

# Java Based Chat Server

**Tanuj Mishra**

*School of Polytechnic, Lovely Professional University, Phagwara, Punjab, India*

**Abstract:** Our principle point is to make a one of a kind undertaking which is valuable not just for investigating our self in the highlights of Java yet in addition helpful for universities, organizations and numerous different workplaces furnished with PCs associated with LAN. This undertaking is particularly required in such zones where representatives can't drive normally for little discussion. On the off chance that we have LAN we can without much of a stretch run this application. So we have chosen to make such application which we have chosen to introduce in our school. For the most part on LAN we can simply share our reports and records however we can't convey yet we have chosen to make it conceivable through "Visit Server".

Keywords: Chat server, Java, Server

## I. Introduction

As a rule, the activity of any server is to offer a unified assistance. Be that as it may, there are a wide range of methods for offering types of assistance, and various approaches to structure the interchanges. Visit is generally portrayed as an association arranged help, in light of the fact that a client sets up an association and keeps up that association, sending and getting content for the term of the meeting. This is as opposed to the Web, where the convention is value-based - the program requests a page, and the server sends it; the association is then shut. (Practically speaking, the association is kept open and reused, yet this is more a speed-streamlining than an organizing representation.) We'll be making a stripped-down, association arranged server. Learning the fundamental structure will help us a lot in making other association situated servers later on. The word system includes a specialized definition inside the article arranged network - it implies a plan structure that can be reused. This isn't equivalent to conventional code reuse, in which bits of code composed for one reason (or for no specific reason by any means) are reused for another reason. Or maybe, a system is a reusable, all-encompassing structure inside which you can actualize your specific application. One approach to think about the distinction between a system and conventional code reuse is that customary reuse includes embeddings prior components into your specific structure, while a system includes embeddings your specific components into a previous structure. The system we'll use right now following components. When we've experienced this instructional exercise and know about these components, we'll have the option to utilize them again when we make our own association situated server. The accompanying components are:

- AWT (Abstract Windowing toolbox)
- Applets
- Events
- Swings
- Threads
- Socket
- Database availability

Our server is an independent program. A solitary Java process running on its own machine. It won't require any help programming other than a Java virtual machine. Also, it won't require a Web server or application server, albeit a Web server or application server will probably be utilized to serve the customer applet to the customer.

Further developed server frameworks regularly insert the server code inside a bigger system. This system may be utilized to supply highlights, for example, load-adjusting, extraordinary libraries for taking care of huge quantities of customers, process movement, and database administrations. Be that as it may, our model is going to stand without anyone else. The code that we'll create right now a system, in that it isn't organized to permit you to effectively repurpose it for something different. Organizing it that way would occupy from the reason for this, which is to investigate fundamental procedures of server programming. Be that as it may, in a plan sense, it is a system in light of the fact that the structure of the code is something, you'll have the option to utilize over and over.

## II. Features of the application

This application is used for sending messages from one client to all clients. In this, server takes a message from the sender client and sends it to all clients. If any client wants to join the chatting then he runs the client application on client side, enters the login name and hits the login button and starts the chatting with every client. We have made a provision that if any client joins the chatting system then username will come on the message list box of every client window. Same like this, if any client exit from the chatting then the message username will disappear from the list box of every client. In this we are maintaining the list of user names (client name) also. After creating the connection with server, client sends the user name to server and server store it in the array list and sends this array list to all clients. Same like this when any client has logged out then server remove this name from the array list and send this updated array list to every client.

Also, we have used color chooser dialog box in order to use or apply the selected color over the background and the text written. We have added font family in order to apply it on the text. If server is not in running condition then you cannot start the chatting. In order to make this project more useful we can send or share our documents online through chat server. But this practice needs a lot of changes. But for the future expansion of this project we can use this feature. Also, there are lots of features which can be added in this project, like other chat servers we can add the facility of "Private Chat". For this we require some changes in socket programming. So, in this way we have made an application which could be helpful if we make it more advance. But still we can install our project in small offices so that proper interaction should be possible between the employees.

## III. Analysis

We are making a chat server, so as we know that for chatting, we require basically two windows, these are:

-One is for the Server containing a text area box for the reception of the messages and a text field for writing the messages. For sending the messages we require a send button and a log out button also.

-Another window is for the Client which contains the same fields and buttons as we have in Server window.

