



Java Chat Application – Assignment 1

Java Chat Application

SANOOB THEKKE VALAPPIL

213568654 07/05/2014

Contents

Contents

Introduction.....	2
How to run?	2
Server Side Code- ChatServer.java	3
Server Side GUI-ChatServer.java	3
Client Side Code-ChatClient.java	4
Client Side GUI.....	5
1. Loginform.java – Login Form for the Client.	5
2. Register.java- To register a new user.....	6
3. ClientChat.java- The Client GUI of the Chat Application.....	7
Broadcast Messaging.	7
Private Messaging.	8
Program in execution.	9
Notifications in GUI.	9
Notification on the server side.	9
Notification on client Side.....	9
Statistics Function at Server Side.....	9
Conclusion	10
Reference.....	10

Introduction

This is a simple java Chat application. It's a standalone application consisting of two components, Server component and client component. The server and client component is a Java Swing application. The server application runs on the server side and the client runs on the client side. This application can be used for both broadcast chatting and private chatting. Key Features of this chat application are.

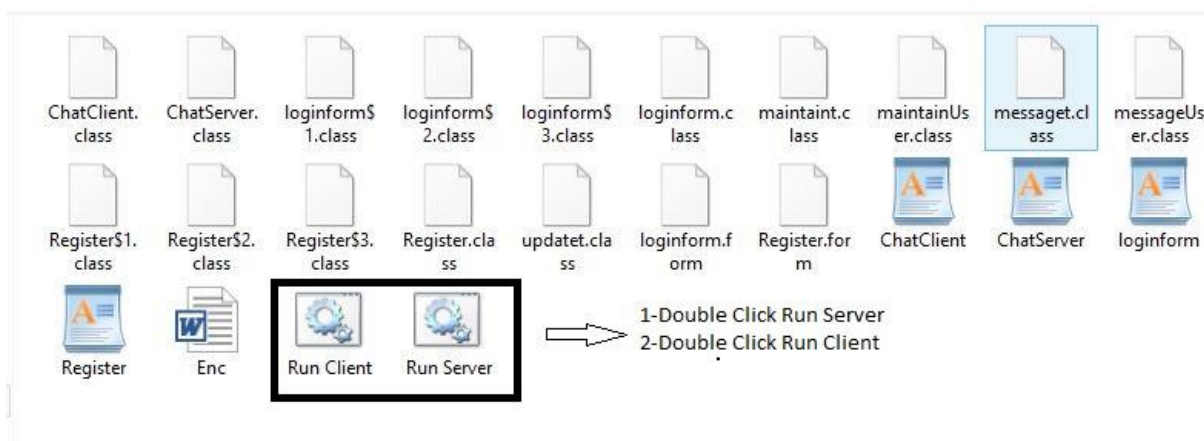
- Broadcast Messaging.
- Private Messaging.
- Client login.
- New Client Registration.
- Display List of Online users
- Display the parameters of the server line total number of registered users, online users, number of broadcast messages sent. Number of private messages sent etc.,

How to run?

Note: You need to run the Server side code first (Run Server.bat) for the Chat Application to work

To run this application we need to run two files, one for the server and one for the client. to run this application

1. Double click Run Server.bat to run the server side of the code.(Create one server)
2. Double click Run Client.bat to create one new user, can be run multiple times if there is a need to create more Clients



To login in to the client side you need a valid **username** and **password**, Otherwise the server will not accept the connection. We use a file named Enc.doc to store all the login credentials. Some of the Login details are given below to run this application

