Design Document

In order to create this Machine Problem, I will start by creating the connectTo and login functions. I will use the example provided to us in the document describing how to begin the project to create a stub to use for communicating between the server and the client. The client will initialize the stub and then use it to call the login function. The login function will initialize a user by assigning it its name, any followers that were saved in a file for it, and itself as a follower. Then, the user will be stored in a vector of users and success will be returned.

For the process command function, I will do a similar string compare to my MP1 and search for the key words for each function: list, follow, unfollow, etc. After they are found, I will have the stub for the client call the corresponding command's function in the server. Then, based on the response from the server about if the commands were successful or not, I will set the status of the Ireply that will be returned at the end of the function.

For the processTimeline function, I will create a read and write thread that will be used to communicate with the server. The write thread will use the username of the client, the msg the client writes, and the time that the client writes the msg to create a Message to send to the server. The read thread will take in a Message and then use the data inside of it to output the necessary information to the terminal such as the msg and time. Finally, the threads will be joined.

The server functions will communicate with the client function using the msg inside of replies to communicate how successful the operation was.

For the List function inside of the client, I will have the function go through the users vector to find the user that inputted the command. Then, it will find the followers of the user and return that and the username in a reply to the client.

For the follow function, I will have the function by having it find the user that inputted the follow command based on the username. After finding the user, it will add the input username to follow to the user's following vector. The function will also find the input user to follow and add itself to that user's follower vector. After it is added to the vector, it will update the file that holds the followers. If the input user to follow is already being followed or is itself, then the function will not add the user and will respond accordingly.

For the unfollow function, I will have the function check if the input user to unfollow is itself or if that user isn't being followed already. If one of these things is true, the function will respond accordingly. If the user does need to be unfollowed, it will remove that user from the current user's following vector. It will then find the user to be unfollowed's follower vector and remove itself. It will then update that user's follower file to represent the changes.

The timeline function will read in the msg given from the client and then format and save that message into every follower's file for following someone. Each user will have a file dedicated to them that holds the messages sent on the timeline from the

other users they are following. The function will output the 20 most recent messages by finding the messages in the file for the current user and then appending the messages into a vector. The 20 most recent messages will be determined by getting the messages at the end of the vector first. After the messages are placed into the vector, the server will write them back to the client one at a time in sequential order. The files will not be overwritten each time so as to keep the timeline and followers even if the server and clients are restarted.

The github link for this assignment is:
https://github.com/Souldefier22/CSCE438/tree/main/MP2