**Summary**

The goal of this project was to simulate an email system with both the client side and the server side. Each client has the ability to send an emails/messages of 80 characters long to any connected, known or unknown user. The server allows a max of 100 clients/users and holds the messages for the respective users until the retrieve their emails/messages.

**Interactions**

Once the server is running the server creates a global TreeMap<String, ArrayList<String>>, a global int[100] and a global String[100].

Once a client is connected the server creates a thread for the client. The server captures the name of the client and uses the name as a key and inserts it into the global TreeMap, set the int[x] to a value of 1 indicating that the user is logged in and also stores the name of the user in the String[x].

If a client sends a message to a known user the server find that user the (recipient) in the TreeMap with the following method TreeMap.containsKey(name) and add the message to the ArrayList in the TreeMap with the following message TreeMap.get(name).add(message).

If a client sends a message to an unknown user meaning a user who has not yet logged in, then the server takes the name of the recipient and adds it to the String[x] and creates a Arraylist<String> and adds the message to the ArrayList. Once the unknown user logs in then the server creates a thread for it and turns on the the int[x] = 1 and get the name from String[x] and adds the user to the global TreeMap<name, ArrayList<String>>.

Finally, once if a client decides to retrieve it’s messages the server will use the client’s name to navigate the TreeMap and finds it’s messages. Once the messages are retrieve the client deletes the message from the ArrayList.

This project was very interesting but also challenging. This is the first time I have use a TreeMap and also the first time working with sockets. I also made used of the semaphore class I created from my previous project. The most difficult aspect of this project was to correctly map out all the possible interaction between the client and the server. It was also very difficult to debug since there are multiple threads running concurrently when the program crashes it was hard to understand what caused the crash. Overhaul I really enjoyed this project.