## Abstract

This chatting system mainly uses the socket, threading and GUI design in java programming, which consists of the server and client. It allows multiple clients to chat at the same time through the network. The server of the system has several functions like broadcast the message to other users, in response to the connect/disconnect request of users. And the client allows user to logon, disconnect to the server, send message to other clients and receive message from other clients.

## Features of the chat system

From the design requirement of this chatting system, it can be see that the chatting room is an integrated application of socket, GUI and thread in java. This system implements basic functions of the assignment requirements, includes:

1, start server: connect to the network

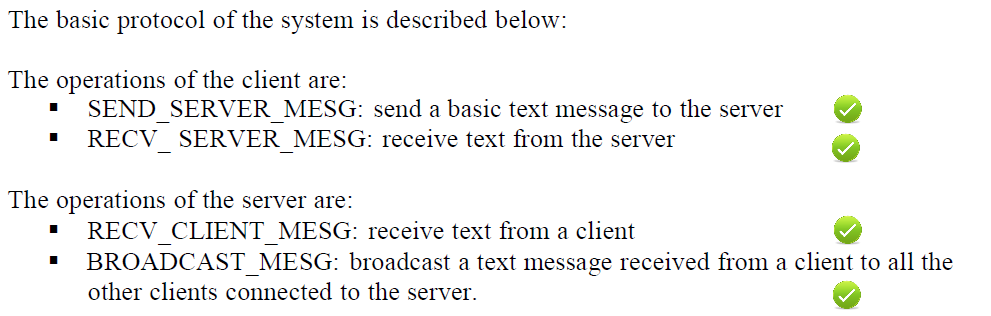
2, login interface: input the user name to login to the system

3, send message (includes private message between users): send message to all the users (public chat) or send message to a selected user (private chat)

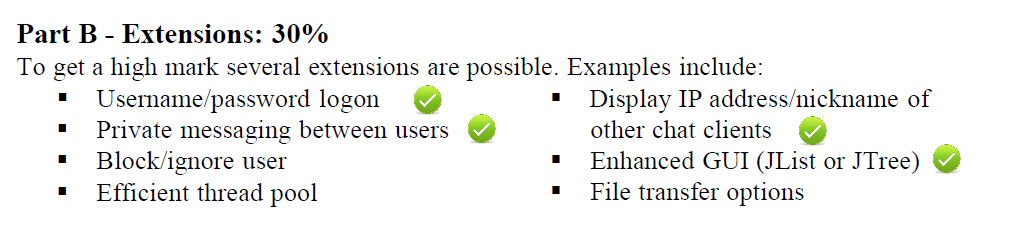
4, receive message: receive message from the users who connect to the server

5, participants list: show other users’ name and IP

## The completion of the basic requirement(70%)：



## The completion of the extensions(30%)：

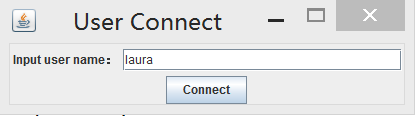


## Source code files

The whole system includes 3 separated java source code files: LoginFrame.java, ClientFrame.java and Server.java.

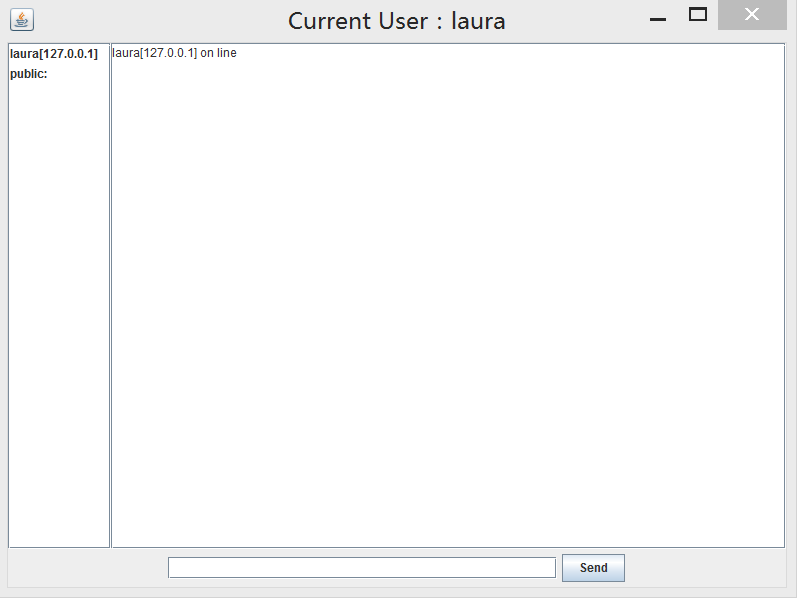
**1,LoginFrame.java:** The login interface of the client, mainly uses the java.awt package and the javax.swing package, which Includes the components of the login window and the layout of each component.

