**SE3313 Final Project – Design**

How do the various parts of the app work together?

The overall design of the application exists of a server and client side.

The server side uses the socket and thread libraries to handle connections. The server is ran and a server thread is made to listen for all connections. When a connection is received, it handles the client on a DIFFERENT thread (this is what allows multiple connections). The server socket can receive messages from clients and sends that message to all the clients connected to the server so they can interact in our chat room. Terminating is done by pressing the disconnect button.

The client side opens a socket, connects to the server with the “connect” button, and then is connected to the server. They can send messages to every client connected to the server and can disconnect and re-connect at any time as long as the server is running.

How many threads do you need?

As many clients as there are in the chat room.

Synchronization objects used and explained?

We have a list of clients that is kept to synchronize the many client threads that are simultaneously connected to the server. When a client connects, they are added to the list. When they disconnect, the socket is closed for that client and THEN removes the client from the list to let the server know not to transmit or receive from that client anymore.

How do you handle termination?

Gracefully – client disconnects which is sent to server which responds and sends message back. It then does not transmit or receive any data to that client. The server does this by closing the socket and removing that socket from the list “clients” which is used to keep track of current clients in the chat room.