Experiment 3

Roll No

2213194: Yash Gurnani
2213196: Mohit Gangwani
2113208: Raghav Rathi

Software Requirement Specification (SRS)

A Software Requirement Specification (SRS) is a comprehensive document that outlines a software system's detailed functional and non-functional requirements. It serves as a contract between the client/customer and the development team, providing a clear and unambiguous description of what the software should do, its behavior, constraints, and quality attributes. The SRS document typically includes information about system features, user interactions, data handling, performance criteria, and more.

Benefits of SRS:

Clarity and Understanding: The SRS fosters clarity by providing an unambiguous description of the software's objectives, reducing misunderstandings between stakeholders. This clarity is instrumental in ensuring that everyone involved has a shared vision.

Basis for Development: As the cornerstone of the development process, the SRS acts as a compass, guiding developers in constructing the software according to the specified requirements. It helps maintain focus and direction throughout the project.

Change Control: Change is inevitable in software development. The SRS serves as a baseline for assessing proposed alterations, allowing project managers and stakeholders to evaluate the impact of changes and make informed decisions.

Testing and Validation: Testing teams rely on the SRS as a reference point to validate that the software aligns with the specified requirements. This alignment is crucial for ensuring the software functions as intended.

Project Management: Project managers use the SRS to plan resources effectively, estimate costs, and establish realistic timelines. It provides a roadmap for successful project execution.

Legal and Contractual Use: In contractual agreements, the SRS serves as a legally binding document, ensuring that the software aligns with the agreed-upon specifications. It offers protection to both clients and developers.

Communication: Effective communication is at the heart of successful software development. The SRS facilitates clear and structured communication among stakeholders, including clients, developers, testers, and project managers. It establishes a common language for discussing requirements and progress.

In conclusion, the Software Requirement Specification is a foundational document that not only defines the blueprint of the software but also provides a structured framework for collaboration, risk management, and quality assurance throughout the development lifecycle. Its significance cannot be overstated in ensuring the successful delivery of software that meets user expectations and industry standards.

IEEE-style Software Requirement Specification (SRS):

Purpose: Explain why the SRS is being created. Typically, it outlines the objectives, scope, and goals of the software project.

Scope: Clearly define what the software will and will not do. This helps in managing expectations and preventing scope creep.

References: List any documents, standards, or external references that are relevant to the SRS. This could include industry standards or previous project documents.
Overall Description:

Product Perspective: Describe how the software fits into the larger system or environment. Explain any interactions with other systems or components.

Product Functions: Enumerate and provide detailed descriptions of the main functions and features the software will provide.

User Characteristics: Describe the expected users of the software, including their skills, knowledge, and any special requirements.

Constraints: Outline any limitations or constraints that may affect the development or operation of the software.

Assumptions and Dependencies: Document any assumptions made during the SRS creation and dependencies on external factors or components.
Specific Requirements:

Functional Requirements: Provide detailed descriptions of what the software should do. Use use cases, flowcharts, or other diagrams to illustrate the behavior.
Non-Functional Requirements: Specify quality attributes such as performance, security, reliability, and usability. Include measurable criteria for these attributes.

External Interfaces: Describe how the software interacts with external systems, databases, hardware, or software components.
User Interfaces: Detail the design and layout of the user interfaces, including wireframes or mockups if available.

System Features: List and describe specific features of the software in a structured manner.
Performance Requirements: Define response times, throughput, and resource utilization targets for the software under various conditions.

Quality Attributes: Specify the reliability, availability, maintainability, and other quality characteristics expected from the software.

Constraints: Reiterate any constraints mentioned in the introduction and provide additional context if necessary.
Appendices:

Diagrams: Include any diagrams, such as flowcharts, data models, or system architecture diagrams that help clarify the requirements.

Use Cases: If applicable, provide detailed use case scenarios to illustrate how users will interact with the system.

Supplementary Documentation: Attach any supplementary documents or references that are relevant to understanding the requirements.
In practice, it's crucial to tailor the SRS to the specific project's needs. Some projects may require more detailed sections, while others may omit certain sections that are not applicable. The goal of the SRS is to provide a clear and unambiguous description of the software requirements to ensure everyone involved understands what is expected from the project. Additionally, maintaining version control and tracking changes to the SRS throughout the project's lifecycle is essential to managing evolving requirements and scope changes.

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.google.com/
https://www.google.com/
https://www.google.com/

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.