

Experiment 3

Roll No

2113208: Raghav Rathi

2213196: Mohit Gangwani

2213194: Yash Gurnani

Aim: Develop SRS document in IEEE format for the project

What is SRS?

The production of the requirements stage of the software development process is Software Requirements Specifications (SRS) (also called a requirements document). This report lays a foundation for software engineering activities and is constructed when entire requirements are elicited and analyzed. SRS is a formal report, which acts as a representation of software that enables the customers to review whether it (SRS) is according to their requirements. Also, it comprises user requirements for a system as well as detailed specifications of the system requirements.

The SRS is a specification for a specific software product, program, or set of applications that perform particular functions in a specific environment. It serves several goals depending on who is writing it. First, the SRS could be written by the client of a system. Second, the SRS could be written by a developer of the system. The two methods create entirely various situations and establish different purposes for the document altogether. The first case, SRS, is used to define the needs and expectations of the users. The second case, SRS, is written for various purposes and serves as a contract document between customer and developer.

Some Characteristics of a good SRS include:

- **Correctness:** SRS should accurately capture all expected system needs, validated by user review.
- **Completeness:** SRS must include essential requirements, input data responses, references, and terms.
- **Consistency:** Ensure no conflicts in requirements regarding object characteristics, actions, or terminology.
- **Unambiguousness:** Every requirement should have a single interpretation, avoiding multiple definitions.
- **Ranking for importance and stability:** Identify the significance and stability of each requirement.
- **Modifiability:** SRS should allow easy system changes and be well-indexed for modifications.
- **Verifiability:** Requirements should be verifiable through cost-effective system checks.
- **Traceability:** SRS should clearly trace each requirement's origin and use forward/backward traceability.
- **Design Independence:** Avoid implementation details, allowing flexibility in selecting design alternatives.
- **Testability:** Ensure the SRS facilitates the generation of test cases and plans.
- **Understandable by the customer:** Use simple language and avoid complex notations for non-technical users.

- The right level of abstraction: Adjust the level of detail in the SRS based on its purpose and objectives.



Fig 3.1 Characteristics of a good SRS

Benefits of SRS:

Establish the basis of agreement between the customers and the suppliers on what the software product is to do.

It reduces the development effort

It also provides the basis for estimating costs and schedules

It provides a baseline for validation and verification

It is also useable during the maintenance phase

It works like a supplement for the project documentation

It helps building up a trustworthy relationship with the end-consumers of the project

Format of SRS:

Software Requirements Specification

for

<Project>

Version 1.0 approved

Prepared by <author>

<organization>

<date created>

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose.....	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Product Scope.....	1
1.5 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions.....	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints.....	2
2.6 User Documentation.....	2
2.7 Assumptions and Dependencies	3
3. External Interface Requirements.....	3
3.1 User Interfaces.....	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces.....	3
3.4 Communications Interfaces	3
4. System Features.....	4
4.1 System Feature 1	4
4.2 System Feature 2 (and so on)	4
5. Other Nonfunctional Requirements	4
5.1 Performance Requirements	4
5.2 Safety Requirements.....	5
5.3 Security Requirements	5
5.4 Software Quality Attributes.....	5
5.5 Business Rules	5
6. Other Requirements.....	5
Appendix A: Glossary	5
Appendix B: Analysis Models	5
Appendix C: To Be Determined List	6

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4 Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

4.2 System Feature 2 (and so on)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

Software Requirements Specification

for

Hotel Management System

Version 1.0 approved

Prepared by Group Raghav, Mohit, Yash

Thadomal Shahani Engineering College

30 August 2023

Table of Contents

Table of Contents	ii
Revision History	ii
7. Introduction	1
7.1 Purpose.....	1
7.2 Document Conventions	1
7.3 Intended Audience and Reading Suggestions.....	1
7.4 Product Scope.....	1
7.5 References	1
8. Overall Description	2
8.1 Product Perspective	2
8.2 Product Functions.....	2
8.3 User Classes and Characteristics	2
8.4 Operating Environment	2
8.5 Design and Implementation Constraints.....	2
8.6 User Documentation.....	2
8.7 Assumptions and Dependencies	3
9. External Interface Requirements.....	4
9.1 User Interfaces.....	5
9.2 Hardware Interfaces	6
9.3 Software Interfaces.....	7
9.4 Communications Interfaces	8
10. System Features	8
10.1 System Feature 1	9
10.2 System Feature 2 (and so on)	10
11. Other Nonfunctional Requirements	11
11.1 Performance Requirements	11
11.2 Safety Requirements.....	11
11.3 Security Requirements	11
11.4 Software Quality Attributes.....	11
11.5 Business Rules	11
12. Other Requirements.....	12
Appendix A: Glossary	12
Appendix B: Analysis Models	12
Appendix C: To Be Determined List	12

Revision History

Name	Date	Reason For Changes	Version

SRS for Hotel management software

1. Introduction

1.1 Purpose

The Hotel Management System will be a comprehensive software application (version 1.0) designed to manage various aspects of a hotel's operations, including reservations, guest services, room management, and more. This system will automate and streamline hotel management tasks, providing an efficient way to handle guest interactions, room bookings, billing, and staff management.

