My final project will be a chat room server and client with user registration/login, messaging between all logged in users, and support for private messages between users. The server will be built using listeners with no actual interface. A CLI and GUI will be built on top of the server using listeners from the server interface. The application will use threading for the server. The server's main thread would be for interfacing with listeners. The server socket will accept incoming connections infinitely on a new thread, and each new connection will be on its own thread as well. The client will also take advantage of multi-threading. The client's main thread will handle the GUI and interface with listeners on a ServerConnection, which will run on its own thread. The separate thread for the client's ServerConnection is to prevent blocking in the case that the server closes the connection or the client connection is lost. It's not necessary but it could also be useful in situations with high latency between the client and server.

The initial implementation of messaging will be very basic. Private messages will only appear as a different color in the same message window as the main chat room. If I have enough time I will implement a separate section purely for private messaging between users. Private messages will be initiated through either a chat command or a drop down menu on the GUI. The client GUI will support the changing of user passwords and possibly disabling notifications of new messages during quiet times.

The server itself will support flat file logging of everything occurring on the server. This includes when the user connects, disconnects, sends a message (either private or globally), as well as any issues the server encounters while running. The client and server will share an interface specifying the port that the server should run on and the client should listen on.

The GUI interface for the server will include all features the CLI has; starting and stopping the server, basic user management (delete, add, change password, change permissions), as well as support for sending global messages and private messages. If time allows for the addition of more complicated features that would only be suitable for a GUI interface, they will be added as well (though nothing comes to mind right now).

The server CLI will be basic. It will only support starting and stopping the server as well as basic user management. The CLI will provide a basic interface for adding new users, deleting users, changing user permissions, and changing their passwords. The CLI is intended to only be run from a computer server environment (where the rendering of a GUI is impossible) and testing. It will also offer the ability to send and show messages to all connected users. It'll also likely be possible for the server console to see the private messages between users, as a sort of all-around logging purpose.

The client’s GUI will initially make a connection to the server. If no connection can be made it will ask the user if they want to try to connect to the server again. This will continue until they cancel or a connection to the server is created. Once a server connection is established it will ask if you would like to log in to an account or create a new one. Once the user logs into their account they will retrieve a brief chat history from the server and then be allowed to send messages to everyone connected or private messages to specific users. All of the networking is handled in a separate thread to prevent blocking and listeners will be used to communicate with the GUI.