Software Requirements Specification

for

Whispr

Version 1.0 approved

Prepared by Dilawar Khan (21L-5254)

Muhammad Amjad (21L-5207)

Subhan Ahmad (21L-5198)

Mashhood Ahsan (21L-5406)

FAST-NUCES Lahore

March 13, 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Initial Document	13-03-2024	-	1.0

1. Introduction

1.1 Purpose

This document specifies the Software Requirements Specification (SRS) for version 1.0 of a real-time chatting website “Whispr”. This SRS focuses on the core functionalities and excludes any future potential features or integrations.

1.2 Document Conventions

- **Bold**: Denotes keywords or terms with specific meaning within this document.
- **Italics**: Highlights references to figures, tables, or sections within the document.
- **Priorities**: All requirements within this document are considered high priority unless explicitly stated otherwise.

1.3 Intended Audience and Reading Suggestions

This document is intended for the following audiences:

- **Developers**: Responsible for implementing the functionalities outlined in this document.
- **Project Managers**: Overseeing the development process and ensuring adherence to these requirements.
- **Testers**: Designing and executing test cases to verify the product meets these requirements.
- **Documentation Writers**: Creating user guides and other documentation based on the functionalities specified here.
- **Evaluators**: Teachers and their assistants to evaluate our project documentation

Reading Suggestions:

- Read this section (Purpose, Document Conventions, Intended Audience) for an introduction.
- Proceed to the "Product Scope" section for a high-level overview of Whispr’s purpose.
- Review the "Functional Requirements" and "Non-Functional Requirements" sections for detailed specifications.
- Refer back to this section for definitions of keywords or terms used throughout the document

1.4 Product Scope

This document outlines the requirements for a real-time chat website designed to provide a secure and user-friendly communication platform. The product aims to address user concerns regarding privacy and data security in existing messaging applications. Key benefits include:

- Secure real-time messaging with strong encryption.
- User customization options for profiles and chat experience.
- Robust search functionality to locate contacts and messages.
- Responsive design for seamless access across various devices.
- The initial focus will be on establishing a core set of functionalities with potential for future expansion based on user needs and market trends.

1.5 References

A list of specific references (user interface style guides, security standards, etc.) will be included here once identified and finalized during the development process.

2. Overall Description

2.1 Product Perspective

This product is a new, self-contained real-time chat website. It is not a replacement for any existing system but aims to offer a secure and user-friendly alternative to current messaging platforms.

2.2 Product Functions

- **User Management:**
    - User registration and login with secure authentication.
    - User profile creation and management.
- **Chat Functionality:**
    - Real-time chat messaging with text, images, and files.
    - One-on-one and group chat capabilities.
    - Search functionality for finding contacts and messages.
- **Security and Privacy:**
    - Implement strong encryption protocols to protect user data and communication.
    - User control over data privacy settings.
- **User Interface:**
    - Provide a user-friendly and intuitive interface for all functionalities.

    - Responsive design for optimal viewing across various devices (desktop, mobile, tablets).
- **System Management:**
    - System monitoring and logging for security and performance analysis.

2.3 User Classes and Characteristics

- **Primary Users:**
    - Individuals seeking a secure and private real-time chat platform.
    - **Technical expertise:** Basic to moderate computer literacy.
    - **Frequency of use:** Regular users who rely on the platform for communication.
- **Secondary Users:**
    - Businesses or organizations seeking a secure communication channel for internal teams.
    - **Technical expertise:** May require additional features or integrations (future expansion).
    - **Frequency of use:** Varies depending on organizational needs.

2.4 Operating Environment

- **Hardware Platform:** Industry-standard computers with internet access.
- **Operating System:** Compatible with major desktop and mobile operating systems (Windows, macOS, Android, iOS) through web browsers.
- **Software:** Modern web browsers with JavaScript enabled.

2.5 Design and Implementation Constraints

- **Technologies:** Utilize the MERN stack (MongoDB, Express.js, React.js, Node.js) for development.
- **Security:** Implement industry-standard encryption protocols (e.g., TLS/SSL) to secure communication and data storage.
- **Scalability**: Design the system to be scalable to accommodate a growing user base.

2.6 User Documentation

- **User Manual:** A comprehensive user guide explaining website functionalities, registration, usage instructions, and troubleshooting steps.
- **Online Help:** Contextual help guides and tutorials integrated within the website interface.

