Name: REVANTH REDDY PAMULAPATI

Student ID: 1001717444

Server: Server is a Swing-based frame with a button and textareas to see the log messages and list of active clients. It is Multithreaded where new thread is created for each Client and listens on 3438 port by default.

Each client process will connect to the server over a socket connection and register a user name at the server which will be stored in an arraylist. Server displays the list of active users who are connected to the server.

Server logs all messages on its GUI.

Client: Client is a simple Swing-based frame with a text field for entering username, buttons to login & kill(logout), and textareas to see the conversation and to display the clock.

Each client will maintain a logical clock. This clock will be implemented as a counter that will increment once every second. The clock will be initialized with a random integer between 0 and 50.

Every nine seconds, each client will randomly choose one other client (e.g., a unicast) and send that client its present local time. The remote clock will be adjusted based on Lamport's Logical Clocks.

When a client sends a message, it will print the intended recipient of the message, as well as its present local time, to its respective GUI. When a client receives a message, it will print the sender's ID, the sender's local time, and whether its local clock needs to be adjusted according to Lamport's Logical Clocks.

If a clock adjustment is necessary, it will print the necessary adjustment, and its clock will continue forward with that value. If an adjustment is not necessary, it should print, "No adjustment necessary." Clients will continue to operate in such a fashion until manually killed by the user.

The messages exchanged between server and client uses HTTP formats and commands. On the server side, the unparsed HTTP message received from the clients are printed on the server screen. And on client screen, the HTTP message is parsed and only the message is printed.

Since the server is Multithreaded, it can handle multiple clients at a time.

Requirements:

- The programs are developed in Java (Version 8). The system must have Java 8 set up.

- IDE used is Eclipse Version 2018-12 (4.10.0).

- The Operating System used is Windows 10

How to Run the code:

1. Import the extracted project folder into the eclipse-workspace. (Eclipse -> File -> Import -> General -> Projects from Folder -> Next -> Set the directory path -> Finish)

2. Using the IDE, Run the 'Server.java' (Run as Java Application) once. And run the 'Client.java' (Run as Java Application) three times.

3. Using the Start button in the Server Window, start the server which will make the server to listen to the new client connections.

4. In the three Client Windows, Login with three different usernames using the Login button. After successful login, the local clocks are updated automatically based upon the Lamport's Logical Clocks with the help of the messages exchanged between the active clients.

5. Client can log out using the Kill button. And server can be stopped using the Stop button.

References:

https://www.dreamincode.net/forums/topic/259777-a-simple-chat-program-withclientserver-gui-optional/