[4.1 Performance Requirements 19](#_Toc148756483)

[4.2 Safety Requirements 21](#_Toc148756484)

[***4.3*** Security Requirements 23](#_Toc148756485)

[4.4 Software Quality Attributes 24](#_Toc148756486)

[4.5 Business Rules 26](#_Toc148756487)

[5. Other Requirements 27](#_Toc148756488)

[Use Case 1 30](#_Toc148756489)

[Use Case 2 31](#_Toc148756490)

[Use Case 3 32](#_Toc148756491)

[Use Case 4 33](#_Toc148756492)

[Use Case 5 34](#_Toc148756493)

[Use Case 6 35](#_Toc148756494)

[Use Case 7 36](#_Toc148756495)

[Use Case 8 37](#_Toc148756496)

[**HIGH LEVEL USE CASES** 38](#_Toc148756497)

**Justification for Client-Server Architecture**…………………………………56

**HOTEL RESERVATION SYSTEM**

# INTRODUCTION

## **Purpose**

## This SRS outlines the requirements for a Hotel Reservation System, aimed at streamlining booking processes for both hotels and guests. It provides a framework for a user-friendly, efficient, and secure platform. The system covers reservation management, user authentication, room availability tracking, payment processing, and reporting functionalities. Benefits include reduced manual workload, increased booking accuracy, enhanced customer satisfaction, and improved revenue management. This SRS serves as a blueprint for creating a sophisticated and user-centric system, benefiting hotels and customers alike.

## Document Conventions

Heading Style: Times New Roman

Heading 1 Font Size: 16

Heading 2 Font Size: 14

Text Font Size: 11

## Intended Audience and Reading Suggestions

* Developers
* Project Managers
* Marketing Staff
* Users
* Testers
* Documentation Writers

Reading suggestions

1. Introduction
2. Overall Description
3. Specific Requirements
4. External Interface Requirements
5. Other Nonfunctional Requirements

## **Product Scope**

The Hotel Reservation System represents a sophisticated software solution designed to revolutionize the hotel booking experience. Its core purpose is to provide a seamless platform for guests and hotel staff, streamlining reservation management, availability tracking, payment processing, and reporting functions. This system aims to simplify and automate the booking process, significantly reducing manual effort while ensuring accuracy. Guests can expect a user-friendly and hassle-free booking experience, complete with real-time updates on room availability and secure payment processing. For hotel staff, the system enhances resource allocation, improves room occupancy rates, and boosts overall operational efficiency. Data security is paramount, with robust measures in place to protect customer information and payment details. In line with broader corporate goals, the Hotel Reservation System is positioned to elevate customer satisfaction and drive revenue growth within the dynamic hospitality industry. This software plays a pivotal role in the overarching business strategy, leveraging technology to deliver superior services and maintain a competitive edge in the market.

## **References**

* Kehinde Wiilams.Retrieved October 20, 2023 from https://core.ac.uk/download/pdf/23477452.pdf
* Anon. Hotel Reservation System Analysis and design. Retrieved October 20, 2023 from https://www.scribd.com/document/9770142/Hotel-Reservation-System-Analysis-and-design
* Syed Hasan.Retrieved October 20, 2023 from https://www.researchgate.net/publication/25097517\_DOCUMENTATION\_OF\_ONLINE\_BOOKING\_SYSTEM

# OVERALL DESCRIPTION

## Product Perspective

The Hotel Reservation System is a self-contained product developed to function independently as a robust platform for managing hotel bookings. It is designed to streamline the reservation process, facilitate efficient communication between guests and hotel staff, and ensure secure and accurate transaction processing. While it may interface with other systems, such as payment gateways and room availability databases, it is essentially a standalone solution, not reliant on any specific external systems. This system does not replace existing systems within the hotel; rather, it complements them by introducing a more efficient and user-friendly method for managing reservations. It is not part of an existing product family, but rather an innovative addition to the hotel's operational framework. While the Hotel Reservation System operates independently, it may have interfaces with external systems such as payment processors, customer databases, and room inventory management tools. These interfaces ensure seamless communication and data exchange between the reservation system and these external entities.

A visual representation of the system's major components, their interconnections, and interfaces with external systems will be provided in the accompanying system architecture diagram for a more comprehensive understanding of the product's perspective within the larger operational context.

## **Product Functions**

User Authentication and Authorization: Authenticate users, including guests, administrators, and staff. Assign appropriate permissions based on roles.

**Room Availability Management:** Display real-time availability of rooms based on specified dates and preferences. Allow users to select and reserve available rooms.

**Reservation Modification and Cancellation**: Enable users to modify or cancel existing reservations within specified timeframes.

**Payment Processing:**Facilitate secure and efficient payment processing for reservations. Support various payment methods, including credit cards and online payment platforms.

**Reporting and Analytics:** Generate reports on reservations, occupancy rates, and revenue for analysis. Provide insights to optimize pricing and room allocation strategies.

**User Support and Communication**: Offer support channels for users to seek assistance or make inquiries. Facilitate communication between guests and hotel staff for special requests or clarifications.

**System Administration and Configuration**:Allow administrators to manage user accounts, room inventory, and pricing configurations. Enable customization of system settings to align with specific hotel requirements.

**Feedback and Reviews:**Provide a platform for guests to leave feedback and reviews about their stay. Allow hotels to respond to guest feedback and address concerns.

**Integration with External Systems:** Interface with external systems like payment gateways and customer databases for seamless operations.

## User Classes and Characteristics

**Guests**

* Characteristics: Novice to moderate familiarity with online booking systems.

Prior experience with basic web navigation.

Limited exposure to system administration tasks.

* Technical Skills: Basic proficiency in using web browsers.

Familiarity with form-based data entry.

* Access Level: Limited access to booking and managing reservations.

No access to administrative or configuration settings.

**Administrators**

* Characteristics: Advanced knowledge of system administration and configuration.

Proficient in managing user accounts and system settings.

Comfortable with technical aspects of the reservation system.

* Technical Skills: Expertise in system administration and configuration.

Familiarity with database management.

* Access Level:

Full access to system configuration settings.

Privileges to manage user accounts and define system parameters.

**Hotel Staff**

* Characteristics: Moderate proficiency in using the reservation system.

Experience in hotel operations and customer service.

Ability to manage room availability and reservations.

* Technical Skills: Proficiency in using the reservation system's operational features.

Familiarity with customer service protocols.

* Access Level: Access to operational functions, including managing reservations and room availability. Limited access to administrative or configuration settings.

**Customer Support Representatives**

* Characteristics: Extensive experience in customer support and issue resolution.

In-depth knowledge of the reservation system's features and functionalities.

Strong communication and problem-solving skills.

* Technical Skills: Advanced proficiency in using customer support tools and the reservation system. Expertise in handling customer inquiries and resolving issues.
* Access Level: Access to support functions, including assisting guests with reservations and issues. Limited access to administrative or configuration settings.

## Operating Environment

**Hardware Requirements:**

Processor: Refers to the central processing unit (CPU), which executes instructions and processes data. A more powerful processor leads to faster performance in handling tasks.

RAM (Random Access Memory): Temporary storage that the system uses for active tasks. Sufficient RAM allows for smoother multitasking and handling of larger datasets.

Storage: The available space on the device's storage drive, where the system and data are stored. Adequate storage is necessary for installation and data management.

Display: The screen's resolution, measured in pixels, which affects the quality and clarity of visuals.

Internet Connectivity: A stable and preferably high-speed internet connection is essential for real-time updates, online transactions, and communication with external services like payment gateways.

**Software Requirements:**

Operating Systems: Specifies the compatible operating systems for running the Hotel Reservation System. Different systems have their own requirements and compatibility.

Web Browsers: Applications used to access web-based interfaces. Compatibility with popular web browsers ensures accessibility for users.

Database Management System: Software used to store and manage data. The system must be compatible with specific database management systems like MySQL or PostgreSQL.

Web Server: Software that hosts the web application. Compatibility with specific web servers ensures the system can be deployed on various hosting environments.

Programming Languages and Frameworks: Technologies used to develop the system. Compatibility with specific versions of languages like PHP and JavaScript, as well as frameworks like Laravel and Vue.js, ensures the system functions correctly.

