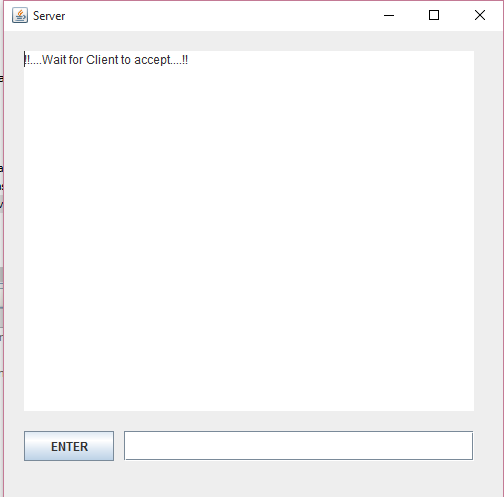
**Shubham Dewangan**

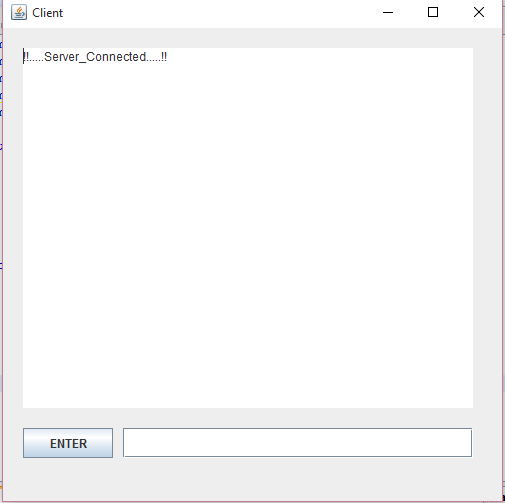
**IEC2013036**

1. Java language is used to make a simple chat application client-server. **Ser.java** and **Cl.java** , both are open but first only the Server.java is compiled and run.

Then, client.java is run. A dialog box will appear for both the server and client as shown in given figure :-

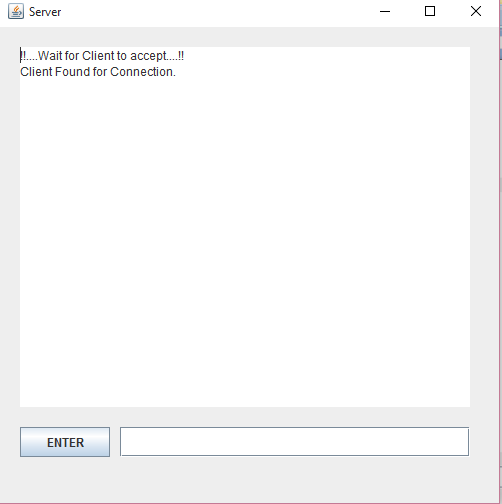
This is how the dialog box will look like.



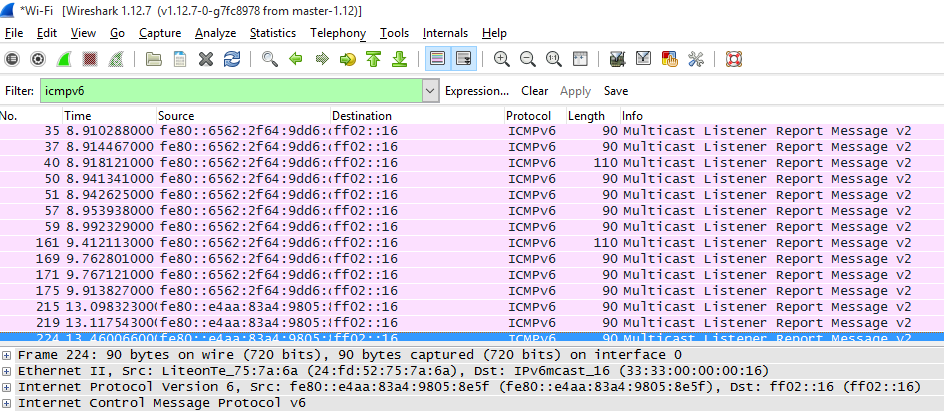


Message is written in the space and press the enter button to send the message.

