|  |  |  |  |
| --- | --- | --- | --- |
| **NAME** | **IDNUM** | **SEC-1** | **Work Ratio** |
| Ahmad Abdullah AL-Qerem | 11819195 | 12:30 – 02:00 | 50% |
| Waseem Samer Ghazal | 11819559 | 12:30 – 02:00 | 50% |

**Programming Assignment # 2**

**Project : ( Peer to Peer )**

Introduction :

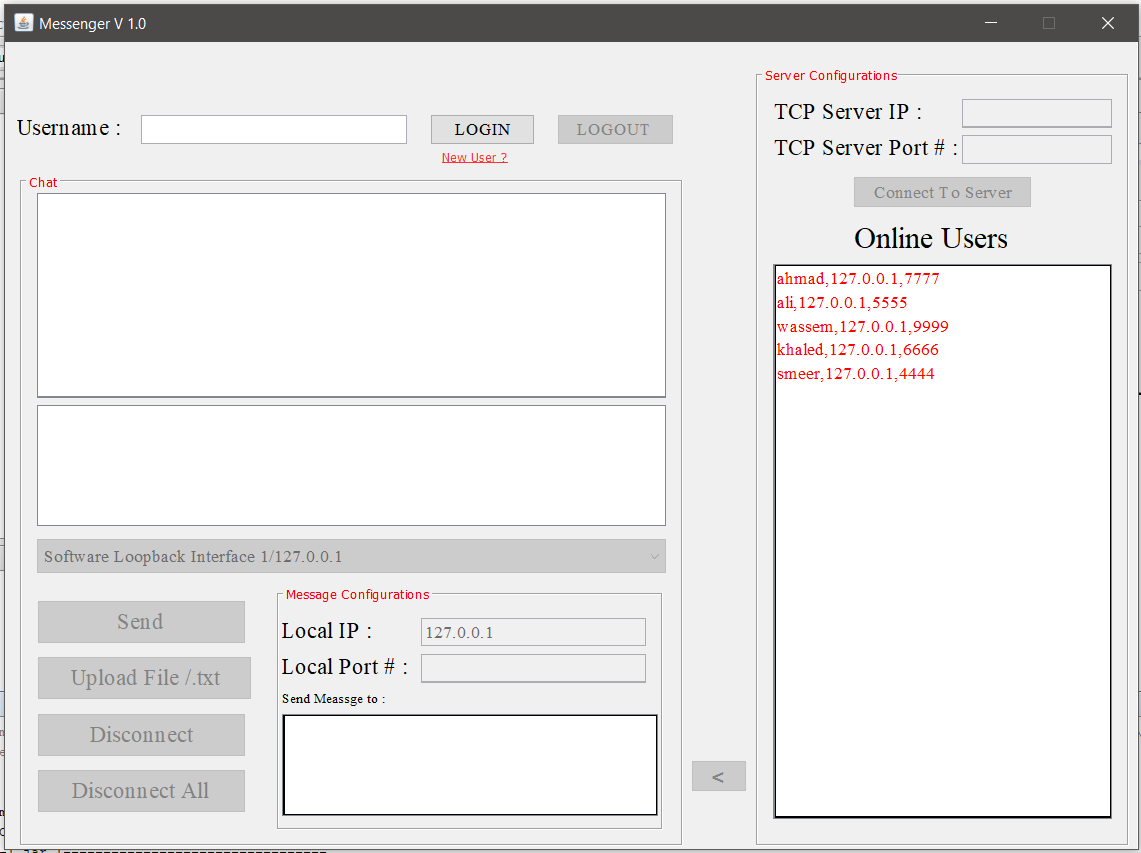
In this project we had written a simple **GUI** application ( Peer to Peer ) that runs on a Network , Using Java Apache NetBeans IDE 12.4 with **JDK 16.1** . Divided to two parts

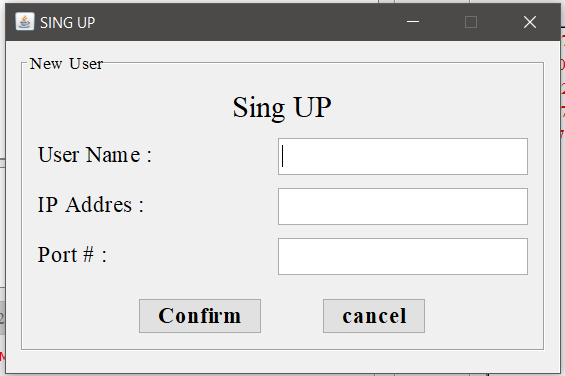
1. **Part 1 :** Implement **P2P** application chatting **UDP** socket in Java.
2. **Part 2 :**  Implement **TCP** Java socket ( Later ) .

Solution :

1. **GUI:**

* Main frame for chatting .
* Sign up ( new user ) frame.





1. Packages :

* GUI package .

