Documentation for Java Chat Room

Introduction

I build a simple chat application using Java sockets. The application consists of a server that handles multiple client connections and allows clients to exchange messages with each other.

Main Features of the Project:

Client-Server Architecture: The chat application follows a client-server architecture, where multiple clients connect to a central server to exchange messages.

Login and Logout system: The login and logout system are implemented to provide user authentication and control access to the chat server. When the client connects to the server, they need to provide a username to log in and when they want to disconnect from the server, they can choose to log out.

Socket Programming: The communication between the server and clients is implemented using Java sockets. Sockets provide a low-level interface for network communication.

Multi-threading: Each client connection is handled in a separate thread on the server to enable concurrent communication between multiple clients.

User Interface: The project includes a graphical user interface for both the server and clients. The server displays event messages, and clients can input messages to send to other clients.

Message Exchange with timestamps: Clients can send messages to each other, and the server acts as an intermediary for message routing. Clients can also request a list of currently connected users.

Change server address and port number: Clients can change the server address and port number to connect to the desired server.

Check list of online users: Clients can press the "List of online users" button to check the usernames of the active users.

Implementation details:

1. Run time environment and project information

The Java was developed using IntelliJ IDE. The project consists of three main Java classes: ChatMessage.java, Client.java, and Server.java. Also it consists of 2 classes called ServerGUI.java and ClientGUI.java for handling the graphics user interface.

2. ChatMessage Class

The ChatMessage class represents the messages exchanged between clients and the server. It contains fields for the message type (e.g., WHOISIN, MESSAGE, LOGOUT), as well as the message content.

3. Client Class

The Client class represents a client connection to the server. It handles the communication with the server, sending and receiving messages. It also includes methods for starting the client, sending messages, and handling disconnections.

4. Server Class

The Server class represents the central server that handles multiple client connections. It accepts client connections, creates separate threads for each client, and facilitates message exchange between clients. The server class also includes methods for starting the server, handling client disconnections, and broadcasting messages to all clients.

How to run:

1. Firstly, you need to install JDK 11 or above.

   You can download it from this link:

   https://www.oracle.com/sg/java/technologies/javase/jdk11-archive-downloads.html
2. Extract the zip file "122040040-Java_Chat_Room.zip" into a preferred folder.
3. Start the server by running the Server.java file.
4. Start a new client by running the ClientGUI.java file.
5. Enter a new username in text field and click "Login" to start chatting.
6. To simulate multiple clients, run multiple instances of ClientGUI.java file.
7. When done chatting, press "Logout" to disconnect from the server.