2.7 Assumptions and Dependencies

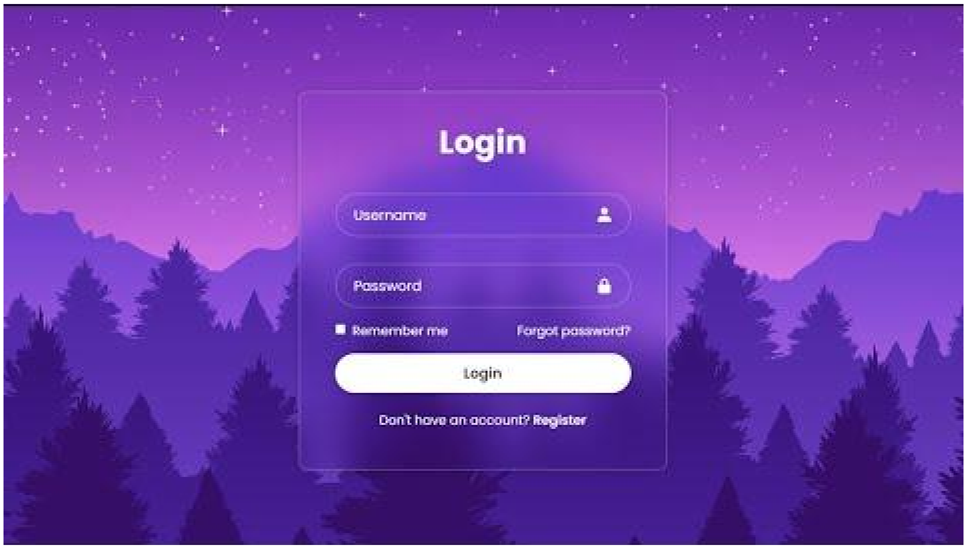
- **Assumptions:** Availability of reliable internet connectivity for users. Users have basic computer literacy skills.
- **Dependencies:** Successful development and integration of chosen technologies (MERN stack). Security of third-party libraries and frameworks used in development.

