CS 4390.501
4/26/22
Team 11

Project Report

I. Group Members

Aidan Johnson (amj180004)
Hyungtaek Kim (hxk151830)
Kevin Lobo (kal160130)

II. Protocol Design

The first message the server receives from a client is assigned to be that client's username. The client then receives an ACK to tell them that the connection is open. The server assumes that 1 of 2 message types can be sent after this: a valid mathematical equation, or a communication end keyword ("quit"). If the server receives an message that does not contain "quit", it will attempt to evaluate it as a mathematical equation. The result is stored as an integer (meaning any decimals will be cut off) and then sent back to the client, also being logged on the server.

When "quit" is sent, the client is removed from the connection.

III. Programming Environment

The server and client programs were both developed using the Java language on the NetBeans IDE and the Windows operating system.

IV. Compiling and Executing

Our program was executed from the Windows command prompt. Each client requires a separate command prompt instance, and the server also needs its own command prompt instance. We have included the class files and the Java source code, as well as a makefile if the user wants to generate class files themselves.

V. Parameters Required

The port and IP address used are hardcoded into the files. The user will not need to input those parameters. The input taken from the user consists of the username, plus math equations being sent by the client, as well as the "Close" keyword which closes the connection for that client only.

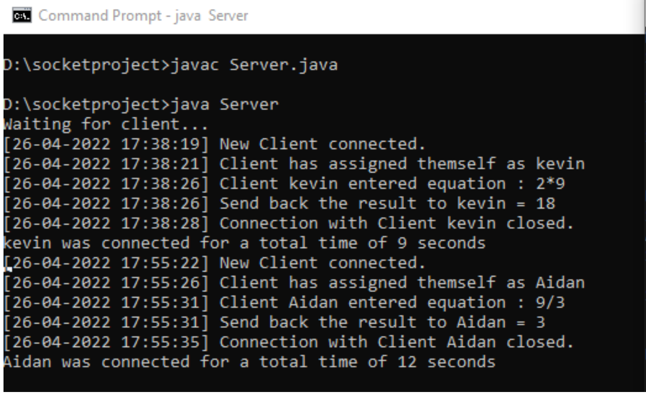
VI. Challenges Faced

Because the client and server files were being worked on by different authors with different coding styles, there was difficulty in standardizing the way these files would communicate with each other. It took a while to get the format of messages consistent between client and server. Also, levels of experience with Java were different between team members, meaning not everyone knew the same concepts.

VII. What We Learned

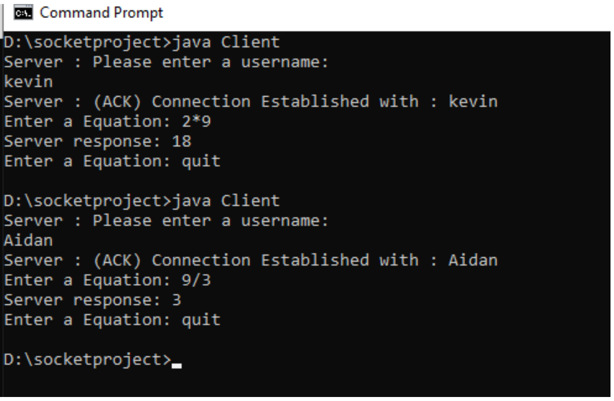
One of the important things we learned was how to implement multiple clients simultaneously. We had not written programs of that nature before, so it was a new experience. We also learned how to gather information about clients into a log.

VIII. Output Screenshots



```
D:\socketproject>javac Server.java

D:\socketproject>java Server
Waiting for client...
[26-04-2022 17:38:19] New Client connected.
[26-04-2022 17:38:21] Client has assigned themself as kevin
[26-04-2022 17:38:26] Client kevin entered equation : 2*9
[26-04-2022 17:38:26] Send back the result to kevin = 18
[26-04-2022 17:38:28] Connection with Client kevin closed.
kevin was connected for a total time of 9 seconds
[26-04-2022 17:55:22] New Client connected.
[26-04-2022 17:55:26] Client has assigned themself as Aidan
[26-04-2022 17:55:31] Client Aidan entered equation : 9/3
[26-04-2022 17:55:31] Send back the result to Aidan = 3
[26-04-2022 17:55:35] Connection with Client Aidan closed.
Aidan was connected for a total time of 12 seconds
```



```
D:\socketproject>java Client
Server : Please enter a username:
kevin
Server : (ACK) Connection Established with : kevin
Enter a Equation: 2*9
Server response: 18
Enter a Equation: quit

D:\socketproject>java Client
Server : Please enter a username:
Aidan
Server : (ACK) Connection Established with : Aidan
Enter a Equation: 9/3
Server response: 3
Enter a Equation: quit

D:\socketproject>
```

IX. Division of Work

Aidan: Document author, and wrote code to process and solve equations
Hyungtaek: Designed and coded most of the client and server
Kevin: Wrote code to log information about clients