PES UNIVERSITY

Department of Computer Science and Engineering

UE21CS341A: Software Engineering

Software Requirements Specification

for

Chat Application

Version 1.0 approved

Prepared by

1.	Meghana Anand	PES2UG21CS291
2.	L Sai Tejas	PES2UG21CS250
3.	Meenal Bagare	PES2UG21CS289
4.	Melvin Jojee Joseph	PES2UG21CS294

PES University, Bangalore

Department of Computer Science and Engineering

22/09/2023

PES UNIVERSITY

Department of Computer Science and Engineering

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

PES UNIVERSITY

Department of Computer Science and Engineering

Introduction

Purpose

The purpose of this SRS document is to define the requirements and specifications for the development of a real time chat application. The document will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate. It serves as a comprehensive guide for the development team, stakeholders, and any external parties involved in the creation of the chat application.

Intended Audience

The chat application caters to a diverse user base, including individuals, business professionals, students, social groups, and event organizers. It encompasses a wide range of stakeholders, from developers and project managers to marketing staff, testers, and documentation writers. This versatile platform offers real-time communication for personal, academic, and professional purposes, making it accessible and valuable to a broad audience with varied communication needs.

Product Scope

The chat application's scope encompasses features such as real-time messaging, multimedia sharing, and a user-friendly interface. These objectives align with corporate goals of improving customer engagement, expanding market reach, and enhancing operational efficiency, contributing to the overall success of the business strategy.

References

PES UNIVERSITY

Department of Computer Science and Engineering

Overall Description

Product Perspective

The chat application is a standalone product designed to provide seamless communication between users. The application enables users to exchange direct messages, send various types of documents, and share media such as photos and videos. It enhances the user experience by offering a wide range of communication options within a user-friendly interface.

Product Functions

The major functions of the product include

- User Registration and Authentication
- Real time messaging
- Search for new users
- Ability to create groups with multiple users

User classes and characteristics

- End users- This category represents all the audience of the application. They may vary drastically in their technical expertise so an easy to use interface must be provided.

Operating Environment

- Hardware Platform: The software will operate on standard web server infrastructure which includes a web server and database server. The hardware requirements for these servers will be determined during the system architecture design phase.
- Operating System and Versions: The software will be compatible with commonly used operating systems such as Windows Server, Linux (e.g., Ubuntu), and cloud-based platforms (e.g., AWS, Azure).
- Software Components: The software will coexist with various software components, including web browsers (e.g., Google Chrome, Mozilla Firefox, Safari), web server software (e.g., Apache, Nginx), and database management systems (e.g., Redis, MongoDB, MySQL, PostgreSQL).

Design and Implementation Constraints

- Hardware Limitations: The software should be designed to work efficiently within the hardware limitations, including timing and memory requirements. Specific hardware requirements will be determined during the system design phase.
- Programming Languages and Technologies: The development team is constrained to use specific programming languages and technologies, such as Golang, React js, and Redis.
- Security Considerations: Stringent security measures must be implemented to protect user data, including secure communication (HTTPS), data encryption, and authentication mechanisms.

- Corporate and Regulatory Policies: The development must adhere to company policies regarding data privacy, security, and ethical data usage. Additionally, compliance with relevant regulations (e.g., GDPR) is mandatory.

Assumptions

• User Assumptions:

- Internet Connectivity: Assume users have a stable internet connection.
- Browser Compatibility: Assume users are accessing the application from modern, updated web browsers. Test for compatibility across different browsers.
- Basic Computer Literacy: Assume users have basic knowledge of how to use a computer and navigate a web application.

• Technical Assumptions:

- Web Technologies: Assume a certain level of familiarity with web technologies like HTML, CSS, and JavaScript.
- JavaScript Enabled: Assume that users have JavaScript enabled in their browsers since most chat applications heavily rely on it for real-time updates.
- Device Capabilities: Assume users have devices (e.g., computer, smartphone, tablet) with sufficient processing power and memory to handle the application.

Dependencies

• Hosting and Deployment: Depending on a reliable hosting platform or a cloud service platform like AWS, Azure, Google Cloud, or Heroku.