3. External Interface Requirements

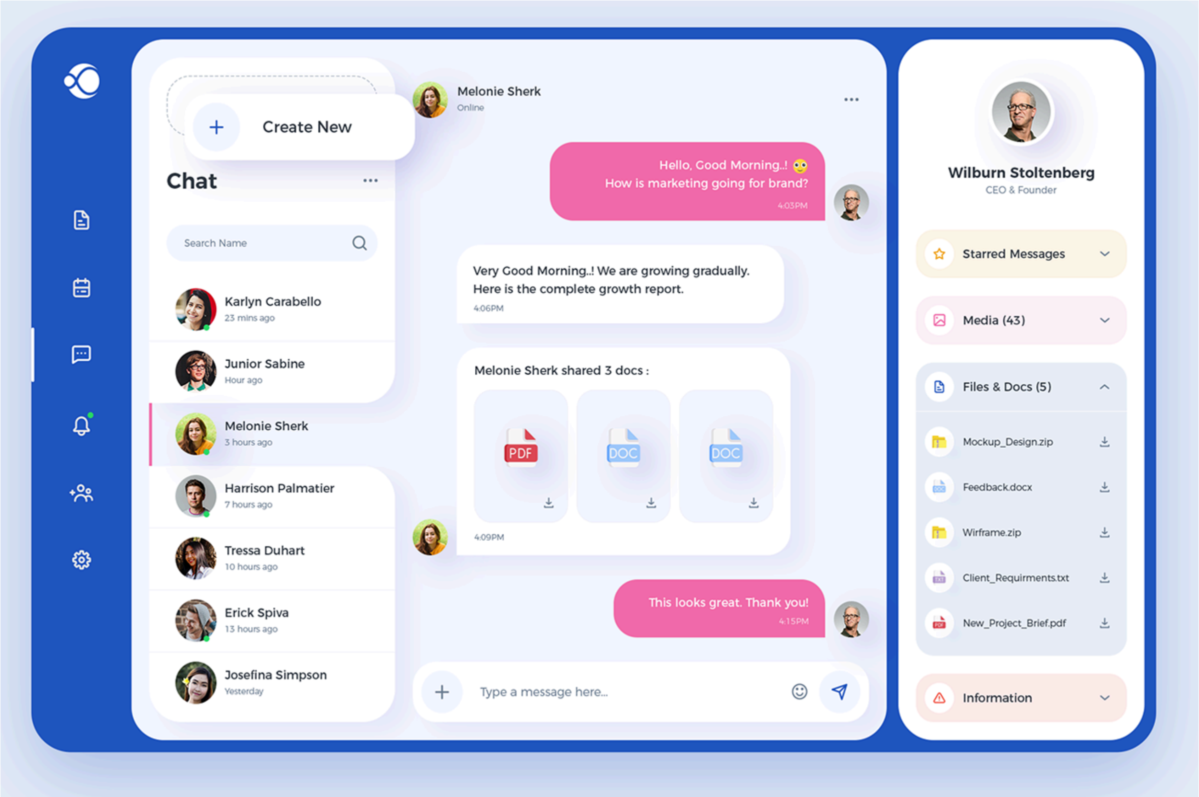
3.1 User Interfaces

While a detailed UI design will be documented separately, this section outlines the key user interfaces for the product:

- **Login/Registration Screen:** User interface for entering credentials for registration or login.
- **User Profile Screen:** Interface for managing user profile information (avatar, username, etc.).
- **Contact List Screen:** Interface for displaying a list of user contacts for initiating chats.
- **Chat Window:** Interface for real-time chat communication with text, image, and file sharing functionalities.
- **Search Bar:** Interface for searching for contacts and messages.
- **Settings Menu:** Interface for managing user settings related to privacy, notifications, and interface customization.



(mock login page)



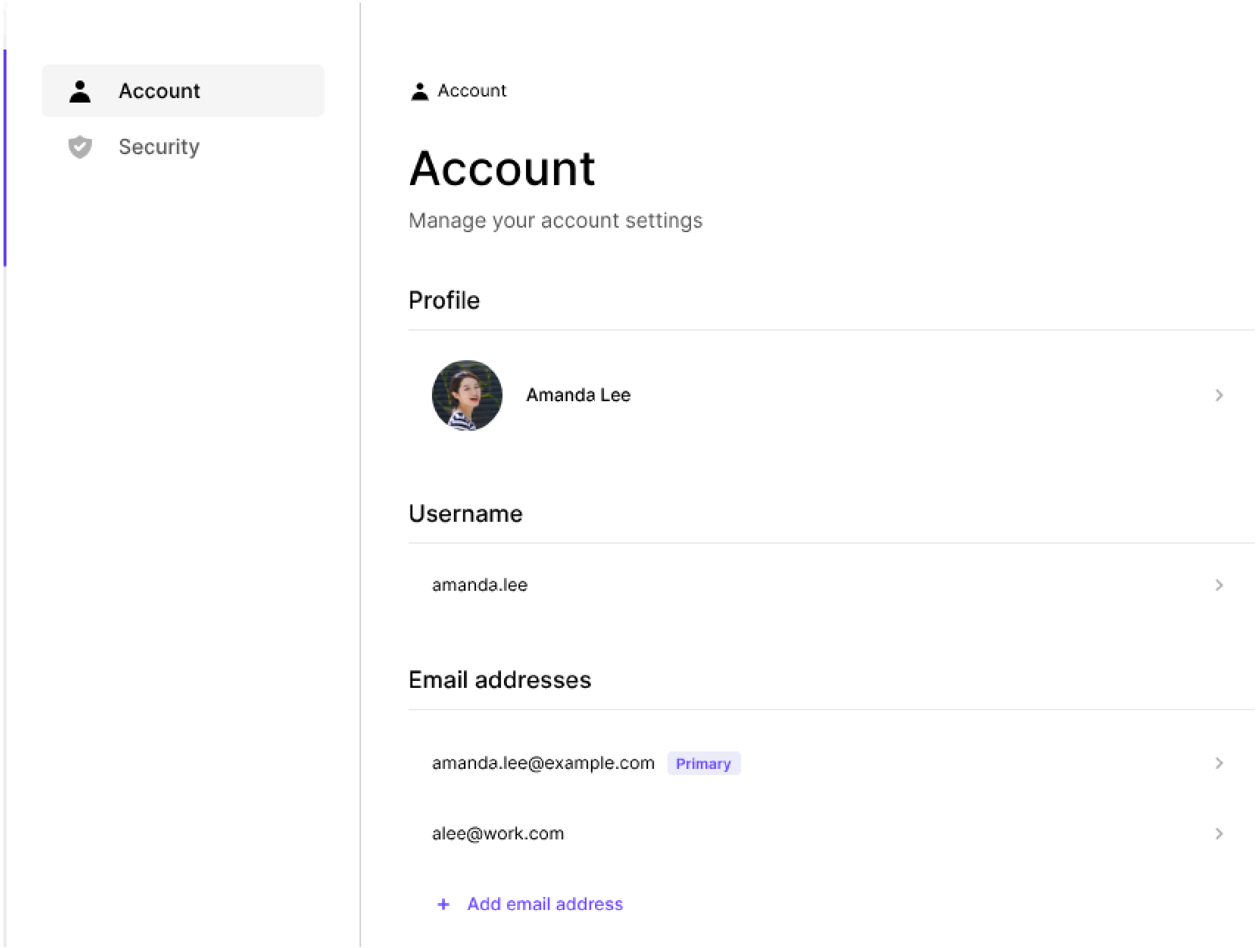
Hello, Good Morning..!
How is marketing going for brand?
4:03PM

