

## **COMP 713**

### ***Distributed and Mobile System***

#### ***Assignment 1***

### **Documentation and Demonstration**

The project is a sample interface of a chatroom where the clients will be able to connect to a server and register their usernames then go into the chatroom where other clients will be in the chatroom as they will be able to have some conversations with each other.

There will be five Java classes: server, client, message, server GUI and client GUI.

#### ***The components in the project are:***

- JButton
- JPanel
- JTextField
- JTextArea
- JFrame
- JLabel
- JStrollPane
- BorderLayout
- Dimension
- Toolkit
- ActionEvent
- ActionListener
- ObjectInputStream
- ObjectOutputStream
- Server Socket
- Socket
- SimpleDateFormat
- Array List
- Date
- EmptyBorder
- GridLayout

#### ***The features of the project are:***

- Object-Oriented
- Distributed
- Threads
- Broadcast
- GUIs

## The details of the client/server protocol in the project

### **public class Server**

- *public Server(int port, ServerGUI serverGUI)*
  - The public class to use for the server GUI and running the server program
- *public void start()*
  - Start the Socket, ClientThread and ServerSocket for the server
- *protected void stop()*
  - Stop the Socket, ClientThread and ServerSocket for the server
- *private synchronized void broadcast(String message)*
  - Broadcasting for the clients to send messages using ClientThread and ArrayList
- *synchronized void remove(int id)*
  - Remove the clients using ArrayList and ClientThread

### **public class ClientThread extends Thread**

- *public ClientThread(Socket socket)*
  - ObjectOutputStream(socket.getOutputStream()) will be created
  - ObjectInputStream(socket.getInputStream()) will be created
  - Socket will be created
  - (String) socketInput.readObject() will read the username enter
  - Date will be created
- *public void run()*
  - BroadcastMessage will be created for the server's IP address
  - (Message) socketInput.readObject() start read
  - Date is formatted and start to use
  - Switch Cases will be activated for message and to broadcast: switch(msg.getType()) as when clients decided to log out, log in or selected to check if user in so broadcast will be notified, and notify everyone.
- *private void close()*
  - use for closing all socket, socketOutput, socketInput connections.
- *private boolean writeMsg(String msg)*
  - socketOutput.writeObject(msg) where will write output the messages to broadcast.

### **public class Client**

- *public Client(String server, int port, String username, ClientGUI clientGUI)*
  - The public class to use for clients to connect and for client GUI
- *public boolean start()*
  - Start the Socket, ObjectInputStream, ObjectOutputStream and ServerListener
- *private void disconnect()*
  - which create the disconnection for all inputSocketStream, outputSocketStream, socket and notify to clientGUI that the connection is failed

### **public class ServerListener extends Thread**

- *public void run()*
  - Threads for server listener will be running and messages will be activation and send to the JTextArea of Client GUI