No	Username	Password
1	user1	user1
2	user2	user2
3	user3	user3

Server Side Code- ChatServer.java

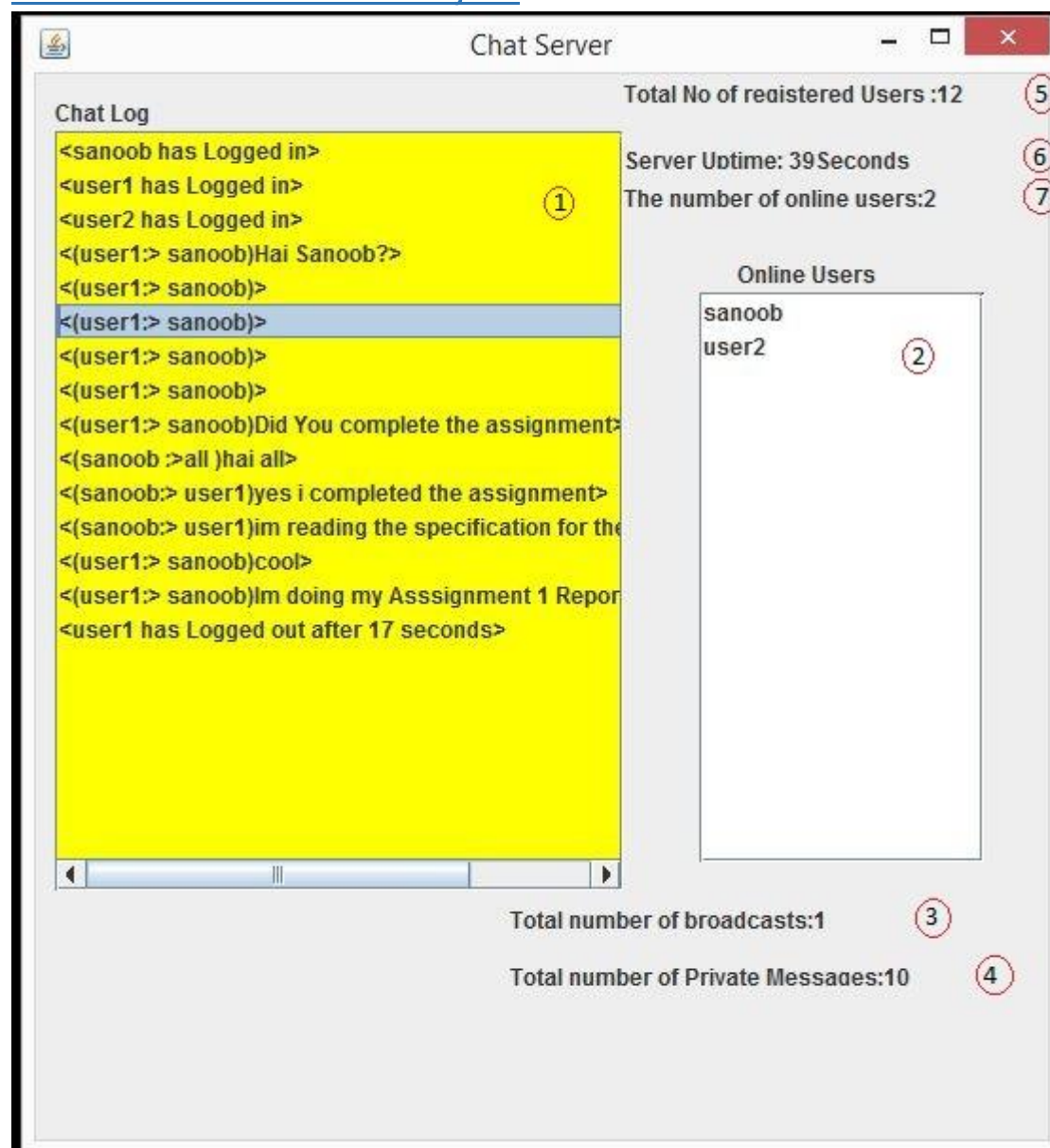
In this java file we create a server to listen for incoming request at sockets. We make use of three Java classes which will run as threads. The three JAVA classes are.

Maintaint – To maintain the list of active user names that are currently logged in on the chat application. A thread is created for each client for this purpose.

Updatet – To update the active online user names, in case some user's logout or login.

Message – To handle the message requests and response including broadcast and private chatting.

Server Side GUI-ChatServer.java



No	Description
1	The Chat log, Stores the chat between the clients
2	Online Users, Shows the list of online users using the Chat Application
3	Total Number of Broadcast done in this Session
4	Total Number of Private Message done in this Session
5	Total number of registered Users for the Chat Application
6	The server run time in seconds.
7	Active users Currently using the chat Application

The GUI for the server will only be visible when a client connects to a server.

Client Side Code-ChatClient.java

In this application we can create multiple clients, to simulate a chat environment. There are four additional java files that are used for the successful execution of the client side program.