Very Good Morning..! We are growing gradually.
Here is the complete growth report.
4:06PM

Melonie Sherk shared 3 docs :
4:09PM

This looks great. Thank you!
4:15PM

(chats screen)



Account

Manage your account settings

Profile

Amanda Lee

Username

amanda.lee

Email addresses

amanda.lee@example.com Primary

alee@work.com

(account settings page)

General UI Considerations:

- Follow modern web design standards for a clean and intuitive user experience.
- Ensure responsive design for optimal viewing across various devices.
- Provide clear labeling and user guidance for all functionalities.
- Implement consistent formatting and visual elements throughout the application.

Error Messages:

- Display user-friendly error messages that clearly explain encountered issues.
- Offer potential solutions or troubleshooting steps for identified errors.

3.2 Hardware Interfaces

The product will primarily interact with the following hardware components:

- **User Devices:** Computers, laptops, tablets, and smartphones with internet access.
- **Web Browsers:** Modern web browsers with JavaScript enabled.

Data Interaction:

- User devices will send and receive data (text messages, images, files) through the web browser interface.
- The product will interact with various hardware components through standard web browser protocols (HTTP/HTTPS).

Communication Protocols:

- The product will rely on standard internet protocols (TCP/IP) for communication between user devices and the server.

3.3 Software Interfaces

- **Database (MongoDB):** Stores user data (profiles, contacts, chat history) and facilitates data retrieval for functionalities.
- **Web Server (Node.js):** Processes user requests, manages real-time chat communication, and interacts with the database and authentication server.
- **Authentication Server:** Handles user registration, login, and session management (potentially using JWT tokens).
- **JavaScript Libraries:** Utilize open-source libraries for functionalities like real-time messaging and UI components.

Data Exchange:

- User data (registration details, profile information, chat messages) will be exchanged between the web client and the database.
- Authentication tokens will be exchanged between the web client and the authentication server.
- Real-time chat messages will be exchanged between connected users through the web server.

Communication Protocols:

- The web client and server will communicate using standard web protocols (HTTP/HTTPS).
- The web server and database will likely utilize a database-specific query language (e.g., MongoDB Query Language).
- Real-time communication protocols (e.g., WebSockets) may be implemented for efficient message exchange.

3.4 Communications Interfaces

- The product will rely on internet connectivity for all functionalities.
- Users will communicate with the product through a web browser interface.
- Real-time chat communication will utilize appropriate protocols (e.g., WebSockets) for efficient message exchange.
- Standard message formatting (e.g., JSON) might be used for data exchange between the web client and server.
- Secure communication protocols (HTTPS) will be implemented for data encryption during transmission.

Communication Standards:

- The product will adhere to standard internet protocols (TCP/IP, HTTP/HTTPS) for communication.
- Specific real-time communication protocols (e.g., WebSockets) might be chosen based on implementation details.

Security Considerations:

- Implement strong encryption protocols (HTTPS) to secure data transmission between user devices and the server.
- Securely store user data within the database using appropriate authentication and access controls.

4. System Features

This section outlines the functional requirements for the product, organized by key features:

4.1 User Management

4.1.1 Description and Priority

This feature allows users to register, log in, and manage their profiles within the application. (Priority: High)

4.1.2 Stimulus/Response Sequences

- User navigates to the registration page.
- User enters registration details (username, password, email, etc.).
- System validates user input and creates a new user account.
- User receives confirmation message and is redirected to the login page.
- User enters login credentials (username/email and password).
- System authenticates user credentials and grants access to the application.
- User navigates to the profile settings page.