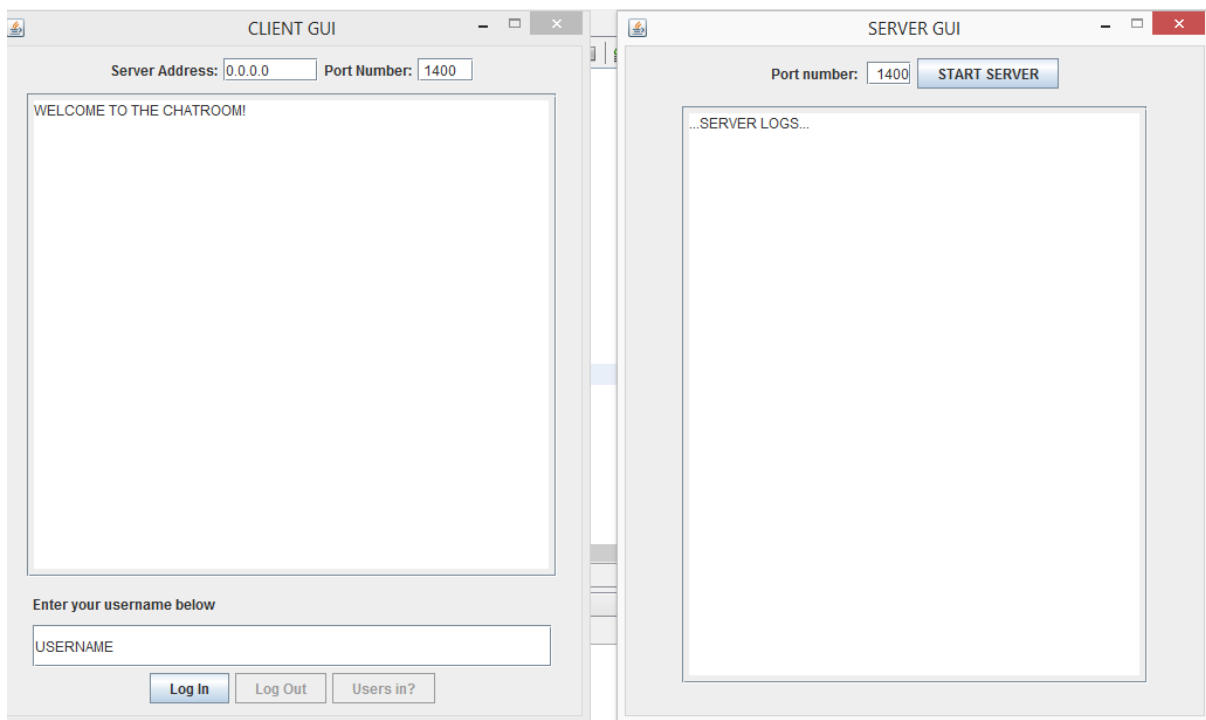
## The user documentation and the demonstration of the system

In the project, there will be two GUIs, one Client and one Server.

When you run both programs, it will be demonstrated the two separate GUIs.

The Server Address and Port Number in Client GUI, will be showed the default numbers, where you can change them into the number you want to use.

The port number in Server GUI, will be showed the same default port number as Client default port number.



When you click start server, it will be showed the port number and commanding the waiting for the clients to connect.

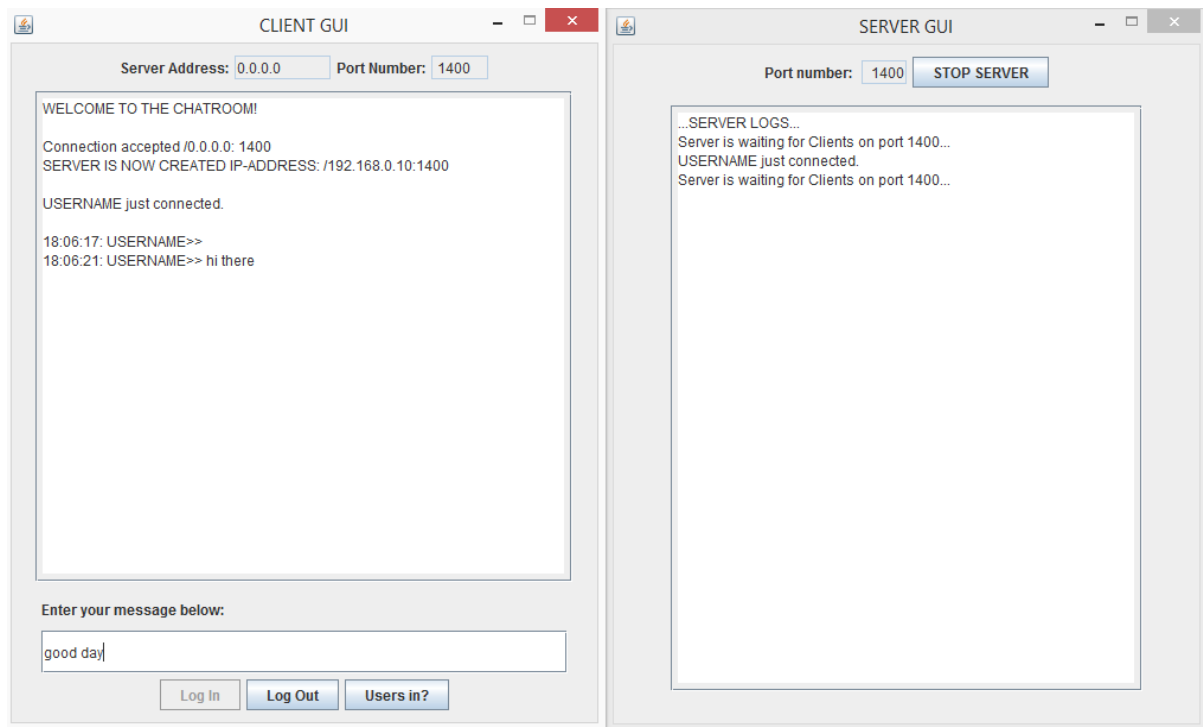
Then you can use the Client GUI, click “log in” after you typed in the username or use the default username. The server IP address will be broadcasted into the Client GUI’s JTextArea so other computer will know what the server’s IP address is so they will be able to connect with each other, using the same IP address of one known server.