These windows contain other components too like logout buttons and some menu items. So, this is the main framework and rests of the windows are just for the authentication and validation which comes under designing or to make our project user-friendly and full of advancement. All these windows are shown below with proper Designing and snapshots.

#### IV. Login Window

**Login window** is basically used in order to make sure that only authenticated and validate users are allowed to login. This can be done by providing unique password and username to each and every user. In our project we have also designed login window so that only valid users can chat. We can create our account and have a nice chat with our friends or colleagues. We will discuss new user window later.

**The login window consists of the following GUI components:**

##### Text Fields

- User Name      This is used to enter the user name
- Password        This is used to enter the password

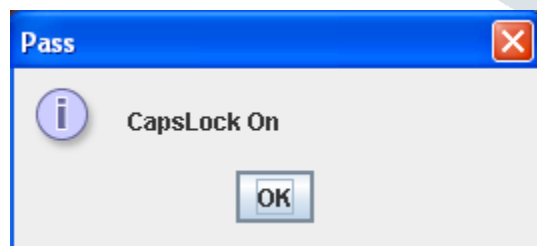
##### Buttons

- Ok
- Cancel
- New User
- Forget Password

##### Labels

- Username
- Password
- Image

Here we have added a feature, if the letters of the username are in uppercase then we will intimate you that caps lock is on as shown below in snapshots.



We call it as “information message” and is used with JOptioPane class. There are certain other messages like:

- Question messages
- Error messages
- Plain messages

So we can only login if we type username in lowercase. This is as shown below:



## V. New User Window

New User is a link in the login window. Clicking that link will open a new window used for Creating New Account. **New User** is another window which is used for the creation of the new user account. If the person is not registered with the chat server, he/she can do so with the help of this window. If the person already had an account, he/she can use this window in order to create another account.

**Following are the GUI components:**

### Text Fields

- Name
- Username
- Password
- Answer

### Labels

- Name
- Username
- Password
- Answer
- Select Question

### ComboBox

This box contains a number of questions from which we have to choose one and answer that question. This will help in case we forget our password.

- Select Question

### Buttons

- Ok
- Cancel

We can see all these components in the figure shown below:

## Forget Window

This is again a link in the login window. This will open a new window called **Forget Window**. This will help us to retrieve the password. There is a question field over the frame. We have to select that question and correspondingly type the answer. Make it sure that we should type the same answer that we have entered at the time of new account.

Following are the GUI components:

### Text Fields

- Name
- Username
- Answer

### Labels

- Name
- Username
- Answer

### ComboBox

- Answer

### Buttons

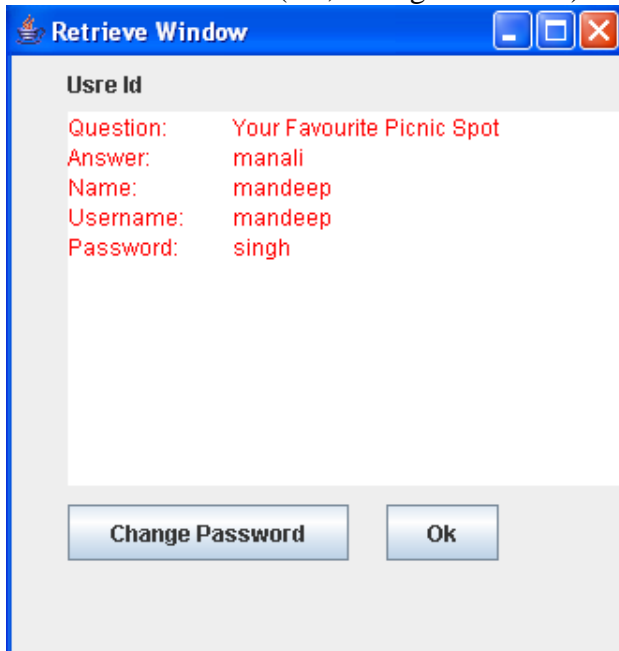
- Ok
- Cancel

## Retrieve Window

By clicking the Ok Button Retrieve Window will appear. This window unveils all the personal data of the user.