- User edits profile information (avatar, bio, etc.).
- System updates user profile data and reflects changes within the application.

4.1.3 Functional Requirements

- REQ-1: The system shall provide a user registration form to capture user details (username, password, email, etc.).
- REQ-2: The system shall validate user input during registration (e.g., username uniqueness, strong password requirements).
- REQ-3: The system shall allow users to log in with registered credentials (username/email and password).
- REQ-4: The system shall implement session management to maintain user login status.
- REQ-5: The system shall provide an interface for users to manage profile information (avatar, bio, etc.).
- REQ-6: The system shall securely store and update user profile data within the database.
- REQ-7: The system shall display user profile information within the application interface.

4.2 Chat Functionality

4.2.1 Description and Priority

This feature enables users to engage in real-time chat conversations with other users on the platform. (Priority: High)

4.2.2 Stimulus/Response Sequences

- User navigates to the chat interface.
- User selects a contact or starts a new chat conversation.
- System displays the chat interface with the selected contact.
- User types and sends a message.
- System sends the message to the recipient.
- Recipient receives the message in real-time.
- Recipient views the message and can respond accordingly.
- System updates the chat interface with the received message.

4.2.3 Functional Requirements

- REQ-1: The system shall provide a chat interface for users to initiate and participate in conversations.
- REQ-2: The system shall allow users to select contacts or search for users to start new chat conversations.
- REQ-3: The system shall support real-time messaging functionality (text, images, files) between users.

- REQ-4: The system shall ensure message delivery within seconds to maintain real-time communication.
- REQ-5: The system shall implement message encryption to ensure user privacy and data security.
- REQ-6: The system shall display message timestamps to indicate the timing of each message.
- REQ-7: The system shall support message deletion functionalities for users.
- REQ-8: The system shall provide notifications for new incoming messages to users.
- REQ-9: The system shall maintain message history for each chat conversation, allowing users to view past messages.
- REQ-10: The system shall optimize chat performance to handle high volumes of concurrent conversations efficiently.

4.3 Authentication

4.1.1 Description and Priority

This feature enables users to securely authenticate themselves to access the real-time chatting website. (Priority: High)

4.1.2 Stimulus/Response Sequences

- User navigates to the login page.
- User enters their username/email and password.
- System validates the user credentials.
- If valid, system grants access to the user's account.
- If invalid, system displays an error message and prompts the user to retry or reset their password.
- User selects the option to reset their password.
- System sends a password reset link to the user's registered email.
- User clicks on the password reset link.
- System verifies the link validity and prompts the user to enter a new password.
- User enters a new password.
- System updates the user's password and confirms the password reset.

4.1.3 Functional Requirements

- REQ-1: The system shall provide a login form for users to enter their credentials (username/email and password).
- REQ-2: The system shall validate user credentials against stored user data (username/email and password).
- REQ-3: The system shall enforce password complexity requirements (e.g., minimum length, combination of alphanumeric characters).

- REQ-4: The system shall provide functionality for users to reset forgotten passwords via email verification.
- REQ-5: The system shall generate and send a unique password reset link to the user's registered email address.
- REQ-6: The system shall verify the validity of password reset links to prevent unauthorized access.
- REQ-7: The system shall allow users to set a new password upon successful verification of the password reset link.
- REQ-8: The system shall implement session management to maintain user authentication status across sessions.
- REQ-9: The system shall provide secure mechanisms for user logout to terminate active sessions.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- **Load Handling:** The system should be able to handle a minimum of hundred concurrent users without significant performance degradation (latency, response times). This will be further defined based on user base projections and scalability testing.
- **Message Delivery**: Real-time chat messages should be delivered between users with minimal latency (target: 100 milliseconds on average).
- **System Uptime:** The system should strive for a minimum uptime of 90%. Downtime for maintenance or updates should be scheduled during non-peak usage hours.
- **Rationale:** These performance requirements ensure a smooth and responsive user experience for real-time communication. They guide development choices towards scalability and efficient message delivery mechanisms.

5.2 Safety Requirements

This real-time chat application does not handle safety-critical systems or directly impact physical safety. However, the following considerations will be implemented:

- **User Safety:** Security measures will be in place to prevent unauthorized access or data leaks that could compromise user privacy.