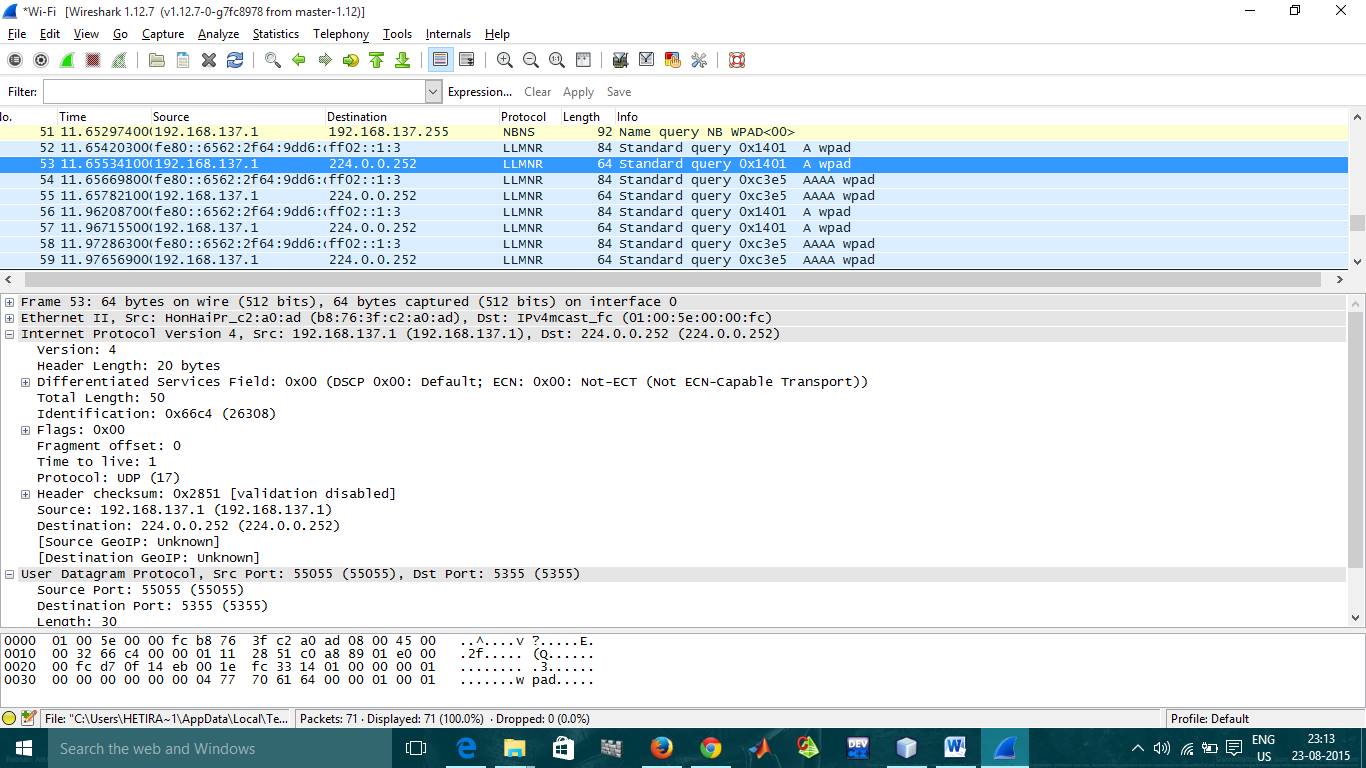
As the client is opened, it gets connected to the server as shown as below:-



Below snapshot represent the wireshark software which captures the packets sent by different network :



WireShark snapshots showing both sending and receiving :



1. For the multiuser chat application, again java language is used for the GUI purpose.

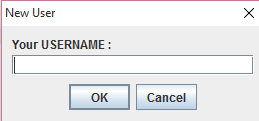
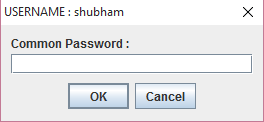
The program can be run by opening both the files named **ChatServer.java** and **ChatClient.java** but run the **ChatServer.java** first in order to build the server. Once it is created then **ChatClient.java** is run to create a client. It prompts the user to enter name and password and identifies the client with that name. Server is assumed to run on the same system (localhost).

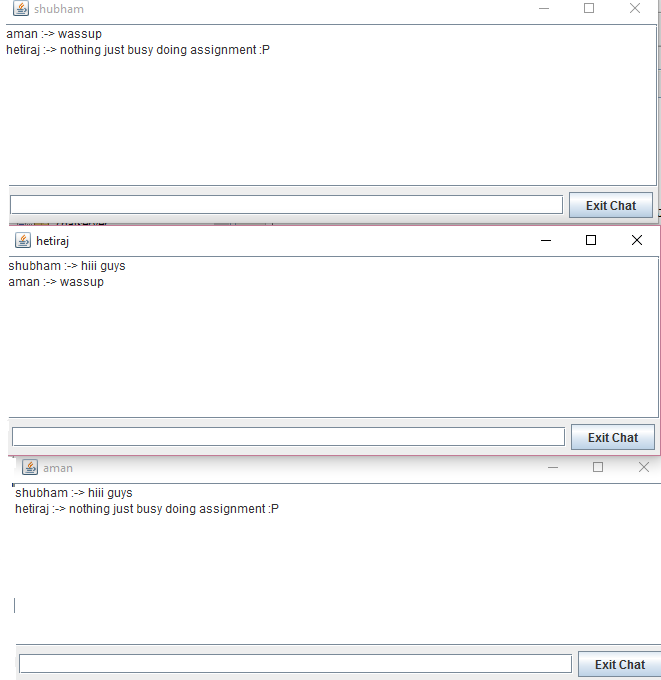
To add more client to the server, again run the ChatClient.java and entered another username but the same common password.

The password is “abcd123” without quotes.

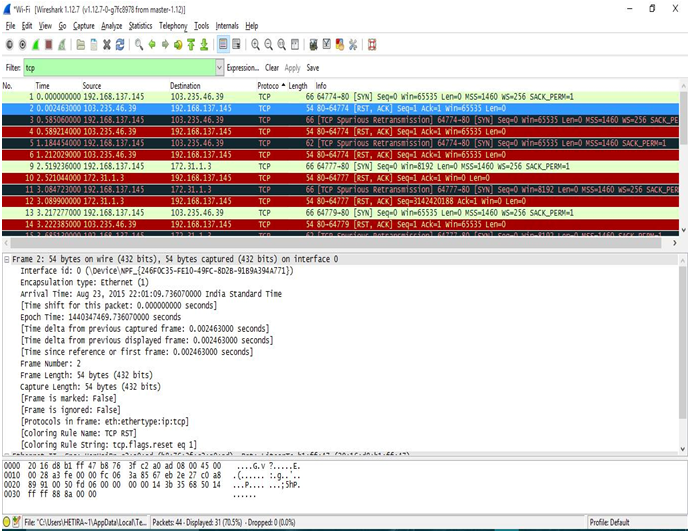
- For n-clients to add, run the **ChatClient.java** n- times and provide n- different usernames.

-The following screenshot shows the program in execution with three clients :-



To see the connection establishment, wireshark packet sniffer is used which captures the packets sent among the clients by the server. It uses the TCP protocol to send and receive messages.



WireShark snapshots showing both sending and receiving :

