**LAB 3: ONLINE ADVISING SIMULATION USING MESSAGE QUEUING**

**Name**: Sameer Moses Murala

**Student ID:** sxm6494

**References:**

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

• https://www.geeksforgeeks.org/multi-threaded-chat-application-set-1/

• <https://www.geeksforgeeks.org/multi-threaded-chat-application-set-2/>

\* https://www.geeksforgeeks.org/remove-element-arraylist-java/

\* https://www.geeksforgeeks.org/randomly-select-items-from-a-list-in-java/

\* https://www.geeksforgeeks.org/generating-random-numbers-in-java/

\* https://www.javatpoint.com/java-string-to-int

\* https://stackoverflow.com/questions/1795808/and-and-or-in-if-statements

\* <https://stackoverflow.com/questions/24104313/how-do-i-make-a-delay-in-java>

\* <https://stackoverflow.com/questions/7190618/most-efficient-way-to-check-if-a-file-is-empty-in-> java-on-windows

\* https://stackoverflow.com/questions/4614227/how-to-add-a-new-line-of-text-to-an-existing-file-in-java

\* https://www.geeksforgeeks.org/java-appending-string-file/

\* https://stackoverflow.com/questions/14721397/checking-if-a-string-is-empty-or-null-in-java/14721414

\* https://stackoverflow.com/questions/6994518/how-to-delete-the-content-of-text-file-without-deleting-itself

**Execution Steps:**

* To start the Server compile and run the ServerGUI.java.
* Press the start button on the server to start the server. It display it is online.
* To start the Student Process compile and run the StudentProcess.java.
* Press the start button to connect to the server
* Enter the student name and course you want to register in the respective text fields and press the register button.
* To start the Advisor Process compile and run the StudentProcess.java.
* Press the start button to connect to the server
* It pulls the request messages from the Message Queuing Server and approves or disapproves the request by some particular probability.
* It displays the decision on the GUI.
* To start the Notification Process compile and run the StudentProcess.java.
* Press the start button to connect to the server
* It pulls the request messages from the Message Queuing Server and displays the advisor decision along with the student name and the course requested.
* To stop the respective process press the stop button.
* To stop the server press the stop button.

**Code Implementation:**

**Server Class:**

This class contains all the methods required for the server. The Methods are:

**Server Constructor:** It initializes the server GUI, port number required to start the server, the data time format for the HTTP encoding and the array list for all the client added to the server.

**Start Method:** It starts the server with the given port number. It starts an infinite loop where it keeps adding new client threads to the client list array list of the server class. If the stop button is pressed in the GUI it breaks the infinite loop and closes all the server socket and closes the thread sockets.

**Stop Method:** It changes the Boolean value keepGoing to false so that the infinite loop in the start method breaks.

**ActiveClients Method:**  It displays a list of all the active clients in the Client Thread Array List and concatenates to a string and calls the append method to display on the server GUI.

**Display Method:** Display all the events on the events occurring in the server on the server GUI along with the date time.

**BroadCast Method:** This method is used to broadcast message to every active client. It runs a for loop to get all the clients from the array list and calls the writemessage to display it on the client GUI.

**SingleMessage Method: :** This method is used to broadcast message to the selected active client. It runs a for loop to get all the clients from the array list and uses an if condition to find the selected clients and calls the writemessage to display it on the client GUI.

**Remove Method:** If a client presses the logoff button on the client GUI it runs a for loop to find the selected client and calls the remove method of the ClientThread class to remove the selected client from the array list.

**Main Method:** It gives the default port number. It starts the server with the default port number.

**ClientThread Class:**

**ClientThread Constructor:** There is a id that is incremented every time so that an unique id is assigned to every client that is added to the clientthread class. A new object input and output stream is created. The username for the client is taken from the client gui and added to client class.

**Run Method:** An infinite loop runs that takes the message from the client GUI and based on the type of the message received a swithcase is created for implementing the various actions. The type of the message changes based on the input coming from the Client GUI. If it is a normal message it is just displayed. If of type logout then Boolean keepgoing is to false. If type whoisin then all the clients in the client array list are sent as an string message to the server GUI. If type clientnamelist then all the clients in the client array list are sent as an string message to the client GUI. If it is type singleclient single message is called.

