MP2

Github link: https://github.com/ninarao09/CSCE438

I created a client struct to deal with clients. The struct looks like this:

```
struct Client{
  std::string name;
  std::vector<Client*> followers;
  std::vector<Client*> following;
  ServerReaderWriter<Message, Message>* stream = 0;
  int timeline_size = 0;
  bool isConnected = true;
};
```

Client

- ConnectTo()
  - In this function, I created a stub and called the login service function to correctly establish a user.
- ProcessCommand()
  - In this function, I processed each of the commands and called their respective service functions. After receiving a reply from them, I would display the appropriate prompt back to the terminal.
- ProcessTimeline()
  - This function just deals with when a client enters timeline mode. It calls the timeline service function.

Server:

- Login
  - Checks if the client was created already -> if it was, then return -1;
- List
  - Gets all the users from the clients_db vector
  - Gets all the followers from the clients_db[].followers vector
- Follow
  - Checks if the user exists and if the user tries to follow themselves
    - If either of those is true it returns the appropriate reply back
  - If not the function will push the user into the follower vector.
- Unfollow
  - Checks if the user exists and if the user tries to unfollow themselves
    - If either of those is true it returns the appropriate reply back
  - If not the function will remove the user from the follower vector.
- Timeline
  - While waiting to receive messages, I create a file for the user with just their name that logged all the posts they made. I also create a user timeline file that held all

of the posts of the clients who followed the main user. I only displayed the most recent 20 posts for each timeline. At the end of it, I basically send the message out to every follower in from the follower vector. I had tried to allow for persistence but I ran out of time.