Tuesday 2.21
- Working on putting the finishing touches on wire protocols and gRPC, mostly fixing up gRPC
- sit and write a detailed readme for both versions
    - describes all functionality of both files
- comment code, document what all functions do, how they operate, what their arguments are
- think through anything else that we'd need to do before the demo tomorrow

Monday 2.20
- Late night, gRPC client and server start working, all the functions we wrote actually seem to work
    - Turns out Jothi forgot to push changes to one of the files and that's why nothing worked. Once they were pushed, everything worked at once.
- Once we get the first gRPC function working (quit messenger) we get how the interface works and are able to get the rest working very quickly based on our wire protocols implementation
- we start splitting up work
    - Justin cleans up wire protocols, really tests things to see if anything can go wrong, fixes last bugs, cleans up UI
    - Jothi working on getting gRPC up and running, filling in other functions that we need to complete, writing gRPC unit tests
- Unit tests were annoying to figure out, but learned about mocks in pytest that were very useful, and could be used in future projects

Sunday 2.19
- Getting a bit stuck on gRPC, we have some functions that we believe should work but we haven't been able to test anything
- we both have a busy week coming up, figuring out best way to handle what is potentially going to be a very busy next couple of days
- Jothi writing a lot of gRPC functions that we think should work, Justin working on getting the gRPC client/server connection up and running

Saturday 2.18
- Clients are now able to send messages and have it be immediately delivered to another client that is logged into their account, otherwise it delivers to undelivered messages
    - Used background thread that could be very useful for gRPC implementation as well. We decided to poll a listening function in a while loop, controlling when it would listen and when it wouldn't. Sending msg length and the message afterwards complicated this a bit so we had to think very carefully about how to work around this, and we ended up using thread events and bool flags for turning on/off listening.
- starting to look into gRPC and find a couple tutorials that may be helpful in figuring out how it works

- refactoring wire protocols code and creating start.py file
  - now the interface will be as follows:
    - python3 start.py server → starts server
    - python3 start.py client → starts client
- Jothi working on writing the unit tests

Thursday 2.16
- In the last couple days we made progress on the following things
  - client and server are running, not seeing as many errors
  - multiple clients are able to connect and send messages to each other
  - clients can connect and view undelivered messages
  - clients are NOT able to immediately view messages from other users
- need to work on having another thread running at all times to look for incoming messages

Monday 2.13
- we have standardized our wire protocol and put the serialization functions into their own file called protocols.py
- we're now starting to work on the user menu and interface
  - plan is to work slowly, piece by piece through the app, fixing things that we break as we go for the most part

Sunday 2.12
- working on wire protocol
- " LENGTH OPERATION VERSION MSGSIZE TO FROM MESSAGE "
- we then realized it might be easier to just have TO FROM be part of the message
- " LENGTH OPERATION MSGSIZE VERSION MESSAGE "
- we thought about it more and were running into some problems with the padding of message size, then we realized it might not even be useful, just going to drop it for now
- wire protocol:
  - " LENGTH VERSION OPERATION MESSAGE "
- wrote serialize and deserialize functions and were able to make this work in the terminal
- evening: coding up wire protocol, things are working, superbowl about to start
- things are working
  - creating a new user now works, server shows all users, cannot create duplicate users

Friday 2.10
- went to Jim's OH in the SEC, figuring out what to do next
- got menu working
- added another menu for the user interface
- now just need to handle:
  - Login
    - view my messages

- - - send message
        - send to a specific other user, list of possible users to choose from
  - Logout
  - create account
  - list accounts
  - delete account

Thursday 2.9
- night: working on creating a menu using the curses library, running into a lot of debugging issues but generally getting things to work
  - three buttons you can pick from in main menu, create account, list accounts, delete account
- night: making server and client interactions clearer, adding comments for more ease of use on client end
- evening: Jothi and I spend a couple hours and are able to create server.py and client.py and pass messages from one computer's terminal to another using sockets
- evening: spoke to Jared Ni who suggested we check out this youtube video on sockets
- afternoon: reviewing sockets and notes, working on connecting our own computers first using just the terminal before beginning to create the project just so we understand how it works