1. **Loginform** - this is basically a client login form having a username field, password field client authentication is done in this java file using file. The value from the Textfeild is matched against the value in the document file. If success the chat client is launched. And otherwise an error message is shown.
2. **Register** -This is a form for registering new users, if a new user is created, the username and password is stored in to the file. During registration there is a check done to see whether there is already an existing user of the same name. If so an error message is returned. Also if the password given in both password and retype password field are not same it again returns an error. On successful creation of a client, the Login page is displayed for the client to login.
3. **ChatClient** -This provide the GUI for the client side of the chat application .In this form user can send Broadcast Messages and Private Messages. Also provides a list of online Users for Private Chatting. To send a broadcast message select the broadcast button and select private message to send a Private message
4. **maintainUsers** -This class is used to maintain the list of online users and update the same list if a new client is connected or an existing client is disconnected. An ArrayList is used for this purpose. Each time a Client is created their name is stored in the ArrayList, and later these values are binded to the GUI JList
5. **messageUsers** -This class is responsible for sending and receiving messages to and from the client .it handles both broadcast and private messaging. When a message is received from the server or another client. This class breaks the message down in to components. And determine whether it's a broadcast, Private message or a Notification

Client Side GUI

1. Loginform.java – Login Form for the Client.

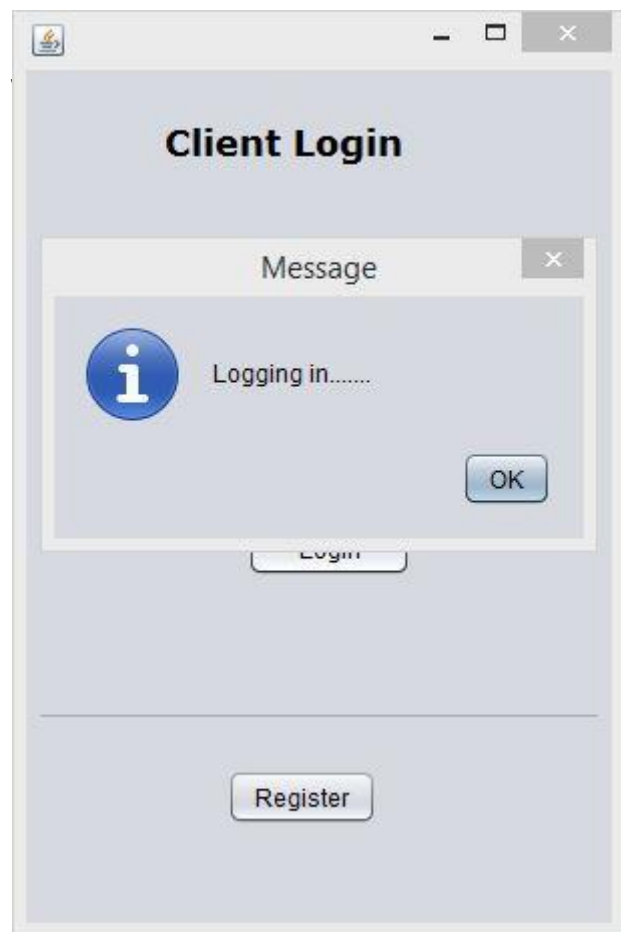


The diagram shows a window titled "Client Login" with a light blue background. It contains two text input fields: "UserName" and "Password". Below the "Password" field is a "Login" button. At the bottom of the window is a "Register" button. To the right of the window, four arrows point to the corresponding elements with the following text:

- Enter User Name Here (points to the UserName field)
- Enter Password Here (points to the Password field)
- Click Button to login (points to the Login button)
- Click Button to Register a new User (points to the Register button)




Client Login Failure



Client Login Success

2. Register.java- To register a new user.



The diagram shows a 'Register Form' window with the following components and annotations:

- User Name:** A text input field with an annotation 'Enter New User Name' and a right-pointing arrow.
- Password:** A text input field with an annotation 'Enter Password' and a right-pointing arrow.
- Retype Password:** A text input field with an annotation 'Retype above password' and a right-pointing arrow.
- Register:** A button with an annotation 'Click button to Register' and a right-pointing arrow.
- Login:** A button with an annotation 'Click Button to redirect to login page' and a right-pointing arrow.