Contain the following classes :

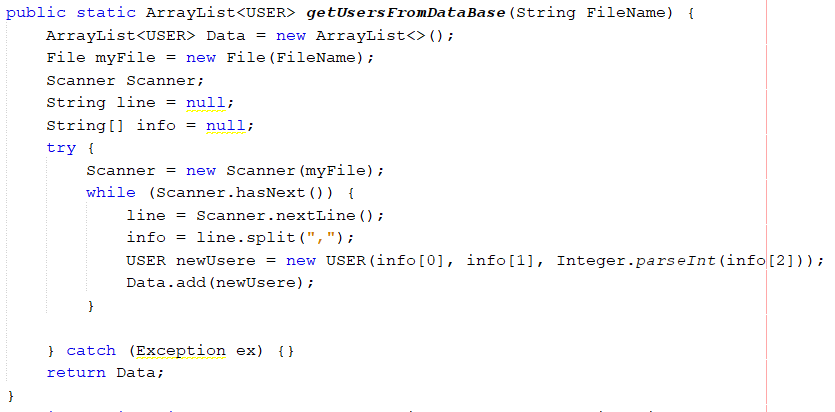
* GUIControler.java : works as **Link** between GUI and NetWork packages .
* GUIFRAME.java : contain the main **GUI**  frame .
* SIGNIN.java : contain the SignIN frame .
* NetWork package .

Contain the following classes :

* ClientThread.java : that used to send messages to **Peers.**
* Message.java : that have the message itself and some information ( source IP address and port number … etc ).
* ServerThread.java : used to work as listener for incoming messages .
* USER.java : have the user information ( name , port number … etc ).

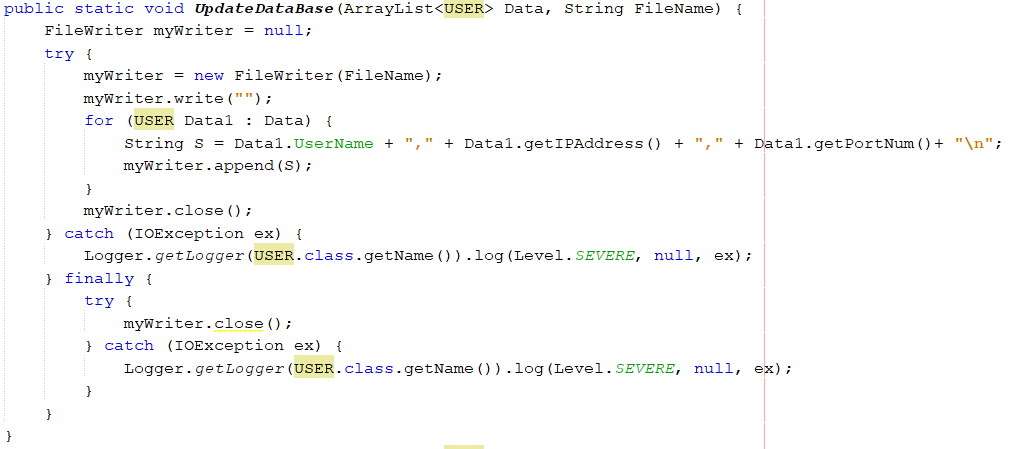
1. Code : will focus in the most important part from our code.

* USER.java functions :

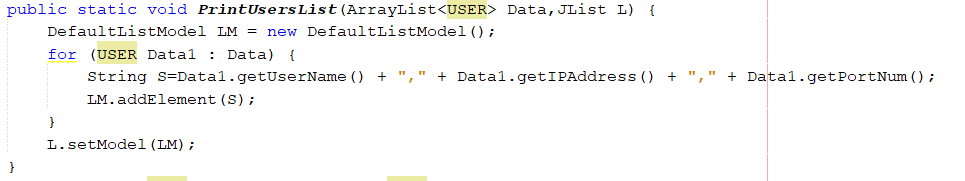


This function will open a txt file in project ( assumed as data base for this project )

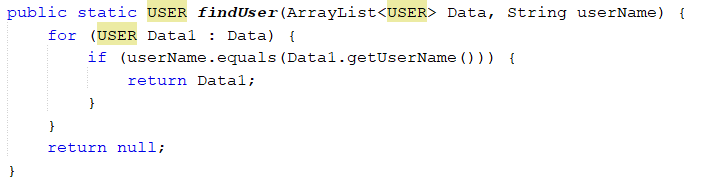
And return its content as an User array List called " Data ".



This function used to update the file ( Data Base ) when new user is sign up and rewrite the file .

