

Daniel's Chat Application - Project Report

Launching Daniel's Chat Application

In order to run Daniel's Chat Application, download and unpack the zip file containing the program files. In your IDE, map your console to these files and run the `Server.java` file--this opens the server so that a client can connect to it. Once that is running, run the `ClientMain.java` file. A new window should open with three text fields and two buttons--one of which being "Sign In." Run the `ClientMain.java` file as many times as you would like; each time you open it, it will open a new client session for a new login.

Registering a New User Account

From the main chat window, click on the "Sign In" button. This will open a new, login window where one can register a new user account or to connect to an existing user account. The username and password is of type `String`, so letters, numbers and special characters are allowed. There is no length restriction. Once a username and password have been entered, click the "Register" button. If registration fails--if the entered username already exists--a message will appear in the status field indicating the error. If registration is successful, the login window will close and take user back to the main chat window. User will still have to log in with newly registered credentials.

Signing In

After running the `ClientMain.java` file and opening a chat window, click on the "Sign In" button at the top right corner. A login window will open. Enter in the username and password and click the "Login" button. If login fails--incorrect username or password--a message will appear in the status field indicating the error. If login is successful, the user is returned to the main chat window; the "Sign In" button should now be the "Sign Out" button. Next to the "Sign Out" button, the current user's username should appear. In the righthand side text field, there should be a list of online users.

Sending a Message to All Online Users

In the main chat window, once logged in, there is a text field at the bottom where the user can type a message. Typing a message here and pressing “Enter” on the keyboard or clicking the “Send” button will send the message to all online users. Sent message will appear in the larger text field above indicating the user who sent the message and the time they sent it. All online users will receive the message and be able to respond in the same fashion.

Sending/Receiving a Private Message to/from Single User(s)

In the main chat window, once logged in, double-click on an online user from the online users list. This will open a new private chat window. Current user can private chat with as many single online users as are available. The label at the top right will indicate who current user is chatting with. The text field at the bottom works similarly to the group chat message--type a message and click on the “Send” button to send message to selected user. User will receive the message; if a window was already opened for communication with the sender, new message will appear in the chat log. Otherwise, a new chat window will open with the new message.

Accessing Chat History with Another User

In a private chat window with another online user, current user can press the “History” button. This will display any chat history between the two user in that chat log text field.

Signing Out

From any window in the chat application, click on the red ‘x’ at the top left of the window or click on the “Sign Out” button on the chat window. This will log user out of the application and close their connection with the server. A notification is sent to all online users’ group chat log indicating that user ‘x’ has logged out.

Notes on Programming Daniel’s Chat Application

This was quite a new experience for me. I had not worked with UI or with a server before. I had to do a lot of research to figure out how it works and how to program for it. The UI turned out to be fairly intuitive since the IDE had a plugin that allowed me to see the design and work with a physical window instead of just code. I definitely relied on a lot of my research to accomplish the UI windows and server connection aspects. Otherwise, I think the code is rather straightforward. There are classes for the different working parts of the chat

application--different types of chat windows, the server, the client-side Client. This was a fun exercise and very rewarding to see a final product where I can actually begin to envision how much larger, more familiar applications similar to this are built.