Additional Software Components: Third-party tools or services that the system may integrate with, such as payment gateways (e.g., Stripe, PayPal), email servers (SMTP), and security certificates (SSL/TLS).

## Design and Implementation Constraints

* **Regulatory Compliance:** Adherence to industry regulations and legal requirements.
* **Hardware Optimization:** Efficient operation within specified hardware requirements.
* **Integration Needs:** Seamless interaction with external systems like payment gateways.
* **Technology Stack:** Utilizing defined programming languages, frameworks, and databases.
* **Security Measures**: Robust data protection, encryption, and access controls.
* **User Interface Standards:** Adherence to established design conventions.
* **Organizational Guidelines:** Compliance with programming and design standards.
* **Concurrency Support:** Ability to handle multiple users simultaneously.
* **Multilingual Support:** Availability in relevant languages.
* **Backup and Recovery:** Mechanisms for data protection and retrieval.
* **Scalability Planning:** Design for potential user and transaction growth.
* **Maintenance Considerations:** End-user organization responsible for ongoing support and updates.

## User Documentation

* user interface prototypes
* system design
* user guide book

## Assumptions and Dependencies

**Assumptions:**

* Third-party Payment Gateways: It is assumed that the chosen payment gateways (e.g., Stripe, PayPal) will maintain their current APIs and integration methods throughout the project development.
* Stable Internet Connection: It is assumed that end-users will have access to a stable internet connection for seamless use of the system.
* Regulatory Compliance: It is assumed that the legal and regulatory environment governing online reservations and payments will remain consistent with current standards.
* Hardware Compatibility: It is assumed that end-users' devices meet the specified hardware requirements for optimal system performance.

**Dependencies:**

**Database Management System (DBMS):** The project is dependent on the chosen DBMS (e.g., MySQL, PostgreSQL) for data storage and retrieval. Any changes in DBMS might require adjustments to the system.

**Web Server Environment:** The system relies on a specific web server environment (e.g., Apache, Nginx) to host the application. Compatibility with the chosen web server is crucial for deployment.

**External APIs:** Dependencies exist on external APIs for tasks like payment processing and authentication. Any changes or discontinuation of these APIs may impact system functionality.

**Language and Framework Versions:** The development is dependent on specific versions of programming languages (e.g., PHP, JavaScript) and frameworks (e.g., Laravel, Vue.js) outlined in the project requirements.

**Compliance with Organizational Standards:** Adherence to organizational programming and design standards is necessary for consistency and compatibility with existing systems.

**Internet Security Protocols:** The system depends on SSL/TLS for secure data transmission. Any changes in security protocols may require corresponding updates.

**Organizational Resources:** Availability of resources and personnel within the organization, including developers, testers, and system administrators, is a critical dependency for successful project implementation.

**Vendor Support and Maintenance:** Third-party components or services (e.g., payment gateways) are dependent on the continued support and maintenance provided by the respective vendors.

External Interface Requirements

User Interfaces

Sample Screen Images: Visual representations illustrating stages of the reservation process, showcasing search results, booking details, and confirmation screens.

**GUI Standards and Style Guides**: Adherence to established Graphical User Interface (GUI) standards, ensuring consistency in design elements like buttons, navigation bars, and input fields. The system follows the corporate GUI style guide for a unified look and feel.

**Screen Layout Constraints:** Designs optimized for usability, incorporating considerations for readability, accessibility, and logical flow of information.

**Standard Buttons and Functions:** Common functions such as "Search," "Book," "Modify," "Cancel," and "Confirm" available on relevant screens for seamless navigation.

**Help Functionality:** A dedicated "Help" button or section providing context-specific assistance, with links to relevant sections in the user manual.

**Keyboard Shortcuts:** Standard keyboard shortcuts implemented for actions like saving, undoing, and navigating, enhancing user productivity.

**Error Message Display Standards:** Clear, informative error messages presented in a consistent format in cases of incorrect inputs or system errors, guiding users toward corrective actions.

**Responsive Design:** Interfaces designed to adapt to various screen sizes and devices (desktop, tablet, mobile), ensuring an optimal user experience across platforms.

**Language Support:** Multi-language support enabling users to select their preferred language for interaction, enhancing accessibility for diverse user bases.

**Accessibility Considerations:** Interface design complies with accessibility standards, ensuring usability for users with disabilities, including screen readers and keyboard navigation.

**Feedback Mechanism:** Visual feedback indicators, such as loading animations, provided during processes requiring some time to complete, offering transparency to users.

**Integration Interfaces:** Seamless integration points for external services, including payment processing and interactions with external APIs, ensuring a cohesive user experience.

**Hardware Interfaces**

Supported Device Types:

The system supports a range of devices, including desktop computers, laptops, tablets, and mobile phones. It is designed to be responsive, adapting to different screen sizes and resolutions.

Data and Control Interactions: The interactions between the software and hardware components involve data exchange for tasks such as processing reservations, accessing room availability, and handling payment transactions.

Communication Protocols: The system utilizes standard communication protocols, such as HTTP/HTTPS for web-based interactions between the client's device and the system's servers. Secure Socket Layer (SSL) or Transport Layer Security (TLS) protocols are employed to encrypt data during transmission, ensuring security.

Peripheral Devices: The system may interact with peripheral devices such as printers for generating receipts or reports, card readers for processing payments, and barcode scanners for checking in guests.

Internet Connectivity: A stable internet connection is a crucial hardware requirement. It allows the system to communicate with external services, synchronize data, and process online transactions in real-time.

Hardware Security Measures: The system may employ hardware-level security features, such as encryption modules or biometric authentication devices, to enhance data protection and access control.

Compatibility with Operating Systems: The system is designed to be compatible with various operating systems (e.g., Windows, macOS, Linux) that users may have installed on their devices.

Browser Compatibility: The system is compatible with a range of web browsers (e.g., Chrome, Firefox, Safari, Edge) to ensure users can access and interact with the application seamlessly.

Camera and Microphone Access (if applicable): If the system incorporates video conferencing or virtual tour features, it may require access to the device's camera and microphone for user interactions.

## **Software Interfaces**

1. Database Management System (DBMS):

Type: MySQL (v5.7+) or PostgreSQL (v10+).

Purpose: The system interacts with the chosen DBMS to store and retrieve data related to

reservations, guest information, room availability, and transaction records.

1. Operating System:

Supported: Windows 10 (64-bit), macOS 10.12+, Ubuntu 18.04+.

Purpose: The system is compatible with these operating systems to ensure users can access and utilize the application on their respective devices.

1. Web Server:

Supported: Apache (v2.4+), Nginx (latest).

Purpose: The system is designed to run on specific web server environments, providing the necessary infrastructure for hosting the application.

1. Programming Languages and Frameworks:

Languages: PHP (v7.0+), JavaScript (ES6+).

Frameworks: Laravel (v5.7+), Vue.js (v2.5+).

Purpose: These technologies form the foundation for development, enabling the creation of dynamic and interactive features within the system.

1. Payment Gateway Integration:

Specific Gateways: (e.g., Stripe, PayPal).

Purpose: Integration with these payment gateways allows the system to securely process online payments for reservations and other services.

1. Authentication Services:

Type: OAuth 2.0 (or equivalent).

Purpose: Authentication services provide secure access control for both guests and administrators, ensuring only authorized users can perform specific actions within the system.

1. Email Services (SMTP):

Purpose: The system utilizes SMTP protocols to send transactional emails, including reservation confirmations, booking details, and other notifications to guests.

1. Purpose: These certificates ensure secure communication between the client and the server, encrypting data to protect it from unauthorized access or interception.

## **Communications Interfaces**

* Email Communication:

Requirements: The system must be capable of sending and receiving emails for purposes such as reservation confirmations, notifications, and support communications.

Message Formatting: Emails will be formatted in HTML for a visually appealing and informative presentation.

Communication Standard: SMTP (Simple Mail Transfer Protocol) will be used for sending and receiving emails.

Security: Email communication will be secured using TLS (Transport Layer Security) encryption to protect sensitive information.

* Web Browser Interface:

Requirements: The system is accessed through web browsers on various devices, enabling users to interact with the application.