1.2 Document Conventions

Acronyms and Abbreviations: Acronyms and abbreviations are spelled out on their first occurrence in the document and are followed by their respective acronyms/abbreviations in parentheses, e.g., "Software Requirements Specification (SRS)."

Formatting: Requirement statements are formatted using bullet points for clarity and easy reference. Key terms, such as requirements types (e.g., functional, performance) or specific requirements (e.g., security, usability), may be highlighted or italicized for emphasis.

1.3 Intended Audience and Reading Suggestions

This project is designed for hotel owners, managers, staff, and IT professionals involved in hotel management. This document serves as a comprehensive guide to all the requirements of the Hotel Management System.

1.4 Product Scope

The main purpose of this project is to simplify and optimize hotel management processes, reducing manual efforts and enhancing guest experiences. The Hotel Management System will provide tools to manage reservations, room assignments, guest services, billing, and staff operations efficiently.

1.5 References

<https://www.cloudbeds.com/articles/hotel-management-software-guide/>

<https://bosctechlabs.com/complete-hotel-management-system-development-guide/>

2. Overall Description

2.1 Product Perspective

The Hotel Management System is a standalone application that does not require additional software or third-party plugins. It interacts with databases to manage guest and reservation information.

2.2 Product Functions

The major functions of the Hotel Management System include:

2.2.1 Reservation functions

- allow for a hierarchical organization of functions, providing a structured approach to documenting various aspects of the software.
- It's essential to define the frequency of use, technical expertise, privilege levels, and responsibilities associated with each user class or group (in this case, students). This information helps in tailoring the user experience.
- Describing user characteristics (e.g., educational level) assists in designing a user-friendly interface that aligns with the users' capabilities and expectations.

2.2.2 Guest Services Functions

- Provides a clear, structured breakdown of functions but for a different user class.
- Staff functions, as outlined in this subsection, may include administrative actions, data management, and responsibilities unique to staff roles.
- Ensuring that the software addresses the needs of staff members effectively and efficiently, supporting their roles within the organization.
- It is vital for developers and designers to design user interfaces and functionalities that align with the tasks staff members need to accomplish.

2.3 User Classes and Characteristics

The Hotel Management System will be used by two primary types of users:

Guests:

- Frequency of use: Occasional when making reservations or accessing guest services.
- Technical Expertise: Low
- Responsibilities: Make reservations, check-in, request services, view bills, and check-out.

Hotel Staff:

- Frequency of use: Regularly for managing reservations, room assignments, and guest services.
- Technical Expertise: Moderate to high
- Responsibilities: Manage room bookings, assign rooms, handle check-in/check-out, provide guest services, and manage billing.

2.4 Operating Environment

Hardware platform:

- Processor: Multi-core processor with sufficient processing power to handle concurrent user requests and data processing.
- Memory (RAM): A minimum of 4 GB of RAM is recommended to ensure smooth operation.
- Storage: Adequate storage capacity for database storage.
- Network Connectivity: Reliable internet connection.

OS:

- Windows: Compatible with Windows Server 2016 and later versions for server deployment. Supports client-side usage on Windows 10 and later versions.
- Linux: Compatible with various Linux distributions, including Ubuntu 18.04 LTS and CentOS 7 and later versions.
- macOS: Supports macOS 10.15 (Catalina) and later versions for client-side usage.

Software components:

- Web Browser: Users will access the Hotel Management System through web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari.

2.5 Design and Implementation Constraints

Hardware Limitations:

The system's performance may be affected by the hardware it runs on. Hardware should meet or exceed recommended specifications for optimal performance.

Data Security: Robust security measures, including encryption and access controls, must be implemented to protect sensitive guest and financial data.

Scalability: The system should be designed to accommodate future growth and additional features.

User Documentation:

- User manuals will provide guidance for using the Hotel Management System, catering to both staff and guests.
- Comprehensive documentation will cover system processes and development details.
- Online help and tutorials will assist users in navigating complex features.