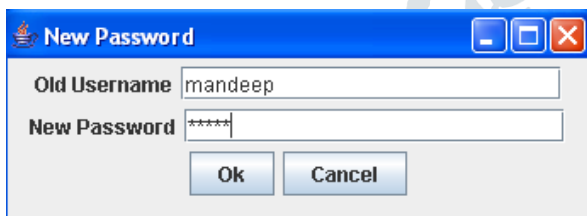
This window contains following components:

- One label named User Id
- Text Area for showing the details
- Two Buttons (Ok, Change Password)



### Change Password Window

If we want to change our password then click on the “Change Password” and a new frame will appear. Fill all the formalities and click on ok button.

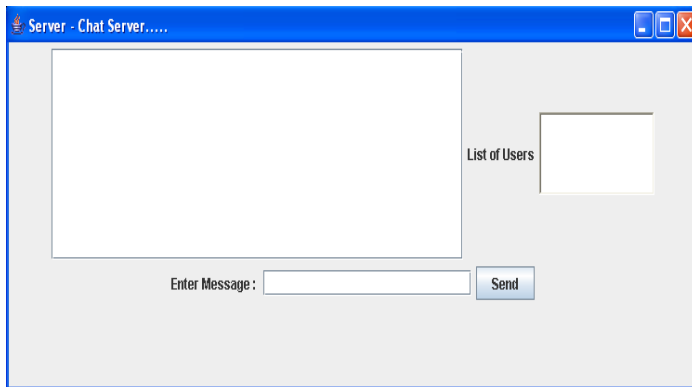


### Server Window

This is the main window in chat server. Only if server is running, we can chat with our friends. Server plays the role of a bridge between the clients.

This is the window of the server. Here we have following components

- TextArea
- TextField
- List
- Button



When we enter to this window it will send a protocol to the clients for connectivity.

We can see in the command prompt.

```

C:\WINDOWS\system32\cmd.exe - java newlogin
C:\project>javac newlogin.java
C:\project>java newlogin
Server is waiting for client to be connected.....

```

After this open another command prompt window and run client. When client starts running we will get a message over the command prompt as shown below:

```

C:\WINDOWS\system32\cmd.exe - java server
C:\new>java server
Server is waiting for client to be connected.....
Socket : 0 - Socket[addr=/127.0.0.1,port=1081,localport=95]
Client connected successfully.....

```

Now we are ready to chat.

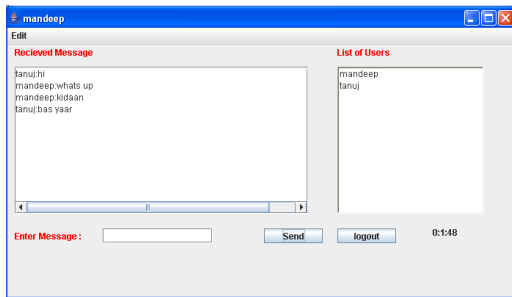
## Client Window

This is the main window through which we'll do chat with our friends or colleagues. This window has certain features or further we can add more.

**So this window contain following GUI components:**

- TextField
- TextArea
- Buttons
- List
- Labels
- MenuItems

## Snapshot of the Client window:



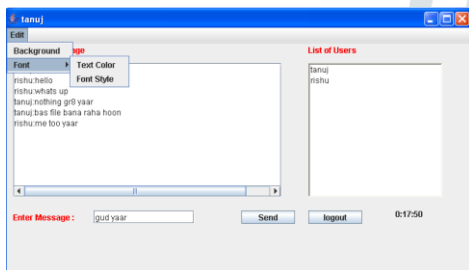
So we can see clearly in the snapshot the each component we have used. The importance of each component is written below:

- Here we can see in the snapshot that TextArea is used for receiving messages.
- The number of users which are online is displayed in the List under the heading “List of Users”.
- TextField is used for writing the message.
- By clicking on the send button message can be send.
- In order to stop the chatting, click on logout button.
- One label is used in order to display the time from when we start chatting.

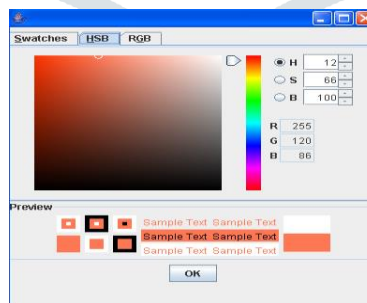
There are some other functions we have added in the client window. These are as written below:

- We have added a MenuItem named “Edit”. Under this there are certain submenu items like Background and Font.
- Under Font there are too submenus like Text Color and Font Style.