Figure 1:Screenshot of the login window



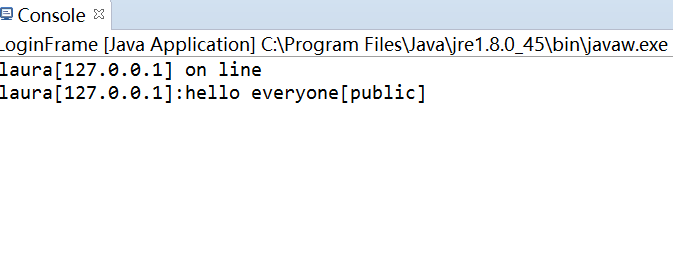
**2,ClientFrame.java:** defines the client interface and the listeners of some components, and implements the connect, send and receive function of the client, as well as the receive thread of the system.

Figure 2: screenshot of the client window



**3,Server.java:** includes the server start/close function, and the server thread which implements the Runnable to handle each client.

Figure 3: screenshot of the server console



## Schematic diagram and flowchart

Figure 4: Schematic diagram of the system

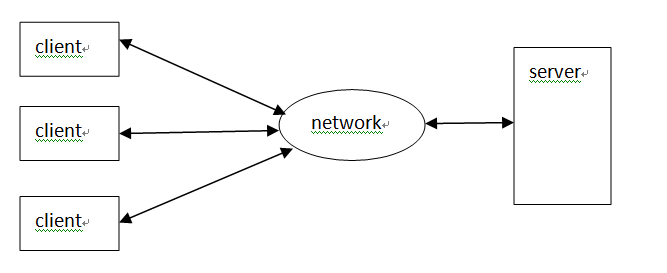
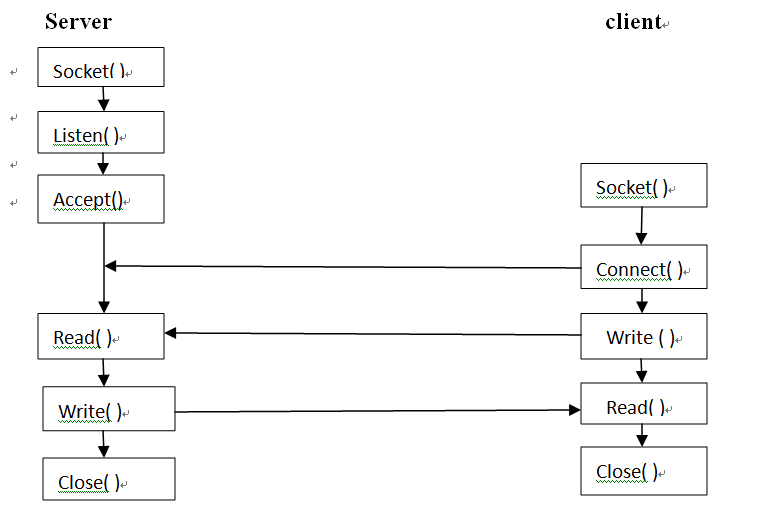
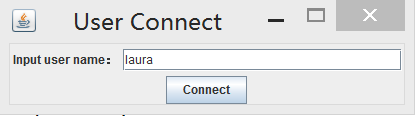


Figure 5: Flowchart of the communication of the server and the clients

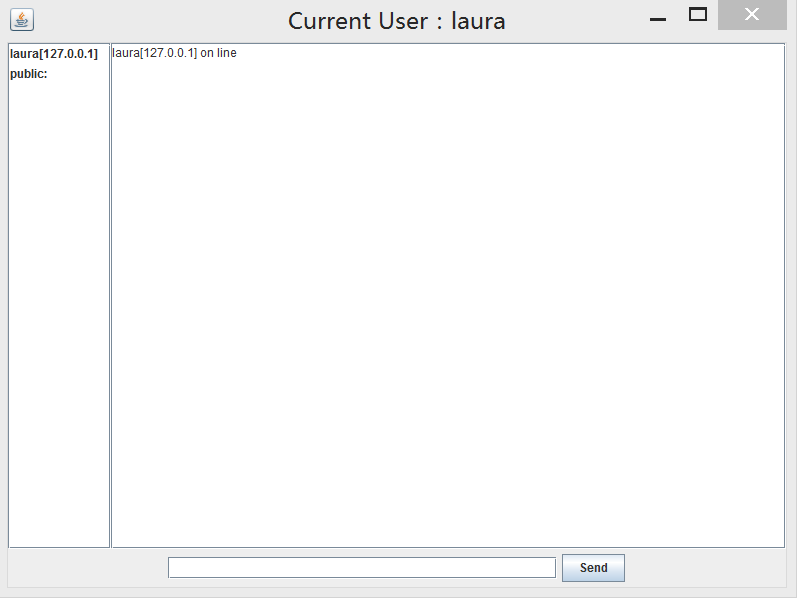


## Usage of the program

1, run the LoginFrame.java, then you will see the following window, input your user name and click the “connect” button to connect to the server.



2, after you successfully connected to the server, you will go to the client window, which shows as follow:



3, you can start chatting with other users, and when you send a message, you can choose to send it in public, which means every user will see your message, or you can send it to a selected user in private by clicking the user name in the participants list on the left. The default setting of the system is sending a message in public.