**Close Method:** Method to close the client thread object.

**Client Class:**

**Client Constructor:** It initializes the server object, port number, username taken from the user and client GUI object.

**Start Method:** It requests for connection from server, else it shows an exception. If connection is accepted from the server it displays the event on the client GUI. Input and Ouput streams are created. A class object is created to listen for messages from the server. Send the username taken from the client GUI to the server to add it to the client array list.

**Display Method:** Method to display the message on the GUI.

**SendMessage Method:** A chat message is sent to the server. Based on the type of message sent different encoding is done. If it is a broadcast message then encode method id called to encode it in the HTTP format and sent to the server. If it is a singleclient message then encode method id called to encode it in the HTTP format and sent to the server. Else a normal message is sent to the server without encoding.

**Encode Method:** In this method the HTTP format encoding is done. The message taken from the client GUI is concatenated a string that contains all the required fields like POST, Host, User Agent, Content Length etc.

**Disconnect Method:**  It closes the input and output streams and closes the socket whenever logout button is pressed in the client GUI.

**Main Method:** The default port number is taken and the client constructor is called.

**ListenFromServer Class:**

**Run Method:** It runs an infinite loop where it listens for any message from the server. It creates an input stream and if it contains clientname it populates the client GUI combo list, else it decodes the message received from the server and displays it on the client GUI.

**Decode Method:**  It takes the HTTP encoded message which is nothing but a string and then splits the string based and some token and extracts the message from the string and returns the decoded message.

**ChatMessage Class:**

This class is used to define the type of messages that are sent between the client and server and decide the actions based on the type.

**ChatMessage Constructor:** It initializes the message type and the message.

**GetType Method:** Getter for the type.

**GetMessage Method:** Getter for the message.

**ServerGUI Class:**

**appendRoom method:** Display the messages in the HTTP format in a TextaArea.

**appendEvent method:** Display the events occurred server side in a TextaArea.

**appendClient method:** Display the active client in a TextaArea .

**stopStartActionPerformed method:** If the start button is visible then start the server with the default port number, else stop the server by calling server stop method.

**formWindowClosing Method:** Closes the server GUi window.

**activeclientsbuttonActionPerformed method** : It displays all the active clients connected to the server by calling the activeclients method.

**ServerRunning Class:**

**Run Method:** Run the server until the stop button is pressed. Once stop button is pressed set start button to stop and server equal to null.

**ClientGUI Class:**

**logoutActionPerformed Method:** Remove Client from the ClientList. It sends a chatmessage to the server of the type LOGOUT.

**loginActionPerformed Method:** It creates a new client with username from the user and connects to the server .Adds client to the clientlist.

**broadcastActionPerformed Method:** Sends a broadcast message to all the active clients by takin the text from the GUI and sending a MESSAGE type message.

**refreshActionPerformed Method:** Requests and displays all the active clients from the server.

**sendActionPerformed Method :**  Sends the message to selected client from the JtextArea and message from the GUI with type SINGLECLIENT message.

**Append Method:** Displays the events on the client GUI.

**connectionFailed Method:** If the connection fails the client GUI buttons are reset.

**Run Method:** It sets the Client GUI window to true.

**Links :**

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

• https://www.geeksforgeeks.org/multi-threaded-chat-application-set-1/

• <https://www.geeksforgeeks.org/multi-threaded-chat-application-set-2/>

\* https://www.geeksforgeeks.org/remove-element-arraylist-java/

\* https://www.geeksforgeeks.org/randomly-select-items-from-a-list-in-java/

\* https://www.geeksforgeeks.org/generating-random-numbers-in-java/

\* https://www.javatpoint.com/java-string-to-int

\* https://stackoverflow.com/questions/1795808/and-and-or-in-if-statements

\* <https://stackoverflow.com/questions/24104313/how-do-i-make-a-delay-in-java>

\* <https://stackoverflow.com/questions/7190618/most-efficient-way-to-check-if-a-file-is-empty-in-> java-on-windows

\* https://stackoverflow.com/questions/4614227/how-to-add-a-new-line-of-text-to-an-existing-file-in-java

\* https://www.geeksforgeeks.org/java-appending-string-file/

\* https://stackoverflow.com/questions/14721397/checking-if-a-string-is-empty-or-null-in-java/14721414

\* https://stackoverflow.com/questions/6994518/how-to-delete-the-content-of-text-file-without-deleting-itself