• Database Management System: Depending on an external database system (e.g., Redis, PostgreSQL, MongoDB) and ensure that it is properly configured and managed.

PES UNIVERSITY

Department of Computer Science and Engineering

External Interface Requirements

User Interfaces

The user interface for the chat application will consist of the following:

Main Page: This is the landing page of the application, where users can log in or sign up. It will include fields for entering login credentials and a registration form for new users.

Components:

- Chat Conversation Display: Where users can see their chat history.
- Message Input Field: Allows users to type and send messages.
- Attachment Feature: Enables users to send files, photos, videos, and other media.
- Contact List: Displays the list of contacts, including group chats.
- Standard Buttons: Including send message, attach file, and create group chat.

User Profile: The user profile page permits users to manage their personal information and profile picture.

Components:

- User Information: Displays user details such as username, email, and profile picture.
- Profile Picture Upload: Allows users to update their profile picture.
- Editable Fields: Users can modify their username, email, and other profile information.

Group Chat Interface: This interface is utilized when creating or participating in group chats. It provides tools for managing group members and group conversations.

Components:

- Group Member List: Displays the list of group members.
- Group Chat Conversation Display: Shows the chat history within the group.
- Message Input Field: Enables users to send messages within the group.

- Standard Buttons: Including send message and attach file.

Software Interfaces

The software will interface with the following components:

- **Redis Repository:** A simple repository manages all Redis operations, including data storage and retrieval.
- **HTTP Server:** A REST API server to serve user-specific functionalities, such as user registration and login.
- **WebSocket Server:** A real-time server that enables immediate message delivery and updates.
- **Frontend (Client):** A ReactJS application with highly interactive components to provide the user interface.

Communications Interfaces

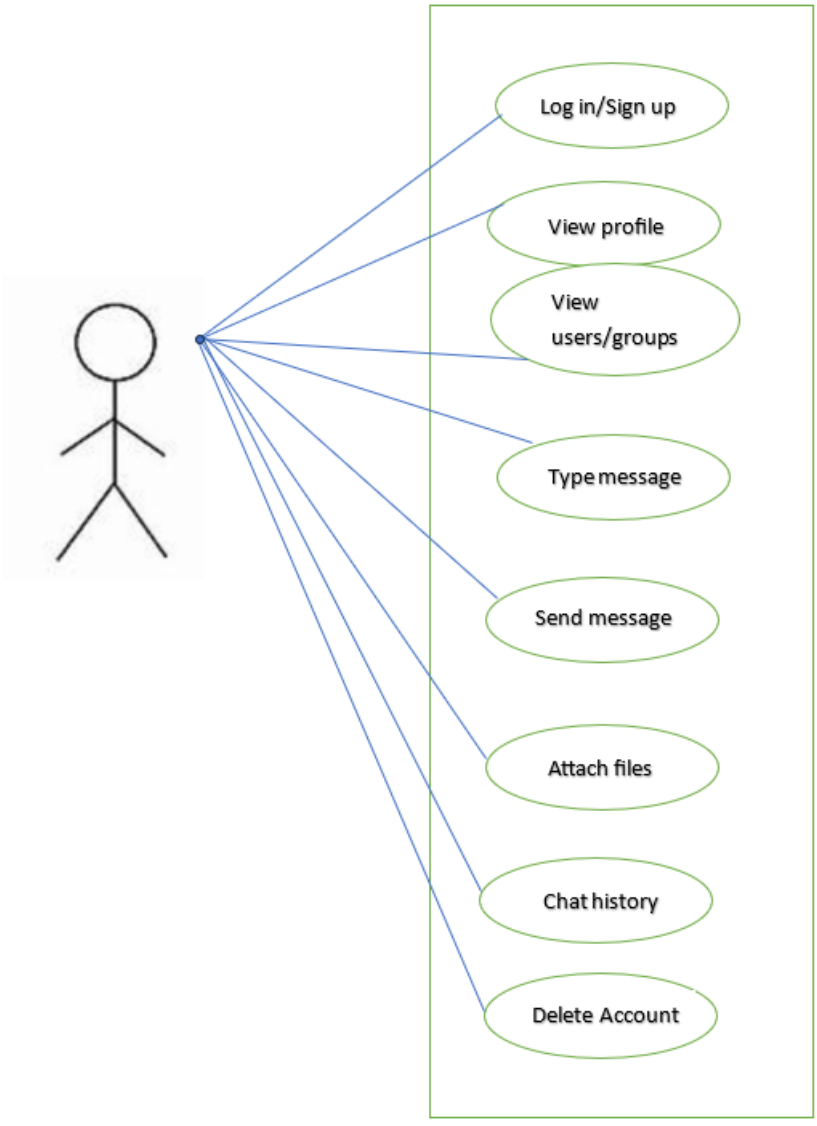
The chat application will use standard communication protocols for real-time messaging:

