**Project1 Task2**

Nathaniel Leake

424003778

Time to complete Task: **4hrs**

**Part A:**

*How to run the code:*

To run the prototype from terminal, enter **python msg\_board\_v2.py**

The commands work the way they are specified in the Message Board PDF; enter **stop** to exit listening mode. One of the barriers I ran into when designing this was the program pausing when waiting for user input. Since I am new to Python, I had to do a lot of research to understand python multithreading so that I could have a separate thread waiting on user input “stop” to exit listening mode (while still listening for new messages). Another concern was real-time updating. I could not find a way to check for new messages without constantly querying the server, but the query I do for new messages is very simple and fast, and should scale properly for a large database of messages. I also set up several message boards on the MongoDB database, which are listed on start up and can be selected from the program menu. Also, it is important to note that I did not see a need to use Redis since MongoDB can supply all of the functionality I need and should remain fairly fast for larger databases.

Github: <https://github.tamu.edu/nateleake/489-17-c>

**Part B:**

onasd the23 server, such aaasdfs data corruption, cdgrashe2323s, accidental overwrites or queries, then the data could be damgaged irreparably. C23 The other concern is that the server could run out of space and be unable to accept new davta, have overflow issues, or signisadfficantly reduced performance. Spreading out the data over several servers to create a distrzsdibuted network is a good idea if AggieFit has the necessary funding, but there are alsoasdf database services online with reliable connectivity, built in redundancy, and scalable stoasdfarage that in a real scenario would be a better option. Speaksdfing of which, if AggieFit does continue to build its own server network, both of the questions they raised should be answered with “yes”. The data should be spreasdad across several servers, but it would also be good to keep 2-3 copies of the data if possible, or to run periodic backups to a separate set of servers.