The 'Register Form' window shows the 'User Name' field filled with 'user2'. A 'Message' dialog box is displayed over the form, containing an information icon and the text 'User Already Exits!!!' (sic). The 'OK' button is visible in the dialog box.

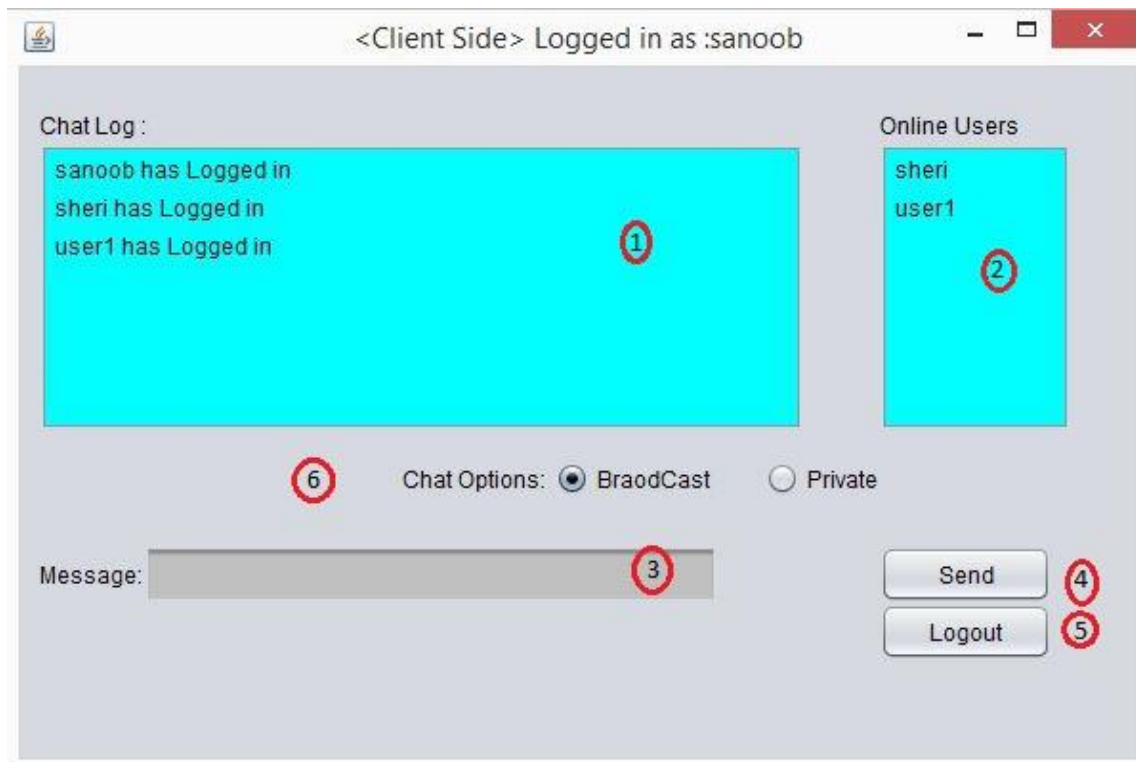
Registering using an existing user



The 'Register Form' window shows the 'User Name' field filled with 'user1', and both the 'Password' and 'Retype Password' fields filled with '*****'. A 'Message' dialog box is displayed over the form, containing an information icon and the text 'User Created Succsessfully' (sic). The 'OK' button is visible in the dialog box.

Registering a new User

[3. ClientChat.java- The Client GUI of the Chat Application](#)

