IT 347 Project 1– Write up

Lehi Alcantara

1. Project description and architecture:

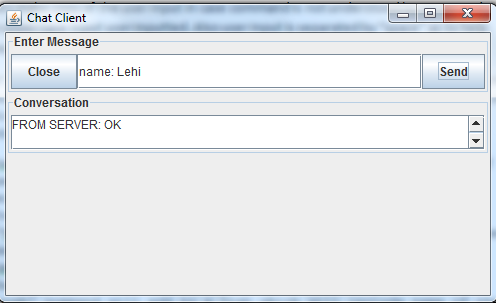
The way this project is structured is by having a server listening on port 9020. The way the server is set up is by running the server and talking to clients that are listening to port 9020 until user input a keyword that closes the socket connection, adios is the work.

Server setup:

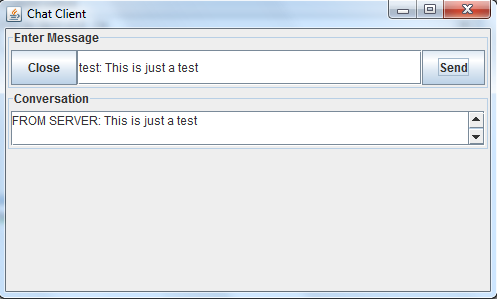
The server set up was done by using sockets, “BufferedReader” to read the client input, and “PrintWriter” to output the message to client. Server loops “forever” until user input the keyword “adios”. The first thing server does is to send and welcome message to client, it will only sent it once since I use a counter to make sure it only runs once. Then make copy of the user input in case command is not understood by server and displays it the upper case input user inputted. Also user input is separated by “space” as to help implement the commands.

The body of the loop consists of many different conditions where it checks for the user input, the following is a list of the checks, after this list I will indicate another list which will show the extra features I have implemented:

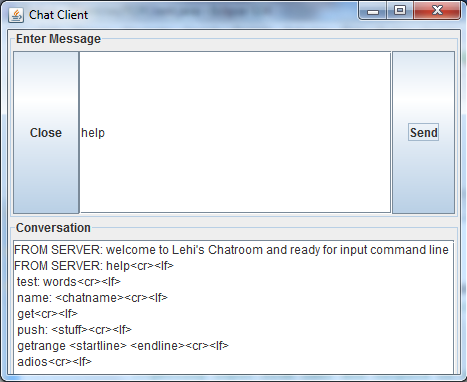
* 1. “name:” command essentially gets the name that user inputted if user inputs first and last name, code is able to handle it as well. Sends “OK” to client if everything goes well.



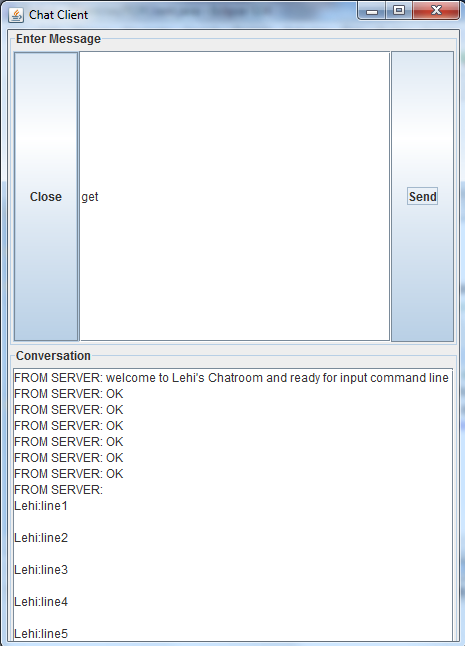
* 1. “test:” command will print out to client what user inputted.



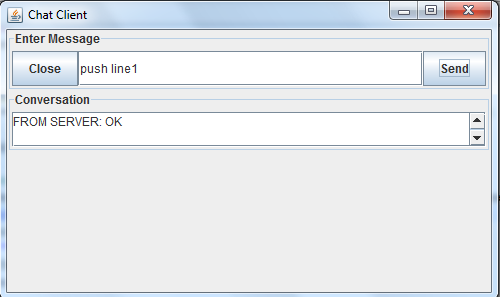
* 1. “help” command prints out to client a list of commands that user can refer to as to know how to use this chat server.



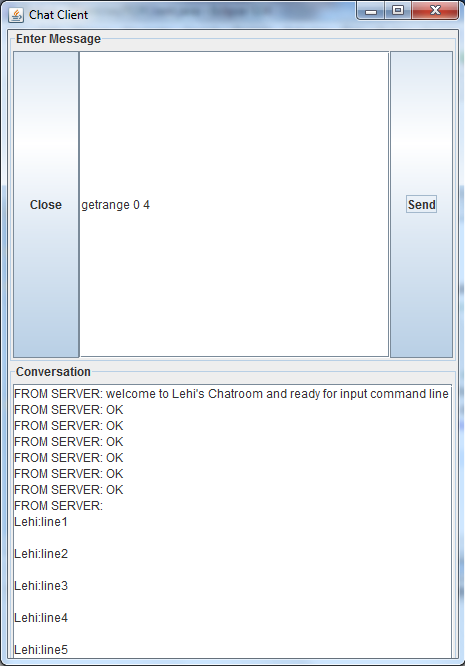
* 1. “get” command will print out a list, that is “pushed” by user. The data structure used is a list.