The message of “Enter your username below” will be changed into “Enter your message below” so you can type in the messages you want to say but you need to use the keyboard “Enter” to send the messages.

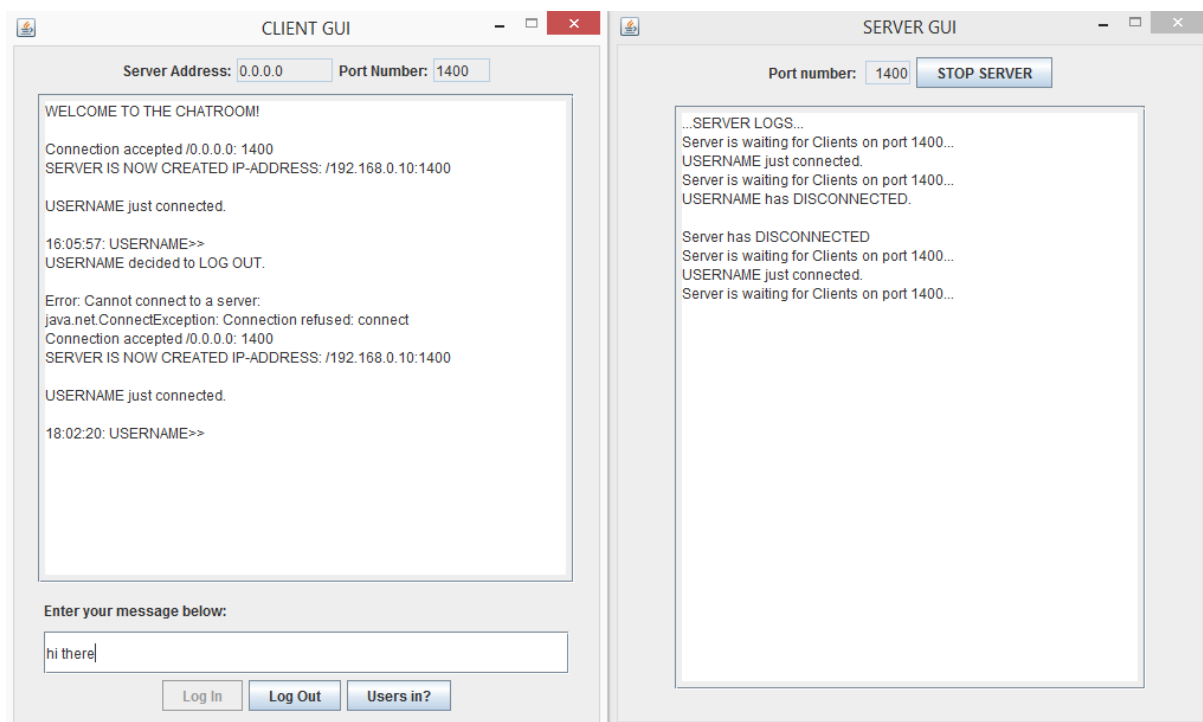
GABRIELA ORELLANA  
STUDENT ID NO.: 1244821

