

William Trefethen

CS300

Term Project

My chat application is implemented in a very “vanilla” fashion - there is 1 simple external library which allows for JSON encoding and decoding on the server side (<https://github.com/fangyidong/json-simple>). Other than this it is pure Java. Because of that it's fairly straight forward to set up. I included in my repository a build folder which contains executable jar files for the server and the client. Each is in its own folder with a resources folder which is required for them to run correctly. For the server, the resources folder contains 2 JSON files for saving chats and users. **Extra Feature** by entering “s” into the server console the server will save all the current chats and user data to the JSON files and shut down. This is useful if the system needs to be put down for maintenance , etc. In the client resources folder is a host of GUI resources. **Extra Feature** my client features a fully polished, cross platform GUI that goes above and beyond the realm of normal Swing applications. For starting either the server or the client, simply take either folder (server or client) and place it on your machine. Navigate to the folder and execute the jar as normally (java -jar talk-server.jar).

Other Extra Features

The server has support for disallowing multiple logins from clients - if an account is correctly logged into while the user is already logged in, a notification is shown client side that the user is already logged in and the system disallows them from logging in.