Project Scope

For this project, we will be creating an application that utilizes multithreading in order to facilitate connections between a client and a server. Specifically, we will use threads to allow for multiple connections to the server while the client will not use threads. Both client and server will ask the user to choose what type of connection they want to have. These connections can be either TCP or UDP. TCP is a more secure and reliable connection that requires acknowledgement of receipt of messages, but it is slower. UDP is not at all reliable or secure, but is very fast. The purpose of this project is to allow us to learn about the differences between TCP and UDP and see examples of those differences in code.

Project Requirements

* Create a client class
  + The client class should ask the user if they want to use TCP or UDP
  + The client class should ask the user for server IP or host name
  + The client class should ask the user for the port number
  + For each connection that succeeds, the client should print out relevant information like the IP, type of connection, hostname, port number, and timestamp
  + For each message received that the client receives from the server, the client should echo the message to the console with a timestamp
* Create a server class
  + The server class should ask the user if they want to use TCP or UDP
  + When the server is successfully set up, it should print the IP, host name, connection type, and the port number
  + For each connection that is made with a client, the server should print out relevant information like the IP of the client and a timestamp
  + For each message that the server receives from the client, the server should echo the message to the console with a timestamp
* Inline documentation should be made to explain every function
* Documentation file explaining how the project works
* Readme file with instructions for running the program

Project Deliverables

* Project Plan: Complete a document that describes the requirements for the project and the plan for completing it
* Implementation: Create the code for the project and any necessary documentation to use and run it
  + Documentation file: Create a document explaining how the project works and how the code works together focusing on the sockets
  + Readme file: Create a document explaining how to use and run the program with all its different options and how to run both the client and the server
  + Project source code: Write the necessary code to solve the problem
  + Compiled project: Compile the finished product and include the class files created
* Reflection: Contemplate the process of completing the project and how it could be improved. Also think about personal contribution and growth from the project.

Project Design

For this project, we are going to use Java. The server will communicate with the client via sockets. Depending on the choice of the user, the sockets will use either TCP or UDP. After the server is running, it will print necessary information like the IP, host name, connection type, and the port number. If a client connects with the server, the server will print out necessary information about the client. When a connected client sends a message, the server will echo it. Similarly, the client will communicate to the server via either TCP or UDP. The client will ask for necessary connection information like IP, host name, connection type, and port number before connecting to a server. After connecting successfully, it will print some connection information again. The client will also echo any messages sent from the server.

Pseudocode

Server:

Print “Enter TCP or UDP:”

Collect user input

Print “Enter port:”

Collect user input

If user input is valid and is TCP

Create TCP server

If user input is valid and is UDP

Create UDP server

Else

Print error

When a connection occurs

Print connection info

When a message comes in from a connection

If connection is TCP

Echo the message and timestamp

Send acknowledgement

If connection is UDP

Echo the message and timestamp

Client:

Print “Enter the name or IP address of the Server: ”

If no ip or hostname

Print error

Else

Save ip or hostname

Print “Enter TCP or UDP:”

If no protocol

Print error

Else

Save protocol choice

Print “Enter Port:”

If no port number

Print error

Else

Save port number

Use user options and connect to server

If connected

Print information about connection

Else

Print error

Once connected, waits for user message

Prints message and timestamp

Repeat

Project Timeline

* Project Plan (4/20) -  Both Lauren and Xin
* TCP client and server (5/4) - Lauren
  + Research how to implement TCP in Java (4/26)
  + Collect user input and validation (4/30)
  + Complete (5/2)
  + Testing and integration complete (5/4)
* UDP client and server (5/4) - Xin
  + Research how to implement UDP in Java (4/26)
  + Collect user input and validation (4/30)
  + Complete (5/2)
  + Testing and integration complete (5/4)
* Reflection (5/6) - Individual

Testing Plan

To test the programs, we will write test cases for good inputs, and as many types of bad inputs as we can think of. We will be looking to make sure that the program recovers gracefully from any bad input and to make sure that the output from the program matches the JAR files provided when good is provided. When problems arise, we will try to solve the errors individually. If we have significant trouble, we will reach out to our partner for help.