# Design Document

A Distributed Chatting System

Designed by: Zhang Tianhui (student number: 16206789)

## Overview:

This distributed chat system was developed by the Java Language, containing five classes.

- ChatMessage.java

  This class defines the different type of messages that will be exchanged between the Clients and the Server. It would be better to use an object to store the information to be passed, because when talking from a Java Client to a Java Server a lot easier to pass Java objects, no need to count bytes or to wait for a line feed at the end of the frame.

  Message type(int) : LIST, BROADCAST, STOP, KICK, STATS

- Server.java

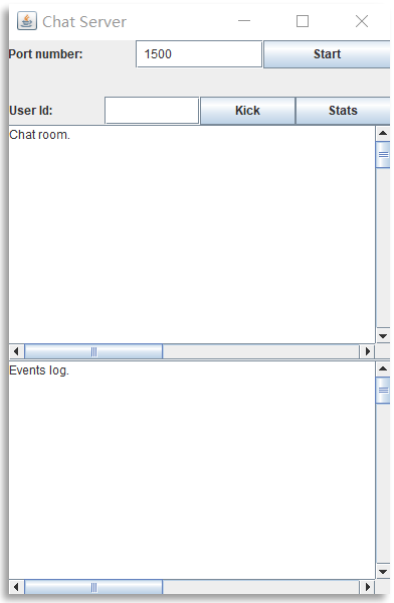
  It is used to create a server object that has specific port number entered by the user. Almost all information is identified and handled correctly in the server class.

  The main variable stored in a server is a list - ArrayList<ClientThread>, which can manage all the ClientThread objects connected to the server.

  To store the client information in this server, create a class called ClientThread, which can keep the socket, username, id, command list of a client (for the STATS function). And though the ClientThread object, we can create Input and Output Streams. So that it is able for the server to accept commands from clients through the ClientThread.

- Client.java

  The source object of ClientThread object. The main function is to create a client according to the specific port number and connect it to the server. Then the client can use socket and streams to pass the commands to the server waiting to be handled.




➢ ServerGUI.java

Create a window for the server program. The north area of this window is inputs and buttons. ServerGUI get the port number entered and create specific server object. After clients connecting, use the second input text field to get user's ID to be kicked or stats.

The center area of this window is two text area. The above one is the message area, which can display the message is broadcast. The below one is the area for protected message which can be seen by manager like STATS.

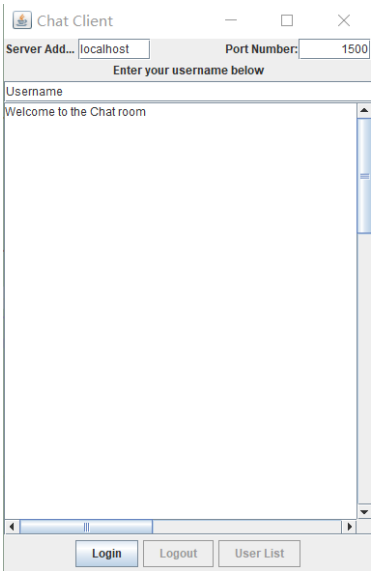
➢ ClientGUI.java

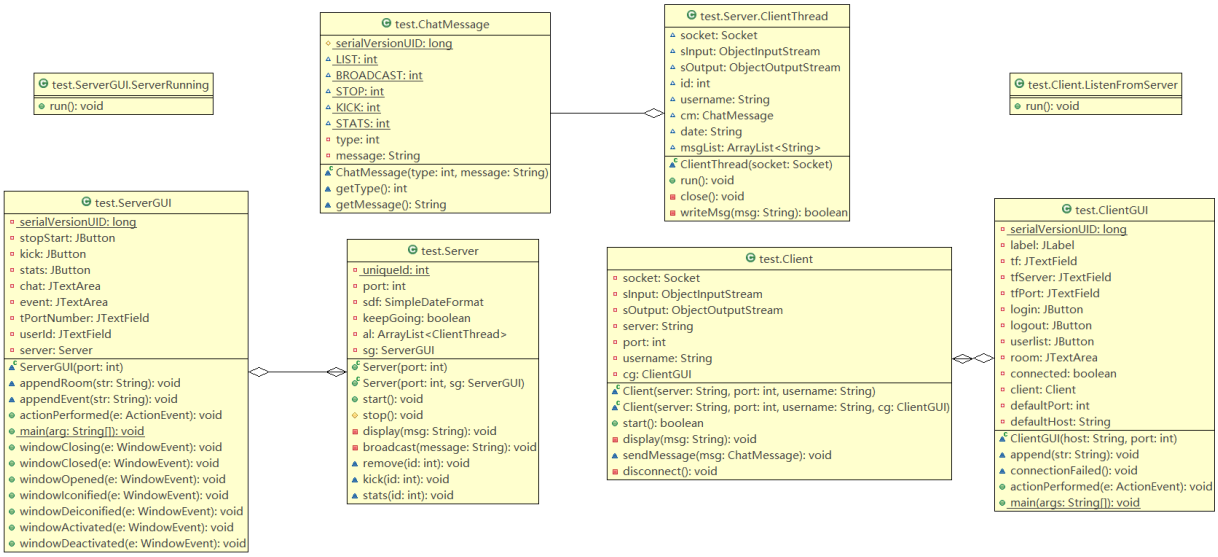
Create a window foe the client program. According to the above inputs, create a client object connected to the server with unique address and port number.

There is only one input text field. After enter the user's username firstly, the prompt will change to ask you to enter your message.

Below the input field, there is a text area that receives broadcast messages or result of online users checking.

In the south area of this window, there are three buttons. The Logout and User List buttons will get clickable after the user log in. Meanwhile, the Login button will be not clickable.



## Functions:

- ✓ The server will be able to support multiple client connections.
- ✓ Each time a client connects to the server the event is announced to all connected clients.
- ✓ The client can send and receive information from the server. Because of the realization of GUIs, the specific command format is not needed anymore.
    i. BROADCAST - {content}}: enables a client to send text to all the other clients connected to the server.
    ii. {STOP}: forces the server to close the connection with the client that initiated the command, this event must be announced to all other clients.
    iii. {LIST}: displays a list of all client IDs currently connected to the server.
    iv. {KICK - ID}: closes the connection between the server and the IP client, and announces this to all clients.
    v. {STATS - ID}: gets a list of all commands used by the client identified by the ID.
- ✓ Provide a simple Graphical User Interface to allow a user access the system.
- ✓ Code with detailed comments.

## Characteristics of technology:

- Serialization:
  When two processes are communicating remotely, they can send various types of data to each other. Whatever the type of data, it will be transmitted on the network as a binary sequence. The sender needs to convert the Java object into a sequence of bytes to be transmitted over the network. The receiver needs to restore the sequence of bytes to the Java object. Therefore, some objects need to be serialized so that they leave the memory space and live on the physical hard drive for long-term storage.

- Synchronized:
  Java language keywords that can be used to lock objects and methods or blocks of code, when it locks a method or block of code, at most one thread at a time executes the code. When two concurrent threads access the lock synchronization code block in the same object, only one thread can be executed in one time. Another thread must wait until the current thread has executed the block of code. However, when one thread accesses a lock block of an object, another thread can still access the unlocked block of that object.

- After actions like start or stop, set the buttons and cursor state:
  requestFocus(): Focus the cursor on the suitable position that calls this method.
  AbstractButton.setEnabled(boolean b): set the JButton clickable state.

## Drawbacks:

- The scroll area displays the first row by default, and is not synchronized with the location of the latest message.
- The default value of the text input box is easily mistyped into the user name.