Message Formatting: Web pages will be rendered in HTML, CSS, and JavaScript for an interactive user experience.

Communication Standard: HTTP/HTTPS protocols will be used for data exchange between the client's browser and the server.

Security: HTTPS (HTTP Secure) will be enforced using SSL/TLS encryption to secure data in transit.

* Network Server Communications:

Requirements: The system relies on network server communications to manage client-server interactions for tasks like making reservations, checking availability, and processing payments.

Message Formatting: Data will be transmitted in JSON format for efficient parsing and processing on both ends.

Communication Standard: HTTP/HTTPS will be used for network server communications.

Security: Secure connections (HTTPS) will be enforced to protect data during transmission.

* Electronic Forms:

Requirements: The system may utilize electronic forms for tasks like guest information input, reservation details, and special requests.

Message Formatting: Forms will be designed in HTML with appropriate input fields and validation rules.

Communication Standard: Form data will be submitted using HTTP/HTTPS protocols.

Security: Input validation and server-side validation will be implemented to prevent unauthorized data submission.

* Data Transfer Rates and Synchronization:

Requirements: The system must support efficient data transfer rates to ensure timely processing of reservations and other tasks.

Synchronization: Real-time synchronization will be implemented for critical functions such as availability updates and payment processing to maintain up-to-date information.

* Communication Security and Encryption:

Requirements: All communications, including email, web browser interactions, and server communications, must be secured to protect sensitive information.

Encryption: TLS (Transport Layer Security) will be used to encrypt data during transmission, ensuring secure communication channels.

# System Features

System Feature 1

4.1.1 Description and Priority:

This feature allows users to create accounts on the Hotel Reservation System, providing access to personalized services, booking history, and preferences. Priority: High.

4.1.2 Stimulus/Response Sequences:

Stimulus: User clicks on "Register" button.

Response: System displays registration form.

Stimulus: User fills in registration form and submits.

Response: System validates information and creates user account. If successful, user receives a confirmation email.

Stimulus: User clicks on confirmation link in email.

Response: System verifies email address and activates the user account.

4.1.3 **Functional Requirements:**

REQ-1: User Registration Form

The system shall provide a registration form with fields for username, email address, password, and optional profile details.

The system shall validate that the email address is in a valid format.

REQ-2: Unique Username and Email

The system shall ensure that usernames and email addresses used for registration are unique.

REQ-3: Password Requirements

The system shall enforce password requirements, including a minimum length and complexity criteria.

REQ-4: Email Confirmation

Upon successful registration, the system shall send a confirmation email to the provided address with a verification link.

REQ-5: Account Activation

The system shall activate the user's account upon clicking the confirmation link in the email.

REQ-6: Error Handling

The system shall display appropriate error messages for invalid inputs or unsuccessful registration attempts.

REQ-7: Optional Profile Details

Users may optionally provide additional profile details such as full name, contact number, and preferences.

REQ-8: CAPTCHA Verification

The system may implement CAPTCHA verification to prevent automated bot registrations.

# Other Nonfunctional Requirements

## Performance Requirements

**Page Load Times:**

Requirement: All pages must load within 3 seconds on standard internet connections.

Rationale: Fast page loading times improve user satisfaction and engagement, reducing the likelihood of bounce rates.

**Search Response Time:**

Requirement: The system must return search results for available rooms within 2 seconds.

Rationale: Swift response times are essential for providing users with real-time information about room availability.

**Transaction Processing Time:**

Requirement: Transaction processing, including booking confirmation and payment, must be completed within 5 seconds.

Rationale: Efficient transaction processing ensures that users can swiftly complete their reservations without delays.

**Concurrent User Handling:**

Requirement: The system must support a minimum of 1000 concurrent users without significant performance degradation.

Rationale: The system should be able to handle a high volume of users, especially during peak reservation periods.

**Database Query Speed:**

Requirement: Database queries must execute within 200 milliseconds.

Rationale: Fast database queries are crucial for retrieving and updating information in real time.

**Error Handling Time:**

Requirement: Error messages must be generated and displayed within 2 seconds of an error occurrence.

Rationale: Prompt error handling provides users with immediate feedback and instructions for resolution.

**Server Uptime:**

Requirement: The system must maintain at least 99.9% uptime on an annual basis.

Rationale: High availability ensures that the system is accessible to users at all times, minimizing disruptions to reservations.

**Data Synchronization Frequency:**

Requirement: Data synchronization between servers and external services must occur at least once every 15 minutes.

Rationale: Regular data synchronization ensures that information remains up-to-date and accurate.

**Load Balancing and Scalability:**

Requirement: The system must employ load balancing techniques and be scalable to accommodate increased user loads.

Rationale: Load balancing and scalability are essential for distributing traffic and maintaining performance under varying workloads.

**Geographic Load Distribution:**

Requirement: The system should be designed to distribute load geographically to minimize latency for users in different regions.

Rationale: This ensures a consistent user experience regardless of geographical location.

## Safety Requirements

**Page Load Times:**

Requirement: All pages must load within 3 seconds on standard internet connections.

Rationale: Fast page loading times improve user satisfaction and engagement, reducing the likelihood of bounce rates.

**Search Response Time:**

Requirement: The system must return search results for available rooms within 2 seconds.

Rationale: Swift response times are essential for providing users with real-time information about room availability.

**Transaction Processing Time:**

Requirement: Transaction processing, including booking confirmation and payment, must be completed within 5 seconds.

Rationale: Efficient transaction processing ensures that users can swiftly complete their reservations without delays.

**Concurrent User Handling:**

Requirement: The system must support a minimum of 1000 concurrent users without significant performance degradation.

Rationale: The system should be able to handle a high volume of users, especially during peak reservation periods.

**Database Query Speed:**

Requirement: Database queries must execute within 200 milliseconds.

Rationale: Fast database queries are crucial for retrieving and updating information in real time.

**Error Handling Time:**

Requirement: Error messages must be generated and displayed within 2 seconds of an error occurrence.

Rationale: Prompt error handling provides users with immediate feedback and instructions for resolution.

**Server Uptime:**

Requirement: The system must maintain at least 99.9% uptime on an annual basis.

Rationale: High availability ensures that the system is accessible to users at all times, minimizing disruptions to reservations.

**Data Synchronization Frequency:**

Requirement: Data synchronization between servers and external services must occur at least once every 15 minutes.

Rationale: Regular data synchronization ensures that information remains up-to-date and accurate.

## Security Requirements

**Data Security:**

Requirement: The system must implement robust encryption protocols (SSL/TLS) to protect sensitive user information during transmission.

Rationale: Encryption safeguards user data from unauthorized access or interception.

**User Authentication:**

Requirement: The system must enforce strong password policies to prevent unauthorized access to user accounts.

Rationale: Strong passwords deter unauthorized access and protect user accounts from compromise.

**Data Privacy Compliance:**

Requirement: The system must comply with relevant data privacy regulations, such as GDPR, to ensure the protection of user data.

Rationale: Compliance with data privacy regulations is essential to safeguard user privacy and rights.

**Secure Payment Processing:**

Requirement: The system must use secure and trusted payment gateways to process financial transactions.

Rationale: Secure payment processing protects users' financial information from fraudulent activities.

**Access Control:**

Requirement: The system must implement access control measures to ensure that only authorized users have access to sensitive functionality or data.

Rationale: Access control prevents unauthorized users from performing actions that could lead to data loss or system disruption.

**Data Backup and Recovery:**

Requirement: The system must have regular data backup procedures in place, with the ability to quickly recover in case of data loss or system failure.

Rationale: Data backup and recovery procedures mitigate the risk of data loss due to unexpected events.

## Software Quality Attributes

**Usability:**

Requirement: The system must achieve a minimum score of 80 out of 100 on standardized usability testing.

Rationale: Usability is crucial for ensuring a seamless and intuitive user experience.

**Reliability:**

Requirement: The system must have a Mean Time Between Failures (MTBF) of at least 1000 hours.

Rationale: Reliability ensures that the system functions consistently without frequent interruptions or failures.

**Availability:**

Requirement: The system must be available for use at least 99.9% of the time, excluding scheduled maintenance.

