1. Project Overview

1.1. Objective:

The main goal of the project is to develop a real-time chat application that allows users to communicate seamlessly in a virtual environment. This application aims to provide:

- An efficient platform for personal and group communication.
- A user-friendly interface for real-time message exchange.
- Enhanced connectivity features such as private messaging and broadcast messaging.

1.2. Scope:

Included Features:

- User Authentication: Secure login process to access the chat features.
- Message Broadcasting: Ability for users to send messages that are visible to all online users.
- Private Messaging: Capability for users to send messages privately to selected users.

Excluded Features:

- Video or voice call functionalities.
- Encryption of messages for security purposes.
- File sharing and multimedia messaging.

2. Functional Requirements

2.1. Core Features:

- **User Authentication**:

  Description: Users must log in with valid credentials to access the chat functionalities. This includes a registration process for new users.

- **Message Broadcasting**:

  Description: Allows users to send messages that all connected users can receive and view, fostering a group chat environment.

- **Private Messaging**:

  Description: Enables users to select one or more users from the list and send messages visible only to the selected recipient(s).

2.2. Additional Features:

- **User Presence Indicator**:

  Description: Indicates whether a user is online, offline, or busy.

- **Message Notifications**:
  - Description: Alerts users to new messages when they are not actively looking at the chat application or when they receive a private message.

3. Technical Requirements

3.1. Technology Stack:

- **Programming Language(s)**: Java
- **Frameworks**:

  Frontend: JavaFX for building the graphical user interface.

  Backend: None; plain Java using sockets for simplicity and educational purposes.

- **Tools**:

  Development: Eclipse as the IDE.

- Version Control: Git for tracking changes
- **Libraries**:

  Networking: Java Sockets for handling real-time communication.

  Concurrency: Java Concurrency Utilities for managing multiple user connections simultaneously.