5.3 Security Requirements

- **Data Security:** User data (credentials, messages) will be encrypted at rest and in transit using industry-standard protocols (e.g., HTTPS, secure hashing algorithms).

- **Authentication and Authorization:** Strong user authentication mechanisms (e.g., password hashing) will be implemented to prevent unauthorized access. The system will enforce access controls based on user roles and permissions.
- **Vulnerability Management:** Regular security audits and penetration testing will be conducted to identify and address potential vulnerabilities.
- **Secure Coding Practices:** Secure coding practices will be followed during development to minimize software vulnerabilities.
- **Compliance:** The system will adhere to relevant data privacy regulations regarding user data collection, storage, and usage.

5.4 Software Quality Attributes

- **Reliability:** The system should exhibit high reliability with minimal downtime or errors during operation.
- **Scalability:** The system should be designed to accommodate a growing user base and increased usage demands.
- **Usability:** The user interface should be intuitive and user-friendly to facilitate easy adoption and navigation.
- **Maintainability:** The codebase should be well-documented, modular, and maintainable for future updates and bug fixes.
- **Security:** The system should prioritize robust security measures to protect user data and system integrity.
- **Prioritization:** Security and Reliability are top priorities for this application. Usability and Maintainability are also crucial for user experience and long-term development.

5.5 Business Rules

- User registration requires a valid email address for verification purposes.
- Users can only create one account per email address.
- Users have the right to manage their privacy settings and control access to their profile information.
- Only registered users can initiate chats with other users.

These business rules define user behavior and system functionalities within the application. They may translate to specific functional requirements during the development process.

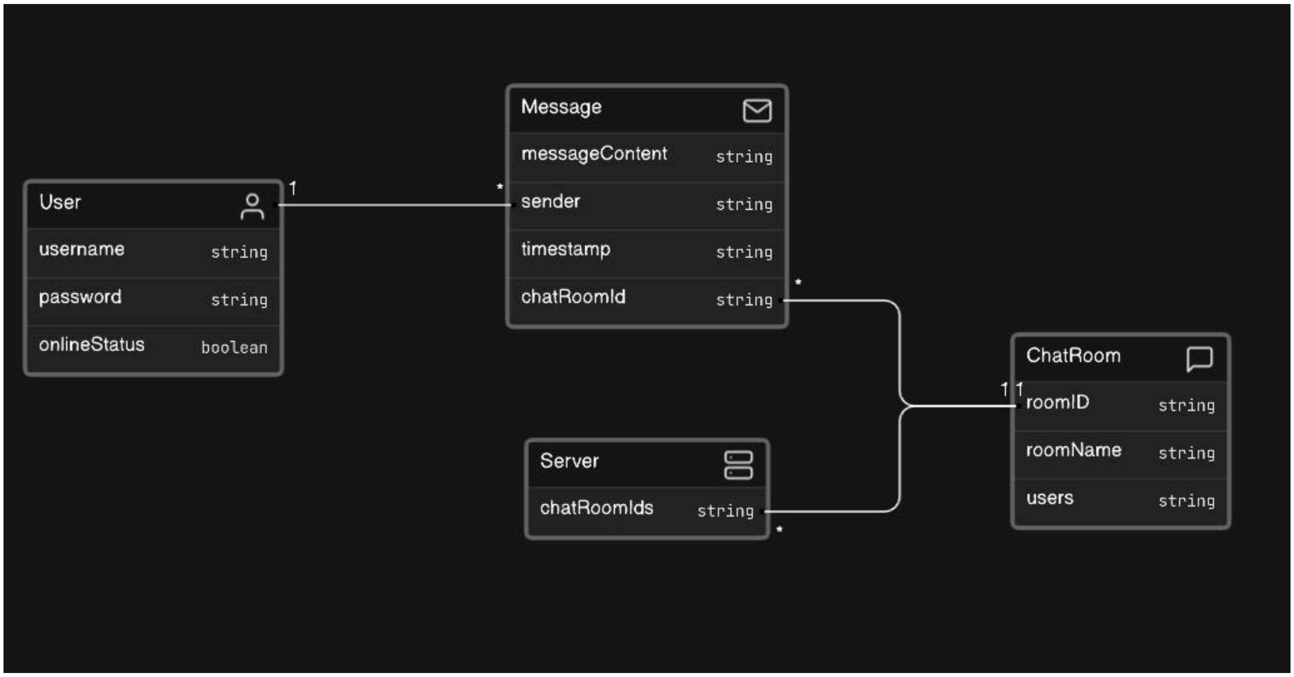
6. Other Requirements

- **Database Requirements:** The database should be able to efficiently store and retrieve user data, chat messages, and other application data. Scalability and reliability of the database solution are crucial considerations..
- **Testing:** A comprehensive testing strategy will be established to ensure the functionality, performance, and security of the application. This includes unit testing, integration testing, and user acceptance testing.