Rationale: High availability guarantees that users can access the system whenever needed.

**Maintainability:**

Requirement: The system must have a code maintainability index of at least 80, as measured by static code analysis tools.

Rationale: Maintainability facilitates efficient updates, bug fixes, and enhancements to the system.

**Security:**

Requirement: The system must score an A grade or higher on security assessments based on industry-standard frameworks (e.g., OWASP Top 10).

Rationale: Security is paramount for safeguarding user data and preventing unauthorized access.

**Performance Efficiency:**

Requirement: The system must handle a minimum of 1000 concurrent users with a response time of under 2 seconds for critical operations.

Rationale: Performance efficiency ensures that the system remains responsive even under high load.

**Scalability:**

Requirement: The system must be capable of scaling to accommodate a 50% increase in user load without significant performance degradation.

Rationale: Scalability allows the system to accommodate growing user demands.

## Business Rules

1. **User Roles and Permissions:**

Rule: Only registered users with a valid account can make reservations.

Enforcement: Implement user authentication and authorization mechanisms to verify user identity and permissions.

1. **Booking Window:**

Rule: Reservations can be made up to six months in advance.

Enforcement: The system should restrict reservation dates beyond the specified window.

1. **Cancellation Policy:**

Rule: Users can cancel their reservations up to 24 hours before the check-in date without incurring a cancellation fee.

Enforcement: Implement a cancellation mechanism that enforces this policy.

1. **Room Availability:**

Rule: A room can only be booked if it is available for the selected dates.

Enforcement: Implement real-time availability checks before confirming a reservation.

1. **Maximum Stay Duration:**

Rule: The maximum duration for a single reservation is 30 days.

Enforcement: Set constraints on reservation durations in the system.

1. **Check-In and Check-Out Times:**

Rule: Check-in time is at 3:00 PM, and check-out time is at 11:00 AM.

Enforcement: Display and enforce these times during the reservation process.

1. **Special Requests:**

Rule: Users can submit special requests (e.g., room preferences, dietary needs) during the reservation process.

Enforcement: Provide a designated field for users to enter special requests.

# Other Requirements

* **User Feedback and Ratings:**

The system should allow users to provide feedback and ratings for their stay, which can be used to improve services and assist future guests in their decision-making.

* **Analytics and Reporting:**

The system must include analytics and reporting features to track key performance indicators (KPIs) such as occupancy rates, booking trends, and revenue.

* **Backup and Disaster Recovery:**

Regular backups of the database and system configurations should be performed, with a documented disaster recovery plan in place to ensure data integrity and availability.

**Appendix A: Glossary**

SRS: Software Requirements Specification

RDBMS: Relational Database Management System

WCAG: Web Content Accessibility Guidelines

PCI DSS: Payment Card Industry Data Security Standard

PMS: Property Management System

KPIs: Key Performance Indicators

**Appendix C: To Be Determined List**

TBD - Database schema design for room details

TBD - Integration method with external mapping APIs

TBD - Data privacy compliance framework and procedures

TBD - Backup frequency and disaster recovery plan details

TBD - Specific user roles and permissions matrix

* **Justification for Agile with Iterative and Incremental Approach:**

**Frequent Deliveries:** Agile methodologies emphasize delivering working software in short iterations. For a Hotel Reservation System, this means that usable features can be delivered quickly, allowing stakeholders to see progress early on.

**Customer Collaboration:** Agile encourages continuous collaboration with stakeholders, including guests, hotel staff, and administrators. Their feedback is integrated into each iteration, ensuring that the system meets their specific needs and preferences.

**Adaptability to Change:** The hotel industry is subject to changing trends, customer expectations, and business requirements. Agile's flexibility allows for easy incorporation of changes and adjustments, ensuring that the system remains aligned with evolving needs.

**Continuous Improvement:** With each iteration, the system is refined and improved based on feedback and lessons learned. This iterative process leads to a higher quality product and reduces the risk of late-stage surprises or deviations from requirements.

**Risk Mitigation:** By breaking the development process into smaller increments, risks are identified and addressed early. This helps in avoiding costly rework and ensures that potential issues are dealt with in a timely manner.

**Incremental Functionality:** The system is built incrementally, with each iteration adding new features or improving existing ones. This allows for a phased approach to development, where critical functionalities can be implemented first, followed by less critical ones.

**Empowered Development Teams:** Agile empowers development teams to make decisions and adapt to changing circumstances. This can lead to a more motivated and engaged team, resulting in higher quality work.

**Transparent Progress Tracking:** Agile methodologies provide visibility into the progress of the project through techniques like sprint planning, daily stand-ups, and burndown charts. This transparency helps in managing expectations and making informed decisions.

**Appendix B: Analysis Models**

**EXPANDED USE CASES**

## Use Case 1

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Authentication** | |
| **Primary Actors** | Guests, Hotel staff | |
| **Pre-condition** | For signup, users do not already have an account in the Hotel Reservation System, but they do have registered accounts with acceptable credentials for login. | |
| **Post condition** | Users who successfully sign up are given access to the system, and those who successfully log in are given a new account with an activation confirmation. | |
| **Main Success Scenario** | Step No. | Action |
| 1 | On the login screen, the Guests chooses the "Register" option. |
| 2 | The Guests enters the necessary information, including their name, email address, password, and Guests ID. |
| 3 | When users successfully log in with valid credentials, their individual dashboards are immediately accessible. |
| 4 | Users easily create new accounts during registration, get emails asking them to confirm their activation, and are greeted with a message of welcome after a successful activation. |
| **Extension** |  | During registration, a password strength meter advises users to create strong passwords for increased security. |

## 

## Use Case 2

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Manage Appointment** | |
| **Primary Actors** | Guests, Hotel staff | |
| **Precondition** | The Hotel Reservation System is logged in with users. | |
| **Post condition** | * The user's updated appointment schedule reflects appointments that have been successfully rescheduled or cancelled. * For accountability and reference, users can review their appointment history. | |
| **Main Success Scenario** | Step No. | Action |
| 1 | Users can access the "Manage Appointments" feature after logging into the Hotel Reservation System. |
| 2 | The system displays a concise list of the user's planned appointments together with valuable information like the date and time. |
| 3 | The system quickly updates the appointment schedule after users select a specific appointment, smoothly select a new time from the available slots, and observe the system in action. |
| 4 | Users may easily access the cancellation option, confirm their choice, and see as the system successfully deletes the appointment from the calendar. |
| 5 | Users can view their thorough appointment history to obtain knowledge of prior conversations and exchanges for improved continuity. |
| **Extensions** |  | * Users receive automatic appointment reminders, which improve timely awareness and attendance. * Users should have the option to add agenda items or notes to planned appointments because this adds more context for more fruitful conversations. |

## 

## Use Case 3

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Manage user** | |
| **Primary Actors** | Administrator, Technician | |
| **Precondition** | The administrator is logged in Hotel Reservation System. | |
| **Post condition** | * According to changes made by the administrator, user information is updated. * A user's access to the system is immediately restricted by deactivating their account. * The user receives an email with instructions on how to change their password securely. | |
| **Main Success Scenario** | Step No. | Action |
| 1 | The administrator enters the SI.  system successfully and moves without any problems to the "Manage Users" section. |
| 2 | The system accurately presents a complete list of users who have registered, classifying them according to roles and making all pertinent information available. |
| 3 | Administrators effectively change user information, guaranteeing its accuracy and reflecting adjustments in real-time. |
| 4 | When a user account is successfully deactivated, access is instantly blocked, and the system records the activity. |
| 5 | Users can easily recover if administrators request a password reset for them. The system automatically sends a secure password reset link to the user's registered email. |
| **Extensions** |  | * To take preventative security steps, administrators are alerted about unexpected account activity. * Administrators can bulk-edit user details, streamlining procedures for significant revisions, to improve the efficiency of user management. |