- **WebSocket Protocol:** The application will use WebSocket for real-time communication between clients and the server.
- **HTTP/HTTPS:** RESTful API calls will be made using HTTP/HTTPS protocols for user registration, login, and other data exchange.

PES UNIVERSITY

Department of Computer Science and Engineering

ANALYSIS MODEL



System Features

1. User Registration and Login

Description and Priority

- Allows users to register new accounts and log in to the application.
- Priority: High

Stimulus/Response Sequences

- User navigates to the registration page.
- User provides necessary information and submits the registration form.
- System validates user data and creates a new account.
- User logs in using the registered credentials.

Functional Requirements

- User registration form with fields for username, email, password, and profile picture.
- Password encryption to ensure data security.
- User authentication and authorization during login.
- User profile creation with the ability to upload a profile picture.

2. Real-Time Messaging

Description and Priority

- Allows users to send instant messages, attachments, and create group chats.
- Priority: High

Stimulus/Response Sequences

- User selects a contact or group to chat with.
- User sends a message or attachment.
- System delivers the message to the recipient in real-time.
- Users receive notifications for new messages.

Functional Requirements

- Real-time chat interface with message input and display.
- Ability to send text messages, photos, videos, and files.
- Notification system to alert users of new messages.
- Group chat creation and management features.

PES UNIVERSITY

Department of Computer Science and Engineering

Other Non-Functional Requirements

Performance Requirements

- **Real-Time Messaging:**
  - **Requirement:** Messages should be delivered in real-time, with a delay of no more than 2 seconds.
  - **Rationale:** Real-time messaging ensures a seamless and interactive communication experience for users.
- **Scalability:**
  - **Requirement:** The application should be able to handle a concurrent user load of at least 10,000 users without significant degradation in performance.
  - **Rationale:** This ensures the application can accommodate a large user base and maintain responsiveness.
- **Latency Tolerance:**
  - **Requirement:** The average round-trip latency for messages (from sender to receiver and back) should be less than 500 milliseconds.
  - **Rationale:** Low latency ensures that users experience a near-instantaneous response when sending or receiving messages.
- **Bandwidth Efficiency:**
  - **Requirement:** The application should optimize data transfer to minimize bandwidth consumption, especially for multimedia content.
  - **Rationale:** This helps in reducing data costs and ensuring a smooth experience for users with limited bandwidth.

Safety Requirements

- **User Reporting Mechanism:**

  - **Requirement:** Provide a feature for users to report abusive behavior, spam, or inappropriate content.
  - **Rationale:** This empowers users to contribute to the safety of the community and take action against unwanted behavior.

- **Data Backup and Recovery:**

  - **Requirement:** Regularly backup chat data and provide a mechanism for data recovery in case of accidental deletion or system failures.
  - **Rationale:** This prevents data loss and ensures continuity of communication.

Security Requirements

- **End-to-End Encryption:**

  - **Requirement:** Implement end-to-end encryption for messages to ensure that only the sender and receiver can read the content.
  - **Rationale:** This protects user privacy and prevents unauthorized access to messages.
- **User Authentication:**
  - **Requirement:** Users must authenticate their identity using strong, secure methods (e.g., password, biometrics) before accessing the application.
  - **Rationale:** This ensures that only authorized users have access to the chat application.
- **Data Privacy Compliance (e.g., GDPR, HIPAA):**
  - **Requirement:** Adhere to applicable data protection regulations and obtain necessary user consent for data processing activities.
  - **Rationale:** This ensures legal compliance and protects user rights regarding their personal information.
- **Security Certifications (e.g., ISO 27001, SOC 2):**
  - **Requirement:** Achieve and maintain relevant security certifications to demonstrate compliance with industry standards.
  - **Rationale:** This provides assurance to users and stakeholders that the application meets recognized security standards.

