Members: Oscar So (ons4) & Aidan Chalnick (ac2597) & Tushar Khan (tak62)

Meeting Plan: WFSu 3-5pm

Statement: Create an instant messaging system that will allow users to communicate with each other and share data from different computers on the same network..

Roadmap:

- **MS1:**
    - Satisfactory Scope:
        - Research different open-source messaging protocols and choose one to use for our system
        - Compile a bulleted summary of findings on chosen protocol and share with group mates
    - Good Scope:
        - Implement the chosen messaging protocol to allow access/communication between two computers.
    - Excellent Scope:
        - Build a contact list system to allow users to identify the users of other computers that are using our instant messaging system.
- **MS2:**
    - Begin adding extensions to the instant messaging system. Such extensions include the following:
        - Profile system and friends list
        - Group messaging
        - Emoji support (Excellent scope?)
        - Message reply threads for group chats
- **MS3:**
    - Anything unfinished that we want to add.
    - Sending more than text
        - Pictures
        - File sharing
        - Videos
    - Security features
        - Password protected chat room

Sketch:

What are the important modules that will be implemented? What is the purpose of each module? [1 point]

- User
  - This module will represent a user of the system.
- Messaging Protocol
  - This will be highly dependent on the messaging protocol system we decide to use (this will be a part of MS1 as we have not decided on one yet). This means this module might actually be implemented by multiple modules.

What data will your system maintain? What formats will be used for storage or communication? What data structures do you expect to use as part of your implementation? [1 point]

- Data about which computers are communicating (friends list)
- System will maintain all messages stored for a user in json format. For messages that are not text, the json would contain references to media stored elsewhere. In the program, the messages will probably be stored as a list structure.
  - Could also store the messages as some sort of key-value pair where keys are the person who sent the message
    - Would allow for searching for all the messages sent by one person, who is replying to who, etc.
-
- When we implement more than messages, we have to add support for storing different kinds of files

What third-party libraries (if any) will you use? [1 point]

- Not sure, but we will add libraries if needed. Probably would need a library to implement the protocol, but will try to implement it ourselves first.

How will you test your system throughout development? What kinds of unit tests will you write? How will you, as a team, commit to following your testing plan and holding each other accountable for writing correct code? [1 point]

- We will try messaging on it to see if all the features work. We think that testing this idea out would require real-time application and tests in order to figure out bugs (if any) lurking inside our project.