The messages will be broadcasted to other clients naturally after you enter the messages.

There will be date and time that displayed right beside the messages where the others can see the time.

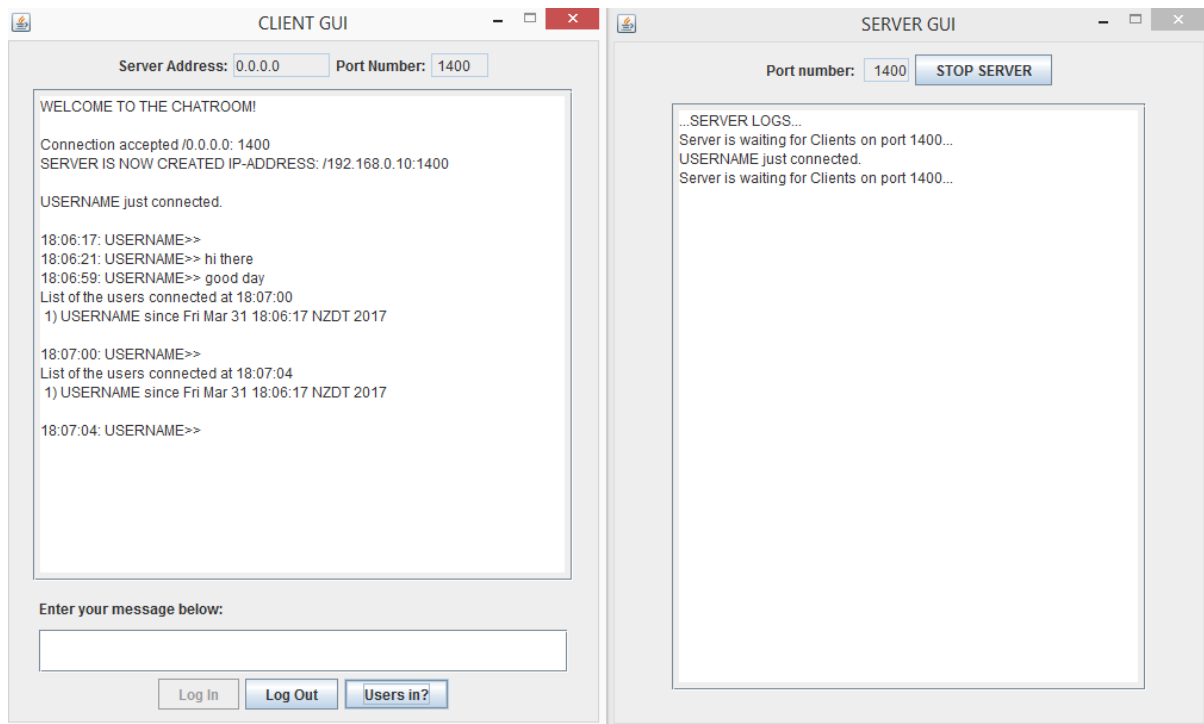


Also, when clients clicked “log in” then the clients who are already in chatroom will be able to see the username that just logging in by broadcasting.



GABRIELA ORELLANA  
STUDENT ID NO.: 1244821

When you click the “User In?”, that will show you the users that are in the chatroom or not.



When you log out, you will be seen by clients where the broadcast will state that you are logging out. Also, you will be removed from the lists where your username will be not showed when you click “User in?”

