# Design Document

**Design:**

As far as designing aspect goes, we kept it simple. We followed the same design as the one in the demo except for a few tweaks here and there. For the database we went with a 2D array. To make it easier on we, made each user take an indexing of 3. That way we would be able to fit the name, Ip address and port. We also used if statements inside the for loop for the client and server. We tried going with python at first and looked at the book's code. Once we figured out that we wouldn't have a lot of trouble handling the memory, we switched back to C and and modified her code!

**Description of message:**

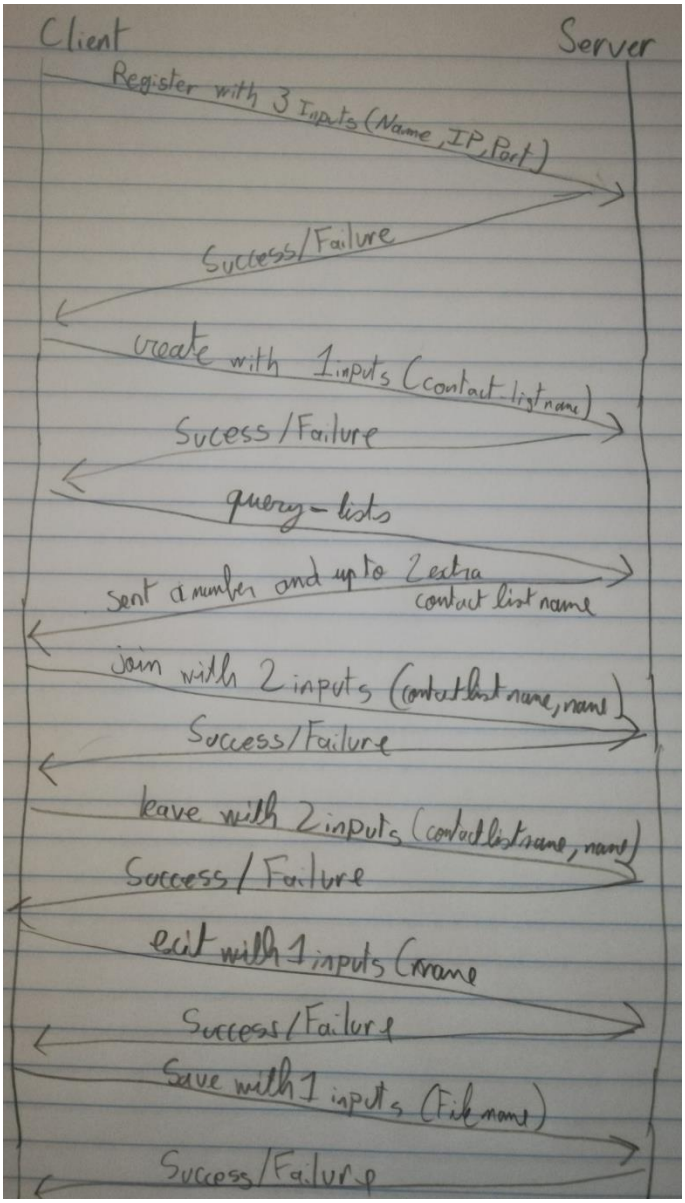
For active users, only 3 would be allowed online at once. Anymore and there would be a FAILURE message. Once the users registered, they would be able to interact with each other if they were in the same contact list. Users also had the freedom to pass the following commands to the server:

1. **register <contact-name> <IP-address> <port> :** This is a command a user could pass to the server to register their info like IP address and port. If the total number of users is less than 3, the return message would output SUCCESS. If there are already 3 users, the return message would output FAILURE.
2. **create <contact-list-name> :** The user can pass this command to the server to create a list that others would be able to join. A SUCCESS message will appear if they are able to create a list and FAILURE if they are unable .
3. **query-lists:** This is a command the user can input to check what lists are already created. It will display the name and the users to the user. Should the contact fail to be removed, a return code of FAILURE will appear.
4. **join <contact-list-name> <contact-name>:** When this command is passed, the server will look up the name as well as port and IP address. If it doesn't find anything, then a SUCCESS message will appear. If not than a FAILURE message will appear.
5. **leave <contact-list-name> <contact-name> :** This command will remove the named contact list and its associated information from the named contact list.
6. **exit <contact-name> :** This command lets the user know that they want to exit the IM and a success message. The user contact name will be removed as a well as from the contact list. If everything goes smooth, a return message of SUCCESS will be outputted if not then FAILURE would be outputted.
7. **save <file-name>:** This command will save all the info to a text file called *file-name.txt*

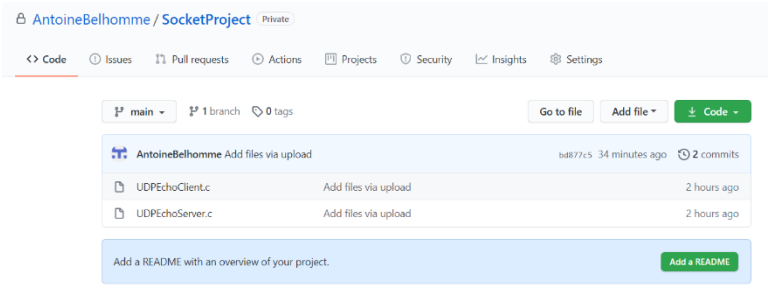
**Testing:**

We tested out commands with hard code. Basically, we found a case where it would have to return FAILURE and another case where it would return SUCCESS. We then applied it to every command and ensured that it was working correctly!

**Time space Diagram:**



**Link to GitHub:** (It's private per request of professor)




https://github.com/AntoineBelhomme/SocketProject

**Link to demo:**

https://www.youtube.com/watch?v=OA5Wl_Eo_7M