2.7 Assumptions and Dependencies

Assumptions:

- Guest and reservation data will be available for migration into the system.

- Hardware and infrastructure meet minimum requirements.
- Third-party services, if used, will be available and reliable.

Dependencies:

- Database Management System (DBMS): The system relies on a DBMS for data storage and retrieval.

- Third-Party Services:

Integration with third-party services (e.g., payment gateways) may be required for specific functionalities.

3. External Interface Requirements

3.1 User Interfaces

Guest Interface

Description: The guest interface allows users to make reservations, access guest services, view bills, and check-out.

Characteristics:

- Authentication: Guests may log in using their reservation details.
- Reservation Management: Features for making reservations, modifying bookings, and viewing reservation details.
- Guest Services: Options for requesting room service, housekeeping, and other guest services.
- Billing: Access to view bills and make payments.
- Check-out: Features for checking out of the hotel.

Hotel Staff Interface

Description: The staff interface is designed for hotel staff responsible for managing reservations, room assignments, guest services, and billing.

Characteristics:

- Authentication: Staff members log in using their credentials.
- Reservation Management: Tools for managing room reservations, room assignments, and availability.
- Check-in/Check-out: Features for processing guest check-in and check-out.

- Guest Services: Options for handling guest requests and services.
- Billing: Tools for generating bills, processing payments, and managing accounts.

3.2 Hardware Interfaces

1. Server Hardware Interfaces

Description: The Hotel Management System runs on server hardware and interfaces with underlying server components.

Characteristics:

- Server Type: The system is designed to run on standard server hardware, including physical servers and virtual machines (VMs).
- Operating System: Compatible with various server operating systems, including Windows Server, Linux distributions, and macOS Server.
- Database Server: Interacts with a database server (e.g., MySQL, PostgreSQL, SQL Server) for data storage and retrieval.
- Communication Protocols: Uses standard protocols such as HTTP/HTTPS for web-based interactions and SQL for database operations.
- Load Balancing: In cases of high user loads, interfaces with load balancers to distribute incoming traffic for scalability.

2. Client Hardware Interfaces

Description: The Hotel Management System provides user interfaces accessible on various client devices.

Characteristics:

- Device Types: Supports desktop computers, laptops, tablets, and smartphones.
- Operating Systems: Accessible on client operating systems, including Windows, macOS, Linux, Android, and iOS.
- Web Browsers: Users access the system through modern web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
- Mobile App Interfaces: May provide mobile applications for Android and iOS devices that interact with the system through APIs.

3. Network Interfaces

Description: The Hotel Management System relies on network interfaces for data transmission and communication.

Characteristics:

- Network Protocols: Uses standard network protocols such as TCP/IP, HTTP, and HTTPS for data transmission between clients and servers.
- Firewalls: Interfaces with network firewalls and security systems to ensure secure data exchange over the network.

- Load Balancers: In load-balanced environments, communicates with load balancers to distribute requests to backend servers.

4. Database Interfaces

Description: The Hotel Management System interfaces with a database management system (DBMS) for data storage and retrieval.

Characteristics:

- DBMS Compatibility: Compatible with multiple DBMS options, including MySQL, PostgreSQL, and Microsoft SQL Server.
- Database Connection: Establishes database connections using standard database communication protocols (e.g., JDBC).
- Data Retrieval: Retrieves data from the database using SQL queries and data manipulation commands.
- Data Storage: Inserts, updates, and manages data in the database using SQL transactions.

3.3 Software Interfaces

1. Database Management System (DBMS)

Description: The Hotel Management System interacts with a database management system (DBMS) for data storage and retrieval.

Characteristics:

- DBMS Options: Compatible with multiple DBMS options, including MySQL, PostgreSQL, and Microsoft SQL Server.
- Data Access: Uses standard database communication protocols (e.g., JDBC) to establish connections and interact with the database.
- Data Retrieval: Retrieves data from the database using SQL queries and data manipulation commands.
- Data Storage: Inserts, updates, and manages data in the database using SQL transactions.
- Shared Data: Data shared with the DBMS includes guest profiles, reservations, room assignments, billing information, and more.

2. Operating Systems

Description: The Hotel Management System is compatible with various operating systems on both server and client sides.

Characteristics:

- Server OS: Compatible with server operating systems, including Windows Server, Linux distributions, and macOS Server.
- Client OS: Supports client operating systems, including Windows, macOS, Linux, Android, and iOS.
- Compatibility: Ensures compatibility with the latest versions of these operating systems to support a wide range of users.

3. Web Browsers

Description: Users access the Hotel Management System through web browsers.