## Use Case 4

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Notify Guests** | |
| **Primary Actors** | System | |
| **Pre-condition** | Automated contact with Guests is required when pertinent events occurring within the Hotel Reservation System, such as appointment updates or system announcements, trigger messages. | |
| **Post condition** | Guests are made more aware of and involved with the Hotel Reservation System through timely and accurate messages. | |
| **Main Success Scenario** | Step No. | Action |
| 1 | The Guests logs in to their account. |
| 2 | The Guests accesses the Hotel Reservation System. |
| 3 | The Guests navigates to the "Hotel staff Availability" section. |
| 4 | The system presents a list of hotel staffs. |
| 5 | The Guests selects a specific Hotel staff from the list. |
| **Extension** |  | * Guests can choose their preferred communication methods (such as email or in-app notifications) when setting up their notification options, which will increase accessibility. * The system may escalate notifications to alternative communication means, including SMS, in the case of urgent information or essential system modifications to guarantee rapid awareness. |

## Use Case 5

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Virtual Queue** | |
| **Primary Actors** | Guests | |
| **Pre-condition** | The Guests has signed in.A Room has been chosen by the Guests for an appointment. The waiting room for hotel staffs has open spots. | |
| **Post condition** | The Guests has successfully joined the virtual queue for their chosen Hotel staff. | |
| **Main Success Scenario** | Step No. | Action |
| 1 | The Guests logs in to their SI account. |
| 2 | The Guests selects "Join Queue" from the menu. |
| 3 | A list of hotel staffs who are available for Rooms is displayed by the system. |
| 4 | The Guests selects a specific Hotel staff from the list. |
| 5 | The system places the Guests in the virtual queue for that Hotel staff. |
| 6 | The system uses past data to determine the Guests’ approximate waiting time. |
| 7 | The Guests can view their position in the queue and the estimated wait time. |
| **Extension** |  | If their availability changes or if they experience unexpected delays, the system gives Guests the option to leave the virtual line and reschedule their appointment, ensuring flexibility in scheduling. |

## Use Case 6

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Location Guidance** | |
| **Primary Actors** | Hotel staff | |
| **Pre-condition** | The Hotel staff is signed in. In their hotel system profile, the Hotel staff has specified the location of their office. | |
| **Post condition** | Directions to the Hotel staff's office are available to the Hotel staff or Guests, making scheduling appointments simple. | |
| **Main Success Scenario** | Step No. | Action |
| 1 | The Hotel staff logs in to their Hotel Reservation account. |
| 2 | The Hotel staff accesses the "My Office Location" section. |
| 3 | The system displays a map interface with the Hotel staff's office location marked. |
| 4 | The Hotel staff can choose to:   * View Office Location * Generate Directions * Share Directions |
| 5 | The Hotel staff or Guests can follow the provided directions to reach the Hotel staff's office. |
| **Extension** |  | With the addition of the direction help tool and the option for hotel staffs to enter their office hours and availability status, the system gives Guests accurate data about when Hotel staffs are in their offices. |

## Use Case 7

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **View Room Availability** | |
| **Primary Actors** | Guests | |
| **Pre-condition** | Guests must be logged in and system shows the Room Availability of Hotel staffs to Guests. | |
| **Post condition** | The Guests can review their room schedule. | |
| **Main Success Scenario** | Step No. | Action |
| 1 | The Guests logs in successfully, chooses a hotel staff, and makes a session for a specific type of a discussion. |
| 2 | Efficient scheduling is made possible by the system's accurate real-time availability display. |
| 3 | When a Room is changed the notification messages are sent out immediately. With successful scheduling , the Guests enters a virtual queue for enhanced scheduling of Rooms. |
| **Extension** |  | The system rapidly updates and reflects any revisions for the Guests if the Room Availability for class is changed. |

## Use Case 8

|  |  |  |
| --- | --- | --- |
| **Use case Name** | **Room Type** | |
| **Primary Actors** | Guests, Hotel staff | |
| **Pre-condition** | The Guests needs to be logged in the system and available Hotel staff should be shown so the Guests select the Hotel staff of choice available at that time. | |
| **Post condition** | Both Hotel staff and Guests notified about the confirmation of Room and their Room Type. | |
| **Main Success Scenario** | Step No. | Action |
| 1 | After successfully logging in, the Guests chooses a hotel staff who is available and makes a Room  for a certain sort of conversation. |
| 2 | A confirmation email is provided to the Guests after the Hotel staff validates and confirms the appointment. |
| 3 | When making an appointment, the Guests successfully views their schedule. |
| 4 | The system provides an accurate and timely way to schedule Rooms based on their availability. |
| 5 | When setting up a Room, notification messages are sent out on time, ensuring effective communication. |
| **Extension** |  | The Guests is informed if the Hotel staff declines the Room. |

## **HIGH LEVEL USE CASES**

**Use Case 1**

|  |  |
| --- | --- |
| **Use case ID:** | 1 |
| **Use case Name:** | Authentication |
| **Actor** | Guests, Hotel staff |
| **Type:** | Primary. |
| **Description:** | Allow users to safely enter their credentials to log into their accounts. |

**Use Case 2**

|  |  |
| --- | --- |
| **Use case ID:** | 2 |
| **Use case Name:** | Manage Appointment |
| **Actor** | Guests, Hotel staff |
| **Type:** | Primary. |
| **Description:** | Makes it easier for Guests and Hotel staffs to schedule Rooms. |

**Use Case 3**

|  |  |
| --- | --- |
| **Use case ID:** | 3 |
| **Use case Name:** | Manage Guests |
| **Actor** | Admin |
| **Type:** | Primary. |
| **Description:** | Manage guests if faces some issue while using system |

**Use Case 4**

|  |  |
| --- | --- |
| **Use case ID:** | 4 |
| **Use case Name:** | Join Virtual Queue |
| **Actor** | Guests |
| **Type:** | Primary |
| **Description:** | Guests can take advantage of expected wait times and a well-structured scheduling room processing by joining a virtual queue for appointments. This guarantees that scheduling is done in a streamlined and effective manner. |

**Use Case 5**

|  |  |
| --- | --- |
| **Use case ID:** | 5 |
| **Use case Name:** | Get Direction Office |
| **Actor** | Hotel staff |
| **Type:** | Primary |
| **Description:** | Allows Guests to get location help, which makes it easier for them to find their hotel staff’s offices. The system's overall user experience is improved by this functionality, which guarantees easy navigation for planned appointments. |

**Use Case 6**

|  |  |
| --- | --- |
| **Use case ID:** | 6 |
| **Use case Name:** | Cancel Appointment |
| **Actor** | Guests, Hotel staff |
| **Type:** | Secondary. |
| **Description:** | Guests will be notified in cancel appointment case. |

**Use Case 7**

|  |  |
| --- | --- |
| **Use case ID:** | 7 |
| **Use case Name:** | Guide User |
| **Actor** | Admin ,Technician |
| **Type:** | Secondary. |
| **Description:** | They will help Guests having any issue while using app and takes answer to question also. |

**Use Case 8**

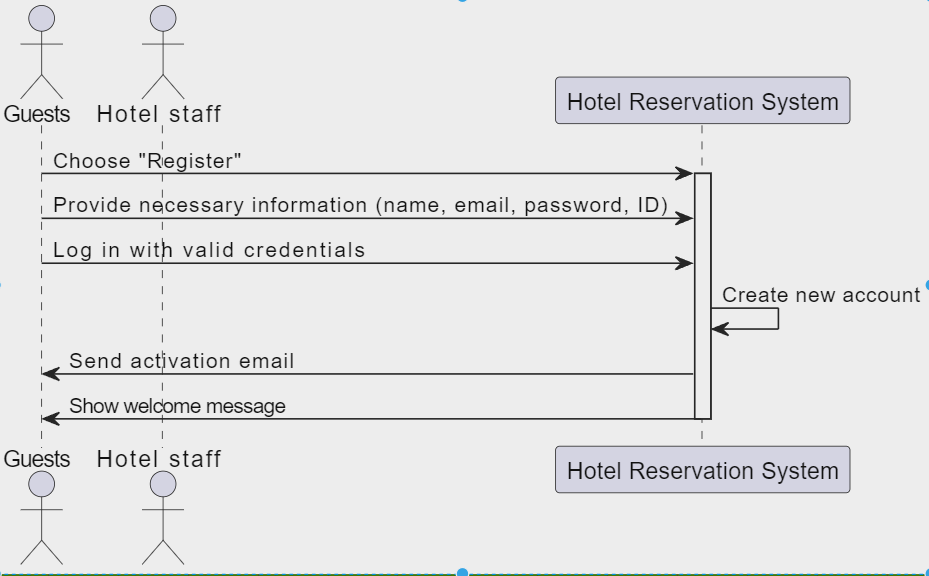
|  |  |
| --- | --- |
| **Use case ID:** |  |
| **Use case Name:** | Availability of Hotel staff |
| **Actor** | Guests |
| **Type:** | Primary |
| **Description:** | Gives Guests access to hotel staff’s real-time availability, enabling careful  appointment scheduling. By giving clear and current information about hotel staffs' timetables, this feature improves the effectiveness of Guests interactions with hotel staffs. |