Software Quality Attributes

- **Usability:**
  - **Requirement:** The application should achieve a user satisfaction score of at least 80% in usability testing.
  - **Rationale:** Prioritize ease of use to ensure that users can navigate and utilize the chat application intuitively.
- **Reliability:**
  - **Requirement:** The application should have an uptime of at least 99 %.
  - **Rationale:** Ensure that the chat service is highly reliable to maintain continuous communication for users.
- **Maintainability:**
  - **Requirement:** Any critical bug reported should be fixed within 24 hours, and non-critical bugs within one week.
  - **Rationale:** Easy maintenance and bug resolution are crucial for providing a seamless user experience.
- **Adaptability:**
  - **Requirement:** The application should be able to accommodate new features or technologies with minimal disruption to existing functionality.

- **Rationale:** Ensure the application can evolve to meet changing user needs and technological advancements.

Business Rules

- **User Roles:**
  - **Rule:** Only registered users can send and receive messages.
  - **Implication:** Implement user authentication to enforce this rule.
- **Message Deletion:**
  - **Rule:** Users can delete their own messages within 10 minutes of sending them.
  - **Implication:** Implement a message deletion feature with a time limit.
- **Reporting and Blocking:**
  - **Rule:** Users can report other users for abusive behavior or content, and have the option to block them.
  - **Implication:** Implement reporting and blocking functionalities.
- **Group Chats:**
  - **Rule:** Users can create and manage group chats, with a maximum of 50 participants per group.
  - **Implication:** Develop features for creating and managing group conversations.

Other Requirements

- **Internationalization Requirements:**
  - **Language Support:** The application should support multiple languages, with English as the default.
  - **Unicode Support:** Ensure that the application handles Unicode characters correctly to accommodate various languages.
- **Reuse Objectives:**

- **Code Modularity:** Design code with modularity in mind to facilitate code reuse and future extensions.
- **Component Reusability:** Identify and document components that can be reused in other projects within the organization.

- **User Documentation:**

  - **User Guides:** Create user documentation including guides, FAQs, and tutorials to help users navigate and utilize the application effectively.

- **Performance Monitoring:**

  - **Monitoring Tools:** Implement performance monitoring tools to track system health, response times, and other relevant metrics.

Appendix A: Glossary

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

HTTP: Hypertext Transfer Protocol - The protocol used for transferring data over the World Wide Web.

HTTPS: Hypertext Transfer Protocol Secure - The secure version of HTTP that encrypts data transmitted between the web server and the client.

SQL: Structured Query Language - A domain-specific language used in programming and managing relational databases

AWS: Amazon Web Servies- on-demand cloud computing platforms

Redis: open-source in-memory storage, used as a distributed, in-memory key–value database, cache and message broker

API: Application Programming Interface - A set of rules and protocols that allows different software applications to communicate with each other

REST API- APIs conforming to the representational state transfer architectural style

GDPR: General Data Protection Regulation - A regulation in EU law on data protection and privacy for all individuals within the European Union and the European Economic Area.

HIPAA- a set of regulatory standards that intend to protect private and sensitive patient data

ISO 27001 - an international standard to manage information security

SOC 2- an auditing procedure that ensures your service providers securely manage your data to protect the interests of your organization and the privacy of its clients

Appendix B: Field Layouts

Fields needed to register a user

Field	Datatype	Description	Is Mandatory
First Name	String	First name of the user	Yes
Last Name	String	Last name of the user	Yes
Email	Alphanumeric	Email id of the user	Yes
DOB	Date	DOB of user	Yes
User name	Alphanumeric	Unique user name	Yes
Profile picture	Image	Image for the profile	No

PES UNIVERSITY

Department of Computer Science and Engineering

Appendix C: Requirement Traceability Matrix

Sl. No	Requirement ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID