MEMO

Project 1 - Chat Central Server and Client

Team Members:

**SHARANYA RAMIDI - sr2876@nau.edu (6226508)**

**YOSHITA BARAMPURAM - yb288@nau.edu (6201104)**

**GOWTAM SAI SANKAR GOTTAM - gg694@nau.edu (6203501)**

**Introductory Overview:**

The classes present are:

Server

Client

ChatMessage

Server:

Methods:

Start( ): Creates a socket server and waits for connections as long as the server is active.

Client Thread is created to add clients which are being added to the server.

Try and catch blocks are present to handle exceptions while trying to stop server.

// method done by Gowtam

Stop( ): Validating the keepGoing and use try catch to stop server connection.

//method done my Gowtam

Display( ): Display event to the console.

//method done by sharanya

sendMessage( ): This method is used to send messages across sever and clients connected to the server.

//method done by Sharanya

Remove( ): When the client uses LEAVE the remove method finds the client id

& disconnects the client from the server & sends message to other clients as

well that a Client has left.

//method done by Yoshita

Close( ): Closes the socket connection and stream i/p , o/p.

//method done by Sharanya

writeMsg( ) : A string message is sent if the client is still connected.

If any error occurs this doesn’t abort the server and client connection but

informs user. This is done using try and catch block.

//method done by Yoshita

serverClose( ): Close the server connection using socket with parameters

as localhost and port.

//method done by Sharanya

clientClose( ): The for loop is used to traverse the array list of clients and close()

is used to close all the clients connected to the server.

//method done by Sharanya

Client: The client constructor is set to server, port, username.

Methods:

Start( ): The start method uses socket connection with parameters server and port

to establish a server connection , if it fails catch block handles the exception

and displays Error message and a message connection accepted if its

established successfully. It also creates a two way data stream to input and

output data from server. ListenFromServer creates a thread and start() is used to

type our username to join.

//method done by Yoshita

sendMessage( ) : Sends message to server using stream output object.

//method done by Gowtam

Disconnect( ): Close the input and output streams.

//method done by Yoshita

The client main method uses infinite loop to get input from user and executes

the message type LEAVE, SHUTDOWN, SHUTDOWNALL depending on

the user input.

The class ListenFromServer extends thread class and runs method run() to read

the input message from server.

//method done by Sharanya, Yoshita, Gowtam

ChatMessage:

The class ChatMessage implements Serializable interface so that an object can be serialized and convert its state back to copy of that object.

It contains message types

JOIN, LEAVE, MESSAGE, SHUTDOWN, SHUDOWNALL.

//method done by Sharanya, Yoshita, Gowtam