COMP3331 Report Henry Ye z5163982

Overview:

Written in Java 11.

My Server uses userThreads to handle concurrent client connections to the Server.

Each Client has a ReadThread and a WriteThread to allow it to concurrently read from and write to the server.

Upon running the server:

1) Server is instantiated using information from the command line prompt (java Server.java 6000 60 60)
2) Server then reads in credentials.txt file and stores details in various hashmaps, which are used for other implementations
3) Server is now ready to listen for requests
4) Upon request, a thread is started for the client
5) This thread handles most of the functionality (most method implementations written in here)

Upon running the client:

1) Client is instantiated using information from the command line
2) Client sets up read and write threads to interact with server.

Application Protocol:

The Userthread in Server reads in line from input, and passes it to applicationProtocol method.

This method breaks down string using StringTokeniser, and uses the first token as a keyword to carry out different methods within the implementation.

This is done with a switch case.

Function is quite intuitive → see Server.java line 153

Design improvements:

- Some implementation could be moved to client
- Exceptions could be handled better
- Logout function made more streamlined for the Client, and the thread within Server attached to that Client
- More functions to make code neater

Problems:

Due to time constraints, I could not implement P2P messaging correctly, and time-out only works after a message gets typed. For example, if timeout is 60 seconds, it will time-out when a message is typed after 60 seconds, rather than timing out after 60 seconds with no message.

Also, I had trouble understanding logout, regarding closing the Client, and the thread handling that Client. This did not affect implementation, as I did a bandaid fix within Client.java, executing System.exit(0) whenever and exception is caught, however it is not good code and will probably not be perfect under rigorous testing.

P2P File Transfer not implemented.

Borrowed code:

https://www.codejava.net/java-se/networking/how-to-create-a-chat-console-application-in-java-using-socket

I had trouble starting the assignment, especially threading, so I used the above website to help build an idea of how this application worked.
readThread and writeThread were inspired by the above source code.