Characteristics:

- Supported Browsers: Compatible with popular web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
- Cross-Browser Compatibility: Ensures that web-based interfaces function consistently across different browsers and versions.
- Data Exchange: Data is transmitted between the system and users' browsers using HTTP/HTTPS protocols.

4. Third-Party Services

Description: The Hotel Management System may integrate with third-party services and APIs for specific functionalities.

Characteristics:

- Communication: Communicates with third-party services using APIs and data exchange protocols.
- Examples: Third-party services may include payment gateways for financial transactions, messaging services for communication, and external booking platforms.
- Data Flow: Data exchanged with third-party services includes payment data, communication logs, and booking information.

3.4 Communication Interfaces

1. Web-Based User Interfaces

Description: The Hotel Management System provides web-based user interfaces accessible through standard web browsers.

Requirements:

- Supported Browsers: Compatible with modern web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
- HTTP/HTTPS: Uses HTTP/HTTPS for secure data communication between clients and the system.
- Data Transfer Rates: Web interfaces are designed to ensure efficient data transfer rates for a responsive user experience.
- Message Formatting: Data exchanged through web interfaces is formatted in standard HTML, JavaScript, and CSS for rendering in browsers.
- Synchronization: Ensures data synchronization between the server and clients to provide real-time updates and notifications.

2. Email Notifications

Description: The Hotel Management System sends email notifications to users for various events and alerts.

Requirements:

- SMTP Protocol: Uses the Simple Mail Transfer Protocol (SMTP) for sending email notifications.
- Message Formatting: Email notifications are formatted in plain text and HTML for user-friendly content.
- Security: Email communication includes standard security measures like secure socket layer (SSL) or transport layer security (TLS) for encryption.
- Data Transfer Rates: Email notifications are sent promptly to ensure timely communication with users.

3. API Interfaces

Description: The Hotel Management System exposes APIs for integration with external systems and mobile applications.

Requirements:

- API Endpoints: Defines specific API endpoints for data retrieval, updates, and interactions.
- Data Formats: APIs accept and provide data in common formats such as JSON or XML for ease of integration.
- Authentication: API endpoints require authentication, ensuring secure access to sensitive data.
- Data Transfer Rates: APIs support efficient data transfer rates to minimize latency in data exchange.
- Security: API communication includes security measures such as token-based authentication and encryption to protect data in transit.

4. System Features

4.1 Reservation Management

4.1.1 Description and Priority

Description: This feature allows guests to make reservations for rooms in the hotel.

Priority: Highest

4.1.2 Stimulus/Response Sequences

a: Guest visits the hotel website and selects "Make a Reservation."

b: Guest provides reservation details, including check-in/check-out dates and room preferences.

- c: The system displays available rooms and their prices.
- d: Guest confirms the reservation and provides payment details if required.
- e: The system processes the reservation and sends a confirmation to the guest.

4.1.3 Functional Requirements

- a: Provide an intuitive reservation interface for guests.
- b: Check room availability based on specified dates and room preferences.
- c: Calculate reservation costs and taxes.
- d: Securely collect and process payment information.
- e: Send confirmation emails to guests upon successful reservations.

4.2 Guest Services

4.2.1 Description and Priority

Description: This feature allows guests to request additional services during their stay, such as room service or housekeeping.

Priority: High

4.2.2 Stimulus/Response Sequences

- a: Guest logs in to their account and selects "Request Services."
- b: The guest chooses the desired service(s) and provides specific instructions if needed.
- c: The system forwards the service request to the appropriate staff or department.
- d: Staff acknowledges the request and provides the requested service.
- e: The system updates the guest on the service status.

4.2.3 Functional Requirements

- a: Offer a user-friendly platform for guests to request services.
- b: Categorize services (e.g., room service, housekeeping) for easy selection.
- c: Notify relevant staff or departments of service requests.
- d: Track the status of service requests and provide updates to guests.
- e: Maintain a record of requested services for billing purposes.

4.3 Billing and Payment

4.3.1 Description and Priority

Description: This feature manages billing and payment processes for guest stays, including check-out and invoicing.

Priority: High

4.3.2 Stimulus/Response Sequences

- a: Guest selects "Check Out" or "View Bill" from their account.

- b: The system calculates the total bill based on the guest's stay, additional services, and any outstanding charges.
- c: Guest reviews the bill and selects a payment method.
- d: The system processes the payment and generates an invoice.
- e: The system sends the invoice to the guest via email or provides a printed copy.

