Assumptions made during the design of this project:

- The client will only terminate by send a “quit” command to the terminal or when the server receives a SIGINT signal and shuts down. By no means will a client shut down with a CTRL + C.

- Inputs will be reasonably formatted. Assuming the user will not input bad input such as “withdraw 100 200” or “deposit 5 5 5 5 5” or “deposit” or “create name anothername”

Design:

-All thread requirements were met for this project. The client program spawns two threads for sending commands to the server and receiving responses from it. The server spawns an accept thread and that threads spawns a new thread for each client connection.
-The server when receiving a SIGINT signal, will kill the accept thread thereby disconnection all clients.
-When a client terminates, it was necessary to cancel the response and connect threads as joining them put them into a deadlock.
-When the server prints out data, since we were unable to figure out semaphores in time, we simply locked all our mutexes before diagnostic printing commenced and then the mutexes were unlocked after the printing was finished.