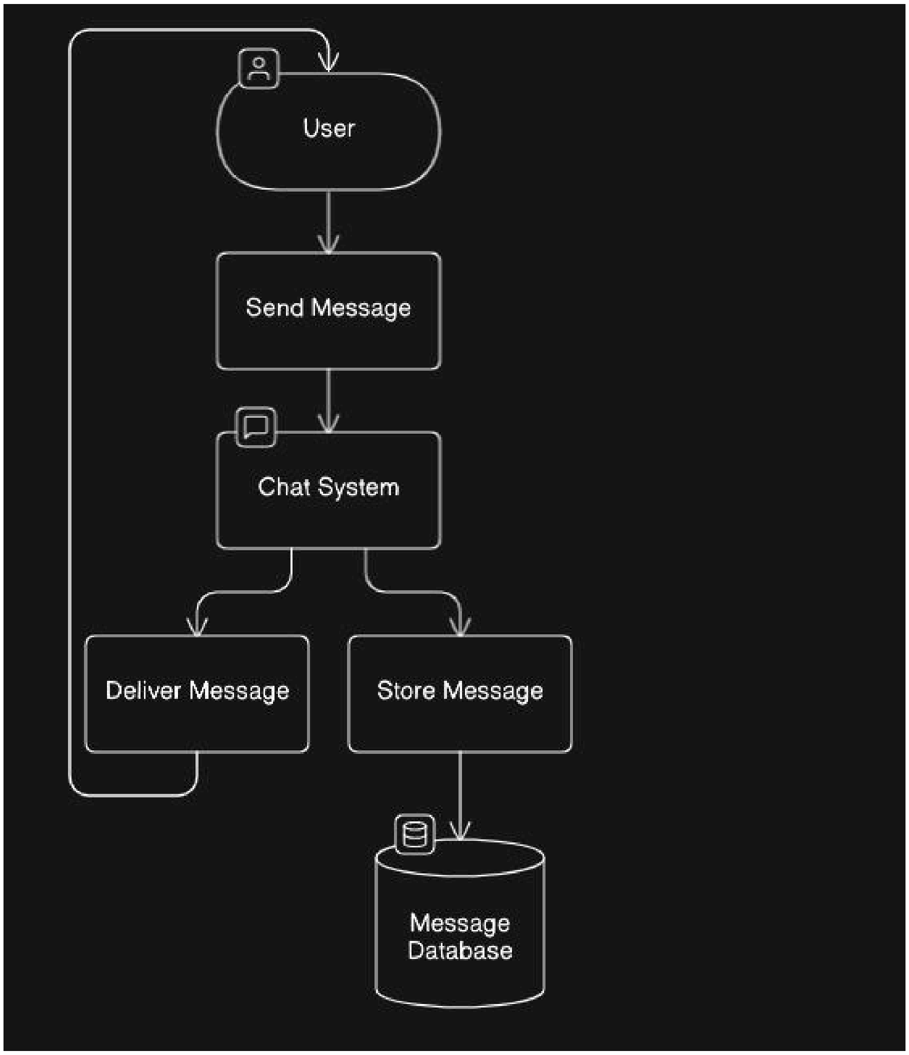
Appendix A: Glossary

- **API (Application Programming Interface):** A set of protocols and tools for building software applications.
- **HTTPS (Hypertext Transfer Protocol Secure):** Secure communication protocol that encrypts data transmission between a web server and browser.
- **JWT (JSON Web Token):** A compact and self-contained way for securely transmitting information between parties as a JSON object.
- **MERN Stack:** A software development stack comprising MongoDB (database), Express.js (web framework), React.js (front-end library), and Node.js (JavaScript runtime environment).
- **Real-Time Communication:** Enables continuous exchange of data between users with minimal delay.
- **TBD (To Be Determined):** A placeholder indicating that specific information is yet to be finalized.
- **UI (User Interface):** The graphical elements of a software application that users interact with.