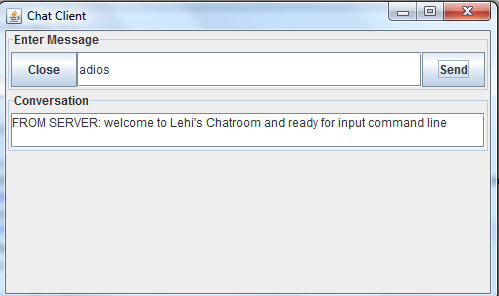
* 1. “push” command will add to a list which will include name of the user who pushed to the list.



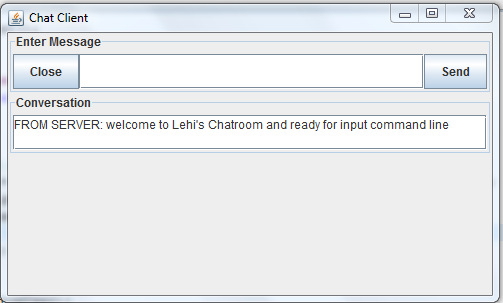
* 1. “getrange” command will take two inputs “startline” and “endline”. It will grab the list which user inputted some information in it. And then it will display user name and what the user inputted to server. using a copy of the first list with right range inputted by user.



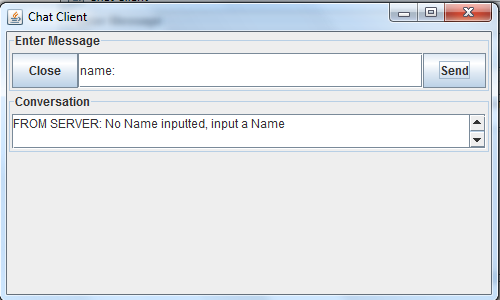
* 1. “adios” command will terminate the connection. By termination the socket.



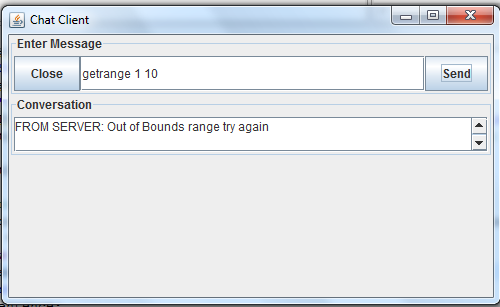
1. List of Extra features, for extra credit, that I have implemented on top of the original requirements:
   1. GUI: for the client user which contains a textField where user can input a command. A textArea where user can see what the server has replied. A send button to send to server and a close button to close the chat.



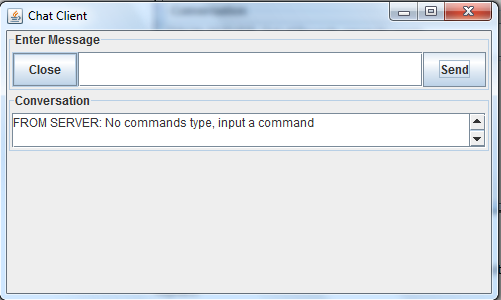
* 1. Name Check: if user starts name Command but does not finish and sent to server then server will reply: “No Name inputted, input a name”



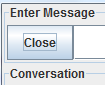
* 1. getrange Range Check: checks if the startline and endline inputted by user are valid, if not it will tell user: “Out of Bounds range try again”.



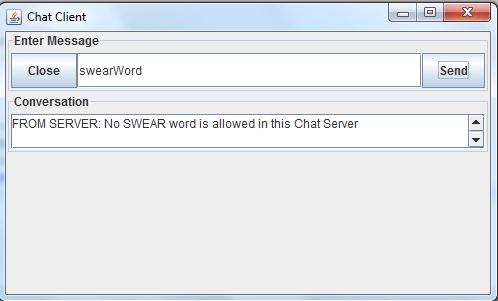
* 1. input Check: if user does not input anything and sends nothing to the server then server replies: “No commands type, input a command”



* 1. close button: when user clicks on close button then the chat server window closes as user wants to close the chat server.



* 1. “Swear word” Check: Checks if user inputted a swear word and lets user know not to use it while on the chat server: “No SWEAR word is allowed in this Chat Server”



1. Code structure and readability: The code has comments where I explain what each data structure does.
2. Source code is found for this project by using an index.html file under “Resource Files”.
3. Analysis of implementation issues is found on index.html under “issues”.