Sameer Valiah Pusunuru
1001552473

DS Lab 2 Execution Instructions

Executing the project:

- This a java project and was developed using eclipse IDE and its best suited to be executed on Eclipse IDE.
- The project can be broadly divided into two parts Server and Client and has the packages with the same name in src folder, please import src folder to your eclipse IDE and run it as java application
- Each package of Server and Client has two java files ServerMain, ServerGUI and ClientMain, ClientGUI respectively.
- To execute the project the files should be exported into Eclipse as java project and run the ServerMain file first then ClientMain file
- You must run the ServerMain file only once and can run ClientMain file as many times as you want client connections to be established (run 3 times for 3 clients)

Execution process of the program:

Server End: -

- Once server starts running GUI will be open for server showing that its waiting for clients to be connected on port 7869 after this run client program
- The messages exchanged between client and server will be displayed in its text area.
- There is no input filed for server as server duty is only to show unparsed http messages and client joining and exiting the network in real time.
- You can terminate server by closing the GUI and if clients are present when terminating server their connections will also be terminated as soon as server is terminated.

Client End:

- As soon as the ClientMain is executed GUI will open and if server is on then it gets connected to the server no need to enter port as they are hard coded.
- In client GUI there will be two buttons on the right and left side of the client GUI one will be for one to one message and the other will be broadcast message and a text field at the bottom where you can enter msg.
- Client will assign a username automatically.
- Client will also assign it self a logical clock which will initiate from any number between 0 to 50.
- By default, client will send out a message to other client on the network in every 8 seconds.
- If you don't press the button after typing in the text box and press enter the message will be considered as a broadcast message.
- To send personal messages you have to type the message in the text field on the bottom of the screen and press the one to one, then on the left hand side top most corner of your screen a pop up will be opened with dropdown list consisting of all the active members in the network and then you can select the name to whom you want to send so the message will be delivered to him personally.
- To stop the client connection from the network any time you can enter "End" as the message in any case(lower or upper) and then client will be terminated from the network and its connection will be closed or optionally you can close the window of the client so that its connection will be closed from the network.

Sameer Valiah Pusunuru
1001552473

Communication: -

- Client and server communicate between each other via http
- Clients encodes the message and sends to server where it decodes the message reads it and performs the stated action and then encodes the message and send to specified client based on type of client request.
- Client will send the message to another client every 8 second pick them up randomly the message will contain the sender client logical clock
- Once the message is received by other client it will check the sender time with its receiver time and if the receiver time is less then the sender time then it adjusts its time to be one plus then sender time.

References:

The program has been written from scratch and may not find any code repeated from the references so the references are not commented in the code however project does gets inspired from the below links for specific functionalities implementations.

Client and Server architecture:
https://docs.oracle.com/javase/tutorial/networking/sockets/clientServer.html

Server Multithreading: -
https://github.com/abhi195/Chat-Server
GUI: -
https://www.youtube.com/watch?v=9gDErZCtdzM