4.3.3 Functional Requirements

- a: Calculate the total bill accurately, including room charges and additional services.
- b: Display the bill to guests for review and transparency.
- c: Support multiple payment methods, including credit cards, cash, and mobile payments.
- d: Generate invoices with detailed breakdowns of charges.
- e: Email or provide printable invoices to guests for their records.

4.4 Room Management

4.4.1 Description and Priority

Description: This feature oversees room assignments, availability, and maintenance.

Priority: Medium

4.4.2 Stimulus/Response Sequences

- a: Staff logs in to the hotel management system and selects "Room Management."
- b: Staff views the room availability calendar and assigns rooms to guests.
- c: The system updates room availability in real-time.
- d: Housekeeping staff mark rooms as cleaned and ready for check-in.
- e: The system alerts staff to maintenance requests or issues with rooms.

4.4.3 Functional Requirements

- a: Maintain a real-time room availability calendar.
- b: Assign rooms to guests based on their reservations.
- c: Notify housekeeping staff of rooms that need cleaning.
- d: Track room maintenance requests and prioritize them.
- e: Provide status updates on room availability, cleaning, and maintenance.

4.5 Staff Management

4.5.1 Description and Priority

Description: This feature manages records of hotel staff members, including personal information and roles.

Priority: Medium

4.5.2 Stimulus/Response Sequences

- a: Admin logs in to the hotel management system and selects "Staff Management."
- b: Admin adds new staff members or updates existing staff information.

c: The system records and displays staff details accurately.

4.5.3 Functional Requirements

- a: Maintain individual profiles for staff members with personal information.
- b: Allow addition, editing, and deletion of staff records.
- c: Assign and manage staff roles and responsibilities.
- d: Ensure data security and restricted access based on roles.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- System Responsiveness: The system must respond to user actions within 3 seconds under normal load conditions.
- Concurrent Users: The system must support a minimum of 100 concurrent users during peak hours without performance degradation.

5.2 Safety Requirements

- Data Security: Implement encryption and access controls for data protection, complying with relevant regulations.
- Regular Backups: Perform daily automated off-site data backups for disaster recovery.
- User Authentication: Enforce strong, unique passwords and implement account lockout mechanisms.

5.3 Security Requirements

- User Authentication: Implement user authentication mechanisms, including strong password policies and account lockout features.
- Data Encryption: Encrypt sensitive data both in transit and at rest to protect against unauthorized access.
- Access Control: Enforce role-based access control (RBAC) to limit user privileges based on their roles.
- Vulnerability Assessment: Conduct regular vulnerability assessments and penetration testing to identify and mitigate security weaknesses.
- Compliance: Ensure compliance with relevant data protection regulations and industry security standards.
- Incident Response: Develop and document an incident response plan to address security breaches and data breaches promptly.

5.4 Software Quality Attributes

- Usability: The system should have a user-friendly interface with a usability score of at least 85 out of 100, as measured by user testing.
- Reliability: The system must have an uptime of 99.5%, ensuring it's available for use except during scheduled maintenance.

- Maintainability: Code changes and updates should be easily accomplished, with at least 80% code coverage for unit tests.
- Portability: The system should be accessible through major web browsers (Chrome, Firefox, Safari) and mobile devices (iOS and Android).

5.5 Business Rules

- User Roles: Only authorized administrators can access and modify staff records, while guests can access their own reservations and billing information.
- Payment Deadlines: Guests must settle their bills upon check-out or as per the hotel's policy to avoid additional charges.
- Access Control: Only authorized staff members can access sensitive financial data and guest profiles.
- Data Retention: Guest records are retained for legal and operational purposes as required by hotel management.
- Maintenance Requests: Maintenance requests from guests should be addressed promptly to ensure a comfortable stay.

6. Other Requirements

6.1 Database Requirements

- The system shall use a relational database management system (RDBMS) to store and manage data.
- Data backups shall be performed daily and stored securely.

6.2 Legal Requirements

- The system shall comply with all applicable data protection and privacy regulations (e.g., GDPR).
- Terms of service and privacy policy pages shall be accessible and up-to-date.

Appendix A: Glossary

SRS: Software Requirements Specification - A document that outlines the detailed requirements for a software project.

DBMS: Database Management System - Software for creating and managing databases.

JDBC: Java Database Connectivity - A Java-based API for connecting and interacting with databases.

SMTP: Simple Mail Transfer Protocol - A standard protocol for sending email messages.

API: Application Programming Interface - A set of rules and protocols for building and interacting with software applications.

RBAC: Role-Based Access Control - A security approach that restricts system access based on user roles and responsibilities.

GDPR: General Data Protection Regulation - A European Union regulation governing data protection and privacy for individuals.