**Use Case 9**

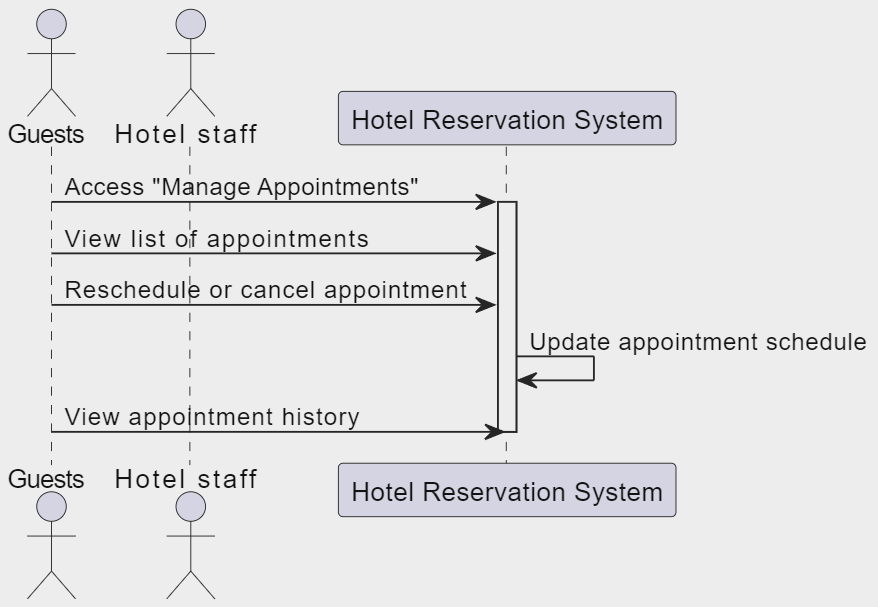
|  |  |
| --- | --- |
| **Use case ID:** | 9 |
| **Use case Name:** | Receive Notification |
| **Actor** | Guests, Hotel staff |
| **Type:** | Primary |
| **Description:** | Notifies Hotel staff and Guests right away of any changes, cancellations, or new appointment requests. This function improves the effectiveness of communication and guarantees that all in the scheduling process receive updates on time. |