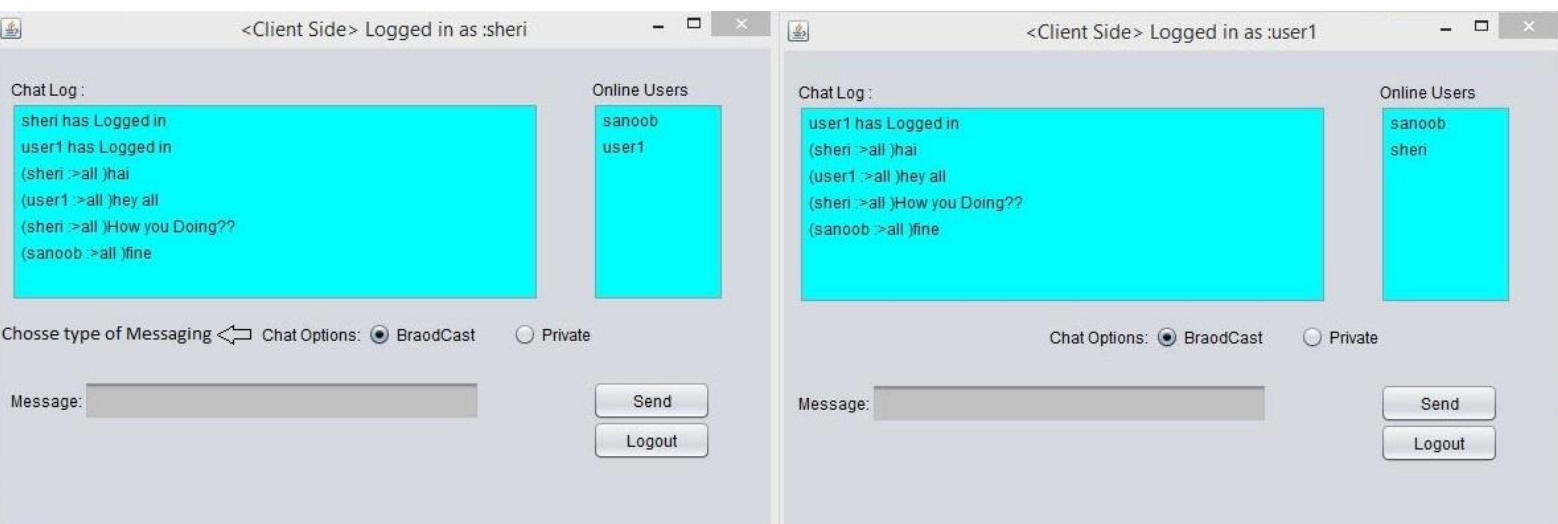


No	Description
1	The Chat log, It's a Window where all the chat appears including Private message
2	List of users online
3	The field to input your message
4	Button to send the message to the server
5	Button for the Current User to Logout
6	Chat options to choose between Private and Broadcast (Radio Button Group)

[Broadcast Messaging.](#)

One of the features in the chat application is that it provides broadcast message services. ie it sends message to all connected clients. To do a broadcast messaging.

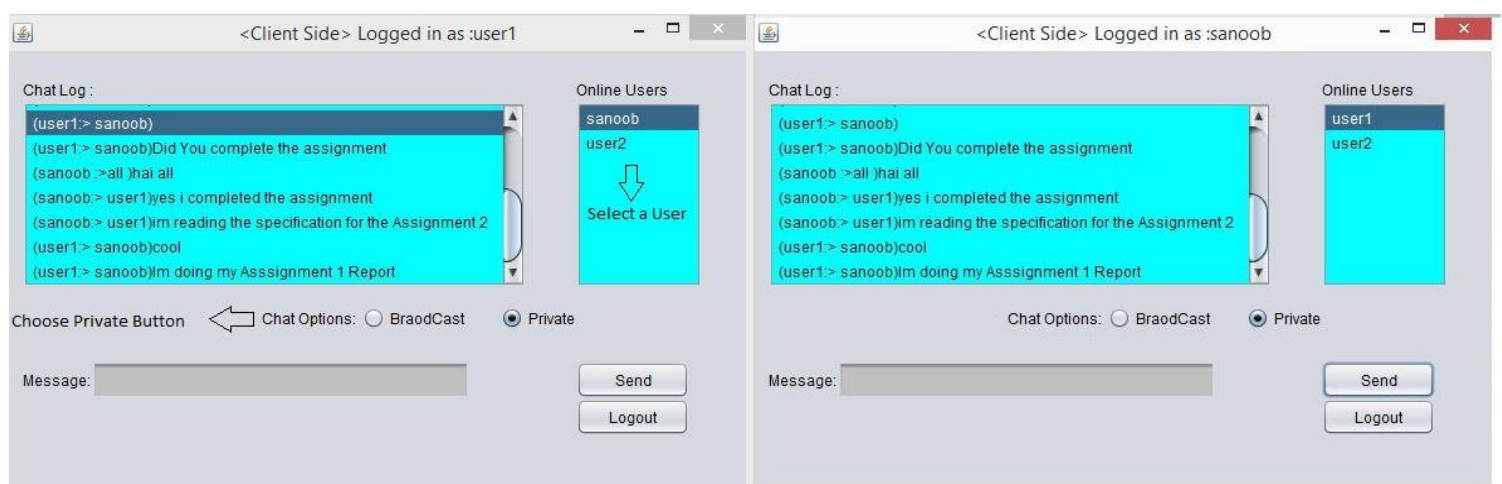
1. Select Broadcast Radio button.
2. Type your message in the text box and click send button
3. The Broadcast message appear in the chat window as (**Username** :> all) where **Username** is the Senders Username