We can see both these properties in the snapshot

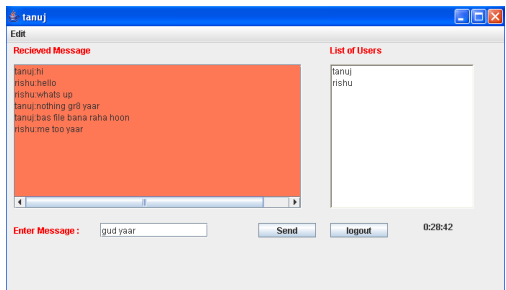


By selecting the option Background, a color chooser dialog box will open. By selecting any color we can apply it to the background of the Text Area as shown below:

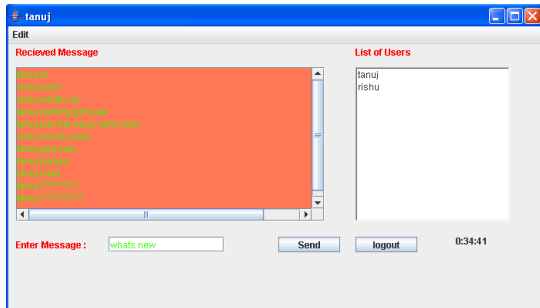


After selecting the color TextArea will looklike as shown below:

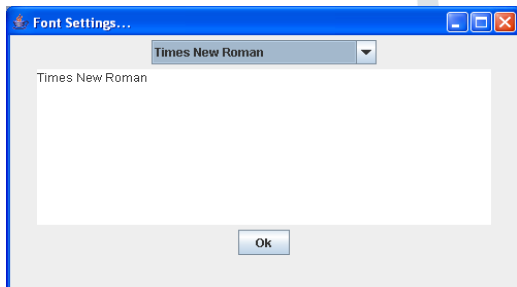




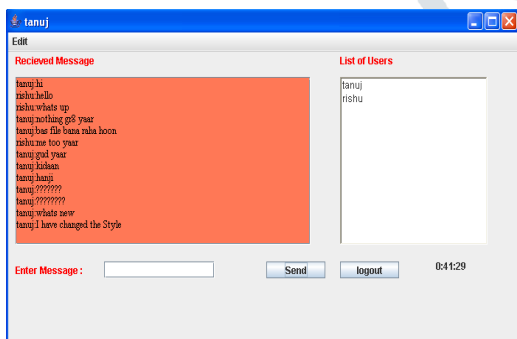
If we will select Text Color under Font then similar type of color chooser dialog box will appear and selecting the proper color text of the received messages will turn into that one as shown below:



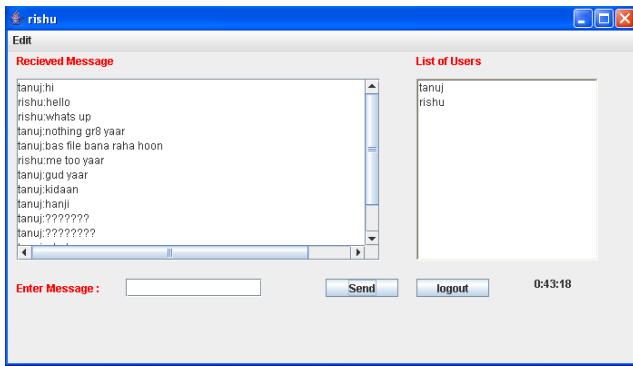
There is another option that is Font Style. We can change the style of the Text into Arial, Times New Roman and all that. We have imported font family for this purpose. We can see in the snapshot.



After selecting Times New Roman. Text style of the Message will convert into selected one as shown below:



The window of the other user looks like this:



So the framework is over we can make further more changes in the designing part. Its not all over we are still working on the designing. But the main windows are ready to work properly.

## V. References

- [1] Handel, M and Herbsleb, J.D (2002) What Is Chat Doing in the Workplace? ACM Publishers, New York.
- {2} Chandra, S.Y and Kumar, S.S (2009) An Introduction to CLIENT/SERVER COMPUTING, New Age International Publishers, New Delhi.
- [3] Reid, F (2004) Network programming in .net with c# and visual basic, Elsevier digital press, 200-wheeler road, Burlington, ma 01803, USA
- [4] Umar, S and Justin, M.P. (2003). Service-oriented Network Socket, Association for computing Machinery Publishers, NewYork.