**SYSTEM SEQUENCE DIAGRAM**

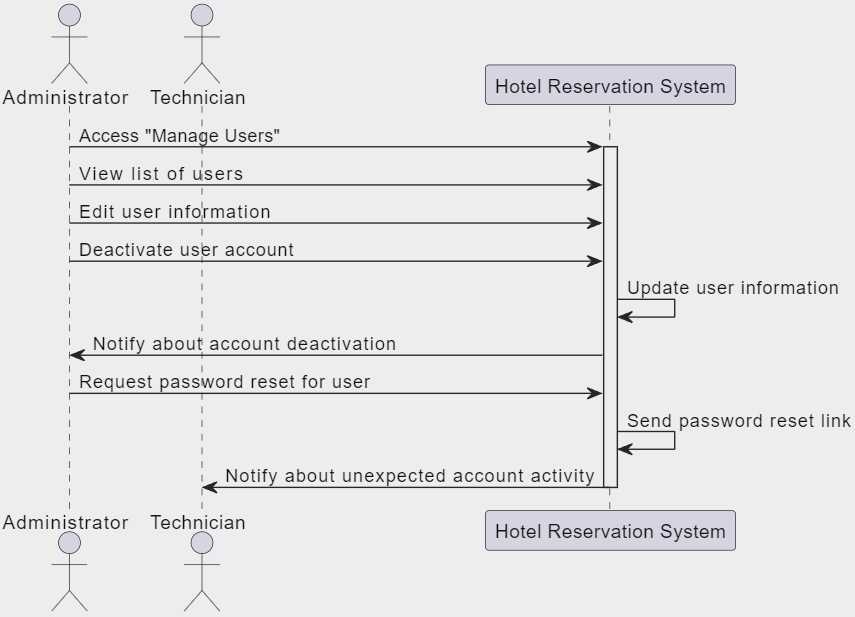
1. **Authentication**

****

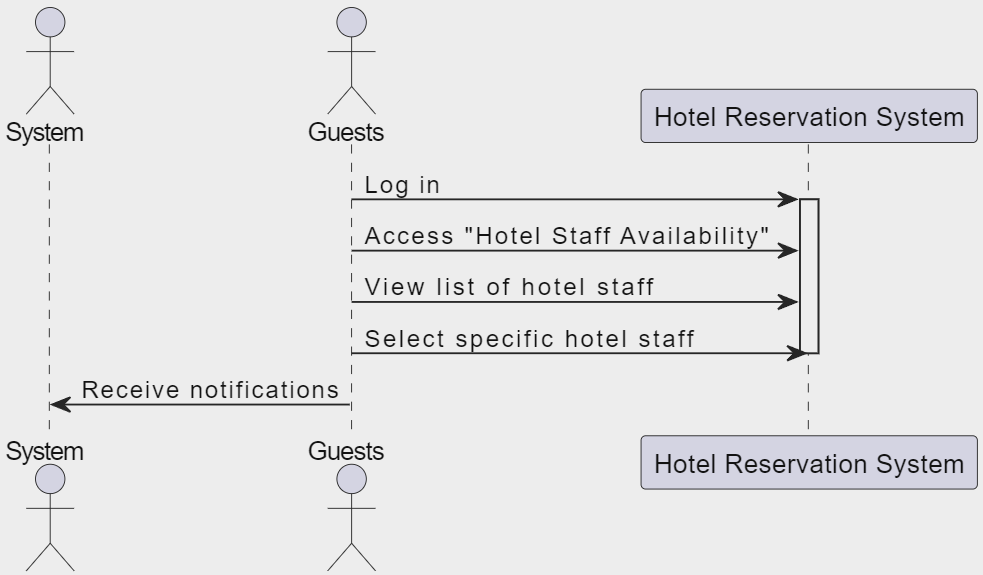
1. **Manage Appointment**

****

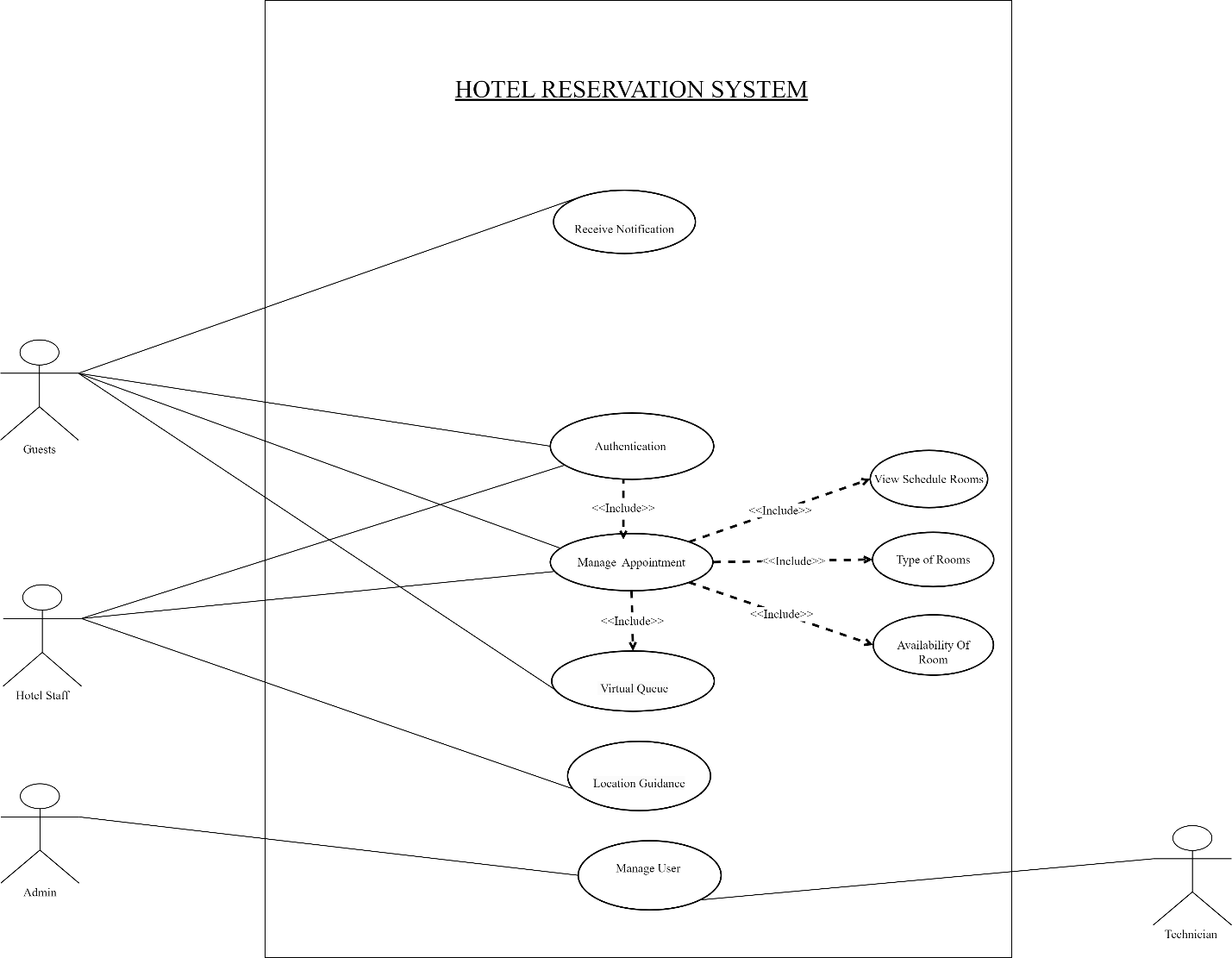
1. **Manage user**

****

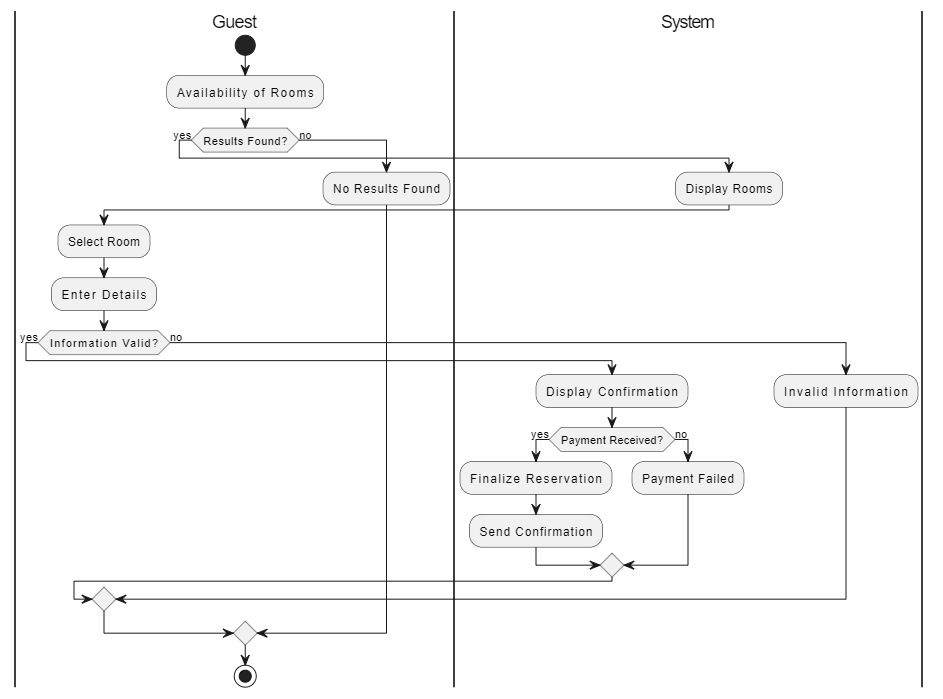
1. **Notify Guests**

****

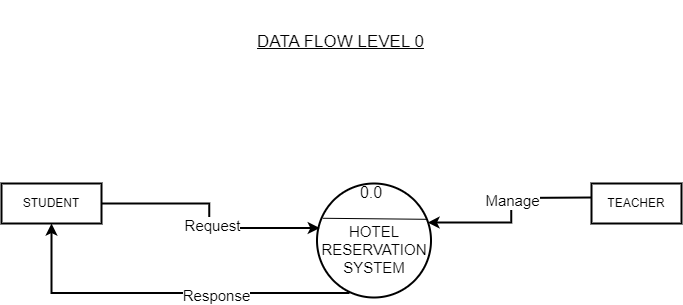
**USE CASES DIAGRAM**

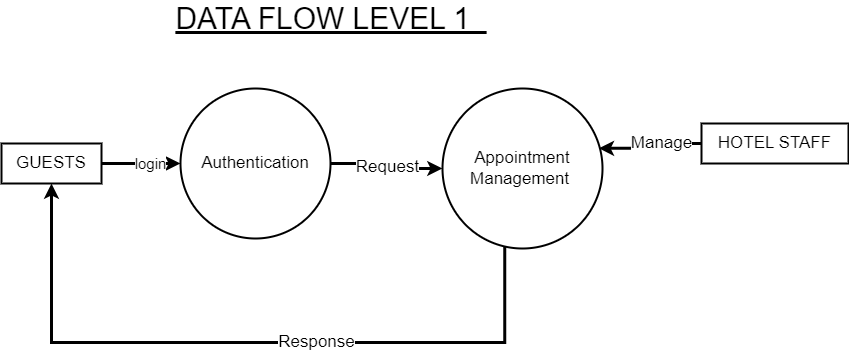


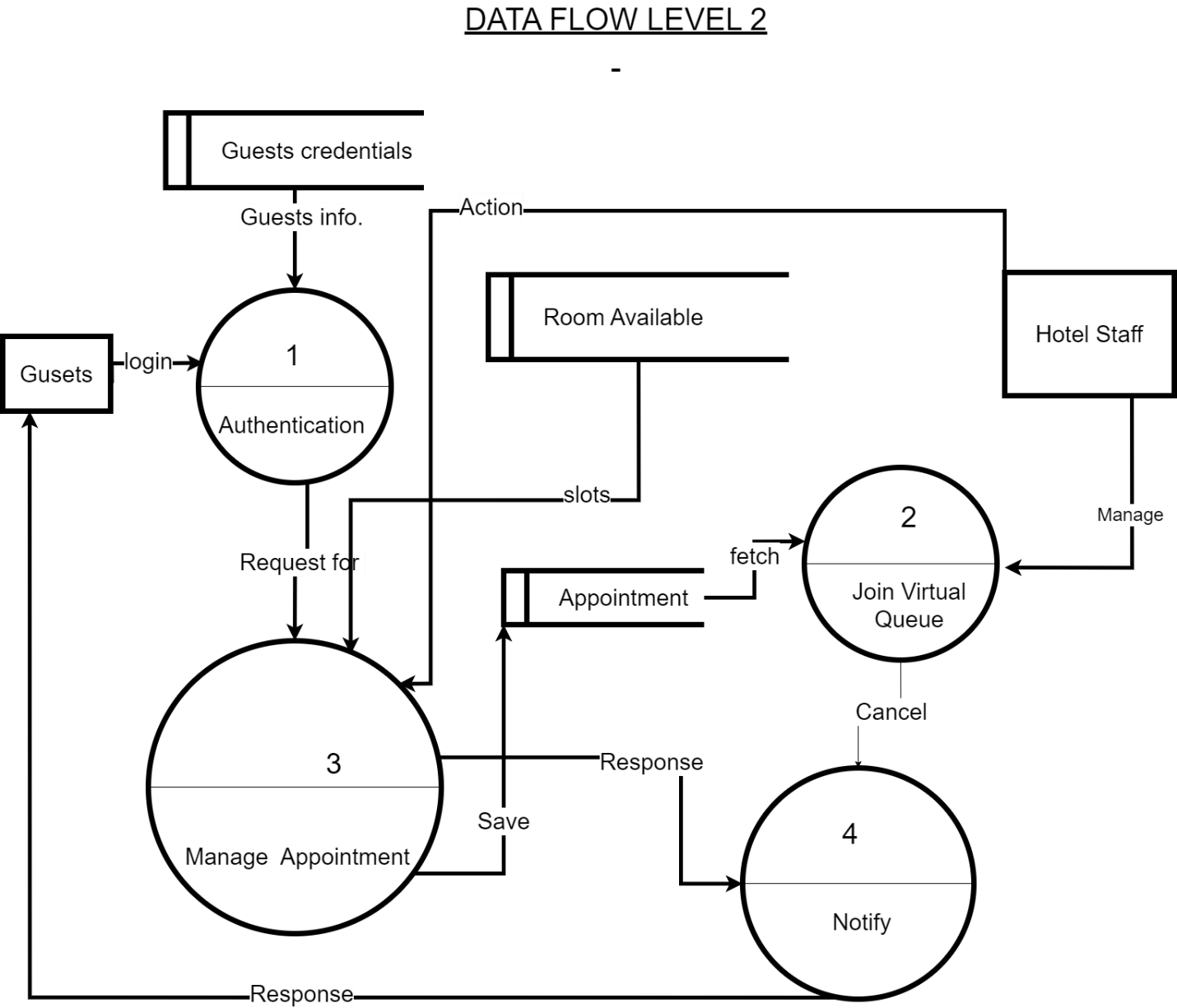
**ACTIVITY DIAGRAM**



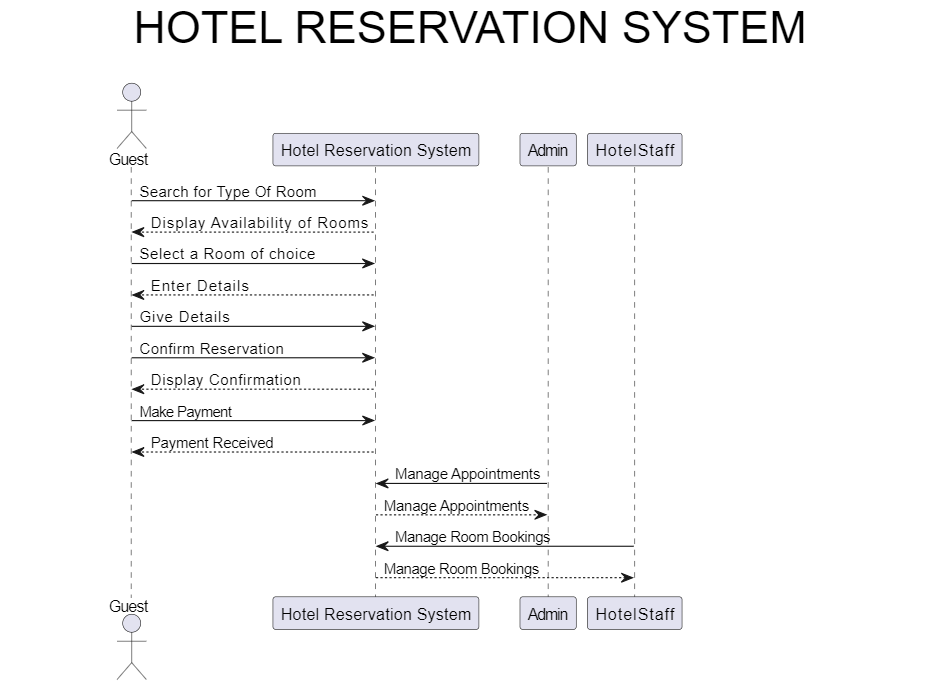
**DATA FLOW DIAGRAM**

****

****



**SEQUENCE DIAGRAM**

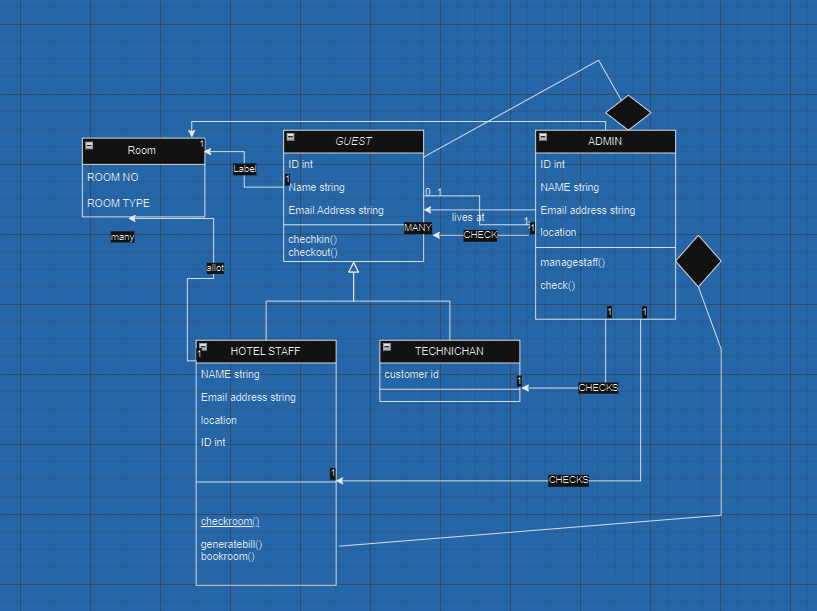


**DOMAIN MODEL DIAGRAM**

A computer screen shot of a computer

Description automatically generated

**CLASS DIAGRAM**

****

**2.1 Justification for Client-Server Architecture:**

**Separation of Concerns**: This architecture separates the client and server components, allowing them to focus on their specific responsibilities. The client handles the user interface and user interaction, while the server manages the application logic, data storage, and processing.

**Scalability:** The client-server model supports scalability by allowing multiple clients to connect to a centralized server. This enables easy distribution of load and resources, making it suitable for systems with a potentially large number of users.

**Centralized Data Management:** In scenarios where data management is a critical aspect (as in a Hotel Reservation System), having a centralized server ensures consistent and secure handling of data. This is particularly important for features like managing user accounts, appointments, and room availability.

**Security:** The server can implement security measures to control access, authentication, and authorization. This is essential for protecting sensitive information in a hotel reservation system.