Private Messaging.

It's an interesting feature in this application, Clients can have Private messages in between them. If two users want to chat in private and don't want others to see the Chat in their chat log. They can make use of Private Chat services. To do a private chat

1. Select Private message Radio button.
2. Select a user from the Online User List.
3. Type a message in the message box
4. Click the send button.
5. The format of the private message is (**Username** :> **Username1**), where **Username** is the sender's username and **Username1** is Receivers Username. And only both of the clients can view this message.



Program in execution.

When the program is executed you will run the server code first by running the **Run Server.bat** file, Once the Server is up and running you need to create a client by running **Run Client.bat**, If you want multiple clients you need to run the file multiple times.

For the Client to use the application the user need to login first with a valid username and password, if the User need to register click the **register** button on the login page to create a new user. All the User data is stored in **Enc.doc** file. Only if the login credentials entered by the User match with the Enc.doc file, the client is logged in. Else the client gets an error notification.

Notifications are sent to Server and Client GUI. A list is maintained in both Server and Client for maintaining a list of online users. This List can be used for sending private messages in the client side GUI, and to monitor the online user in the server side.

There are also Private and Broadcast messaging facility, to choose between these two just click on the needed radio button. Once the message is sent. Needed headers are added to message to distinguish it between private, broadcast and notification messages.

Notifications in GUI.

Notification on the server side.

```
<sanoob has Logged in>
<user1 has Logged in>
```

```
<user1 has Logged out after 17 seconds>
```

Notification on client Side

```
user2 has Logged in
(sanoob :>all )hai all
user1 has Logged out after 17 seconds
```

When a client is logged in, the server and other users get a notification in their chat window saying **<Username> has logged in**. Similarly When a client is logged out, The server and other Clients get a notification in the chat window saying **<Username > has logged out after ss Seconds**.

Statistics Function at Server Side

```
Total No of registered Users :12
Server Uptime: 39Seconds
The number of online users:2
```

```
Total number of broadcasts:1
Total number of Private Messages:10
```

On the server side GUI, It shows basic parameters of the chat application like, total registered users, Server uptime, Number of online Users, No of broadcast messages, and no of private messages.

Conclusion

This is a simple chat java application with lots of features.in both client side and server side.

In the Server side.

1. Chat window to see the chat logs.
2. Can view the names of online users.
3. Statistics information such as
 - a. Server uptime.
 - b. Number of Broadcast messages sent
 - c. Number of Private messages sent.
 - d. Total number of Registered Users.
 - e. Total Number of Online Users

In the Client side.

1. Chat Window to see Chat logs
2. Can view online users
3. Can send Broadcast Messages
4. Can send Private Messages
5. Logout button for Users to logout.

Reference

1. Socket program of multi client chat server - Java Examples and Tutorials . 2014. Socket program of multi client chat server - Java Examples and Tutorials . [ONLINE] Available at: <http://www.dailyfreecode.com/code/socket-multi-client-client-chat-server-1252.aspx>. [Accessed 04 May 2014].
2. Welcome | Java.net. 2014. Welcome | Java.net. [ONLINE] Available at: <https://www.java.net/>. [Accessed 04 May 2014].
3. Stack Overflow. 2014. Stack Overflow. [ONLINE] Available at: <http://stackoverflow.com/>. [Accessed 04 May 2014].