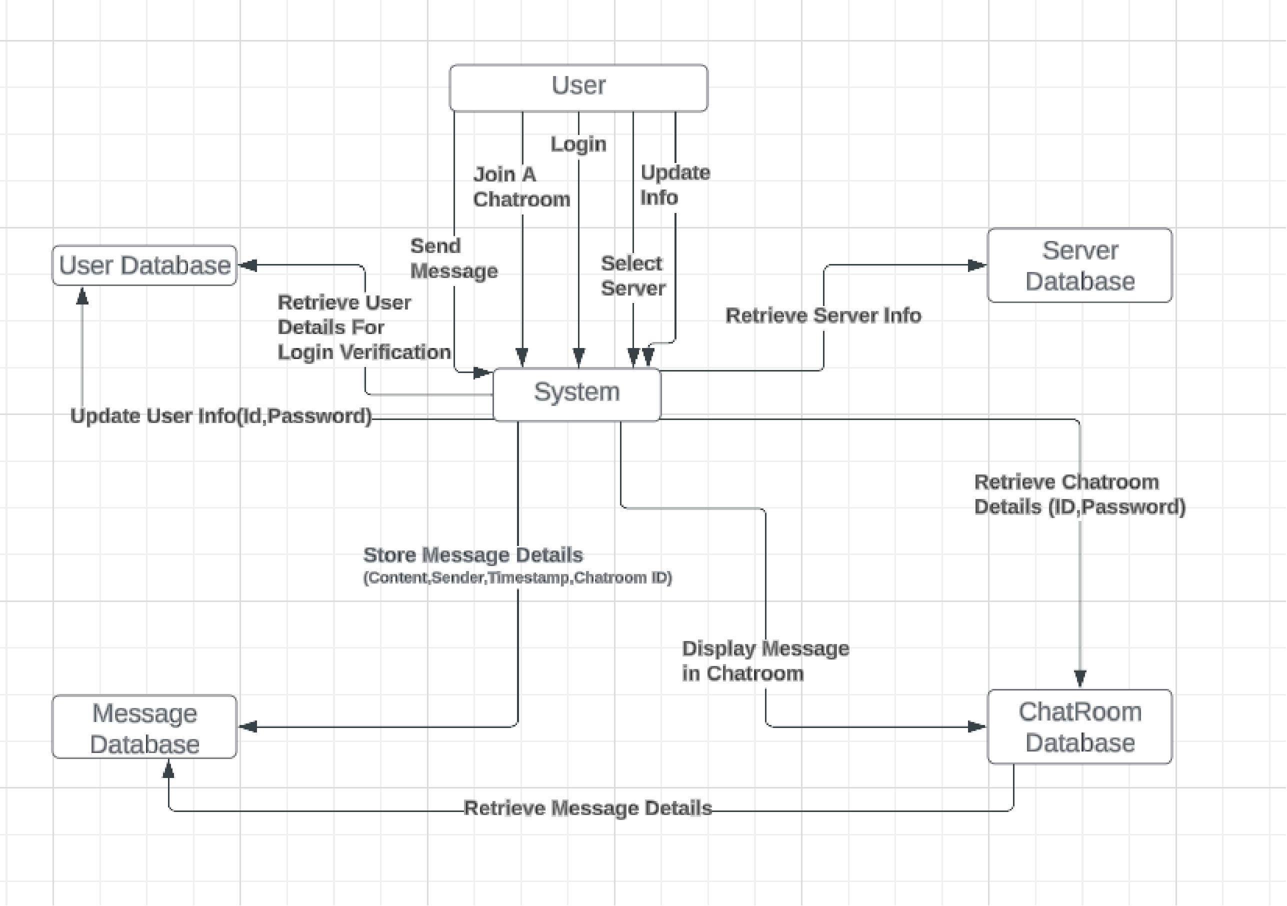
Appendix B: Analysis Models



(analysis class diagram)



(level-0 data flow diagram)



(level-1 data flow diagram)

Appendix C: To Be Determined List

- Load Handling capacity - Needs further analysis.
- Average message delivery latency - To be determined through performance testing.
- Additional security requirements that need to be addressed.

This list will be updated as requirements are finalized throughout the development process.