

## Programming Assignment 1

### REPORT

As noted in my code, I used several references that are denoted below this report.

#### *Server.java*

The Server file has two classes: *Server* and *ClientHandler*. The main goal was to take advantage of threading to handle multiple clients so they could all be online and talk with each other instead of having the server dedicated to just one client. I used localhost and port 1775 for the server connection. The server is always listening for client connections, and once a connection is established, it listens for client input.

#### *The Server Class*

I used an ArrayList to keep track of the clients as they logged in and as they left the chat service. The constructor uses ServerSockets and Sockets to establish the connection, enable the communication between the server and the client (by invoking the ClientHandler class and the start() method), and add new clients to the ArrayList.

#### *The ClientHandler Class*

Threads are used here. Every time a new client signs into the server, a new thread is created for them. The main portion of this class is the run() block. It uses a series of try-catch blocks with embedded if statements to ensure incoming strings meet requirements and action is taken on specific commands:

- Client username *must* be alphanumeric,
- Client is welcomed if their username is alphanumeric
- Clients' requested information (e.g., allusers) is printed to all the consoles,
- The client is disconnected when they're ready (e.g., bye)

Finally, this class ensures the strings are printed to all active consoles. A PrintWriter and BufferedReader handle the output to the consoles and reading in of the input streams.

#### *Client.java*

The client connected using localhost and port 1775. I took advantage of threads again in my sendMsg and readMsg methods. For both methods, I overrode the run() method send/receive messages. Once the threads were set-up, I started both methods so the client was always listening for other client input.

## REFERENCES

- \*Baeldung's "A Guide to Java Sockets"
- \*Geeks for Geeks' "Introducing Threads in Socket Programming"
- \*Geeks for Geeks' "Multi-Threaded Chat Application in Java"
- \*Java8 API (various searches)
- \*Sheldon Guillory, classmate
- \*Stack Overflow (various searches)

## SERVER

```
gtrj@DESKTOP-GS MINGW64 ~/OneDrive/Desktop/gtrs/UNO/4311/PA1/Swanson_PA1
$ java Server 1775
Server started.
Waiting for client.

Client accepted.
Client 1
SERVER: Welcome, Maria

Client accepted.
Maria
Client 2
SERVER: Welcome, Josue

Client accepted.
Maria
Josue
Client 3
SERVER: Welcome, Juanita

Juanita: AllUsers
Juanita: bye
Josue: Hi, everyone
Josue: Who's here?
Maria: Me, but Juanita left.
Josue: AllUsers
Josue: Bye, everyone
Josue: BYE
Maria: Is it just me, or is everyone gone?
Maria: ALLUSERS
Maria: Adios.
Maria: Bye
```

## CLIENT 1

```
gtrj@DESKTOP-GS MINGW64 ~/OneDrive/Desktop/gtrs/UNO/4311/P
A1/Swanson_PA1
$ java Client
Enter your alphanumeric name:
Maria
SERVER: Welcome, Maria
SERVER: Welcome, Josue
SERVER: Welcome, Juanita
Juanita: AllUsers
Josue: Hi, everyone
Josue: Who's here?
Me, but Juanita left.
Josue: AllUsers
Josue: Bye, everyone
Is it just me, or is everyone gone?
ALLUSERS

The list of all users is:

Maria

Adios.
Bye
SERVER: Goodbye, Maria
```

## CLIENT 2

```
gtrj@DESKTOP-GS MINGW64 ~/OneDrive/Desktop/gtrs/UNO/4311/P
A1/Swanson_PA1
$ java Client
Enter your alphanumeric name:
Josue
SERVER: Welcome, Josue
SERVER: Welcome, Juanita
Juanita: AllUsers
Hi, everyone
Who's here?
Maria: Me, but Juanita left.
AllUsers

The list of all users is:

Maria

Josue

Bye, everyone
BYE
SERVER: Goodbye, Josue
```

### CLIENT 3

```
gtrj@DESKTOP-GS MINGW64 ~/OneDrive/Desktop/gtrs/UNO/4311/  
PA1/Swanson_PA1  
$ java Client  
Enter your alphanumeric name:  
Juanita  
SERVER: Welcome, Juanita  
AllUsers  
  
The list of all users is:  
  
Maria  
  
Josue  
  
Juanita  
  
bye  
SERVER: Goodbye, Juanita
```