

This chat Application consists of a chat server and multiple number of chat clients. The server accepts connections from the clients and delivers all messages from each client to other clients or to a specific client for private message. Users login with an account and gain access to the chat server. This application is a real time instant messaging app that has been implanted using the Java language and frameworks. The project can be downloaded at this link <https://github.com/Eduardo-Robles7/ChatApplication.git>

This Chat Application has been divided and implemented in components that work together to build the complete Chat Application. The main components are the Server, User accounts, Client Handler and Client Chat Window. Together these form the instant messaging service and allow for storage of chat records.

The Server component is built using sockets and server sockets from the java networking tools. The Server is always running and waiting for clients to connect. Once a client connects it transfers that socket connection to a client handler so the Server can keep on listing for new clients.

The Login client is then used to connect to the Server and login in with an existing account or register a new one. Server validates the info and if valid it will present the User with a new client chat window. If the info is not valid, the server will reject and the person can keep on trying to login. When people are using the client chat window, they have space to enter text and a chat area where all the messages will be displayed.

The client chat window will also have a list of online users and will keep track of when people join or logoff. Every chat account has a place to store all chat records. When a user sends or receives a message, it will automatically be written to their records for future access. All this is managed by the Chat Log class. Together all these components makeup the full functional chat application.

Many bugs and errors were found during the unit testing phase. One of the main problems I encountered was accepting multiple socket connections in my Server class. If two users tried to login at the same time it would crash. This was fixed by introducing threads and making the method synchronized. I had similar problem with multiple Chat Windows being opened. If one user logged out of the chat, it would end closing all the chat windows instead of the single one. This was fixed by making multiple threads and fixing the action when someone pressed the closed button. The multiple testing of all the classes helped eliminate most of the bugs and allow for a smooth experience.

Instructions to run the Program

1. First launch and run the Server.java. Without this nothing else will work or open. It is the central main component.
2. Run the LoginClient.java, This will prompt a user login screen where you can enter your username and password. If no account, users can register a new account by clicking button.
3. If login was successful, Client Chat Window will appear automatically and will have you signed in. From here you can chat and receive messages.
4. To send a private chat, click the person's username on the sidebar and send your message.
5. To exit the Chat simply press the exit button on the top left corner of the window. Everything will be saved to the chat log.
6. Server will keep on running and listening until it is shutdown.
7. To shutdown Server, kill the process.