**BTC4201 / ICS4104: Distributed Systems**

**Assignment: Inter-process Communications in Distributed Environment**

Maina Allan Karanja, 101358

Shawn Kodhe, 112634

RUNNING THE PROGRAM:

1. Run server.java
2. Then run client.java, which opens an GUI
3. The program terminates when all the questions are answered.
4. Thank you.

There are four java files namely Client, Server, SocketClient and SocketServer.

SocketClient and SocketServer are used to keep the connection to the server and client, respectively, open, and thus able to communicate with each other. We use ServerSocket which waits for requests to come over the socket and does something.

In the client.java we implement a way for the client to send messages to the server through a buffer which is more efficient than reading one line at a time from the underlying reader. Calling the flush method clears the buffer getting it ready for the next line. Although buffers are mostly used with large files, we use it for the efficiency as mentioned above.

In the server.java, we echo what we receive from the input stream back to the client.

GUI

To generate a GUI (Graphical User Interface), we use the Swing Java library, though old it gets the job done.

JFrame creates a frame object(window) that we can attach a panel to. Panel forms the scaffold with which we can attach GUI objects i.e., buttons, labels, text etc. Using textField we can get the users input and write to the stream using the buffer above.

The program is written in a sequential manner to make it easier to understand program flow.

**References**

<Any additional references you use (either for background information or for citation) should be listed here, using the APA style>

(ThenisH, 2018)

(WittCode, 2020)