**Flexibility:** This architecture allows different types of clients (e.g., web browsers, mobile apps) to interact with the server. This flexibility is beneficial for accommodating various user preferences and device capabilities.

2.2 **Modeling according to Client-Server Architecture:**

In a Client-Server Architecture for the Hotel Reservation System, you would have:

**Client Side:**

**User Interfaces:**

User interfaces for Guests (web portal or mobile app).

User interfaces for Hotel Staff (staff portal).

**Client Application Logic:**

Handling user interactions, form validation, and sending requests to the server.

Displaying information retrieved from the server.

**Server Side:**

**Application Logic:**

Managing user authentication and authorization.

Handling appointment scheduling, user management, and notifications.

Processing requests from clients and performing necessary operations.

**Data Management:**

Database for storing user accounts, appointment details, room availability, etc.

Handling queries and updates to the database.

**Communication Layer:**

**Protocols:**

HTTP/HTTPS for communication between clients and server (for web-based applications).

Custom protocols or APIs for mobile applications.

**Request Handling:**

Processing incoming requests, executing necessary actions, and sending responses back to clients.

**Security Layer:**

**Authentication and Authorization:**

Implementing mechanisms for user login, session management, and access control.

Ensuring that only authenticated and authorized users can perform specific actions.

**Notification Services:**

Handling notifications to Guests or Hotel Staff for events like appointment updates or system announcements.