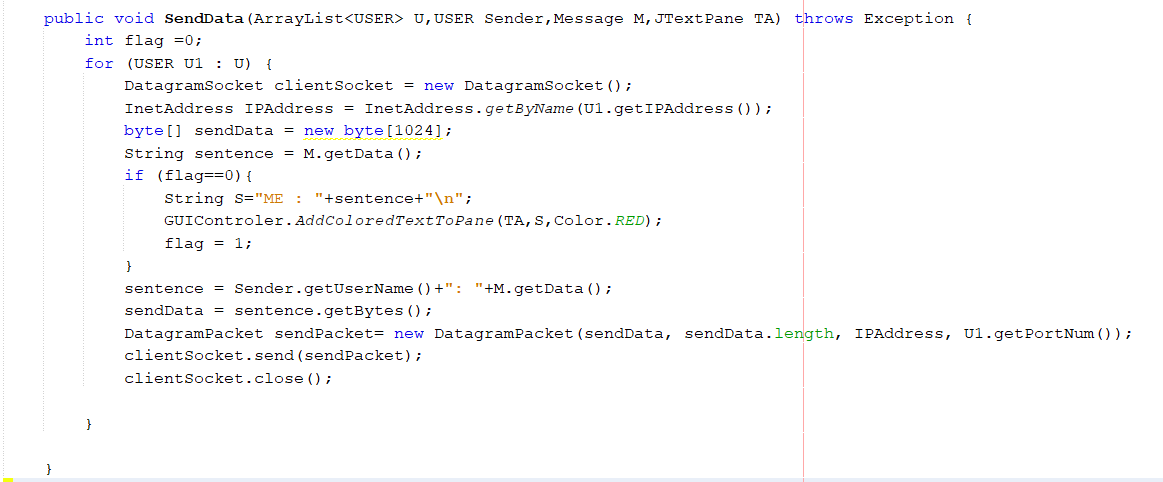


Will get all users are in the data base show them in a list in **GUI** .



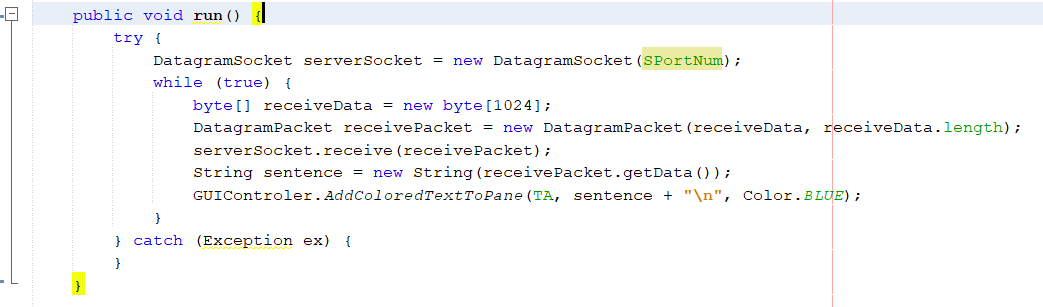
Will search about certain user in the data base .

* ClientThread.java functions :

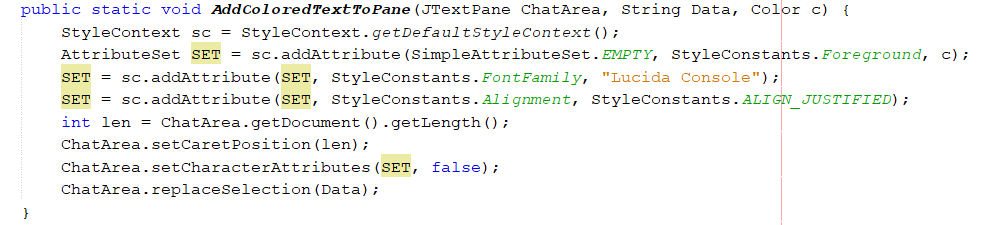


This function responsible about sending messages using **UDP client socket** . takes **"U"** as Array List ( used to send message for multiple users ) , "**Sender**" as USER ( to know who is the sender name and print it later as text in Text Pane ) ,and **"M"** as Message to use port number and destination **IP** are saved in .

* ServerThread.java functions:

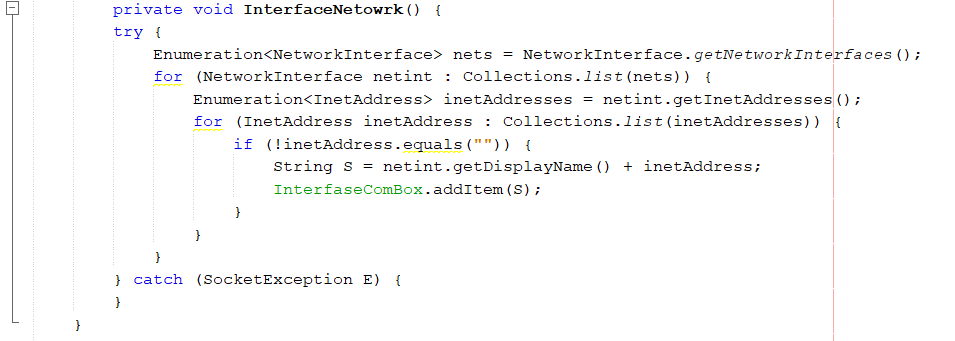
This class extends Thread class to run in parallel with **GUI** frame , need to keep listening to other users who send messages at port **" SPortNum "** and print the incoming messages in Text Pane .

GUIControler.java function :



This is who print messages in text pane with deferent colors .

* GUIFRAME.java function :

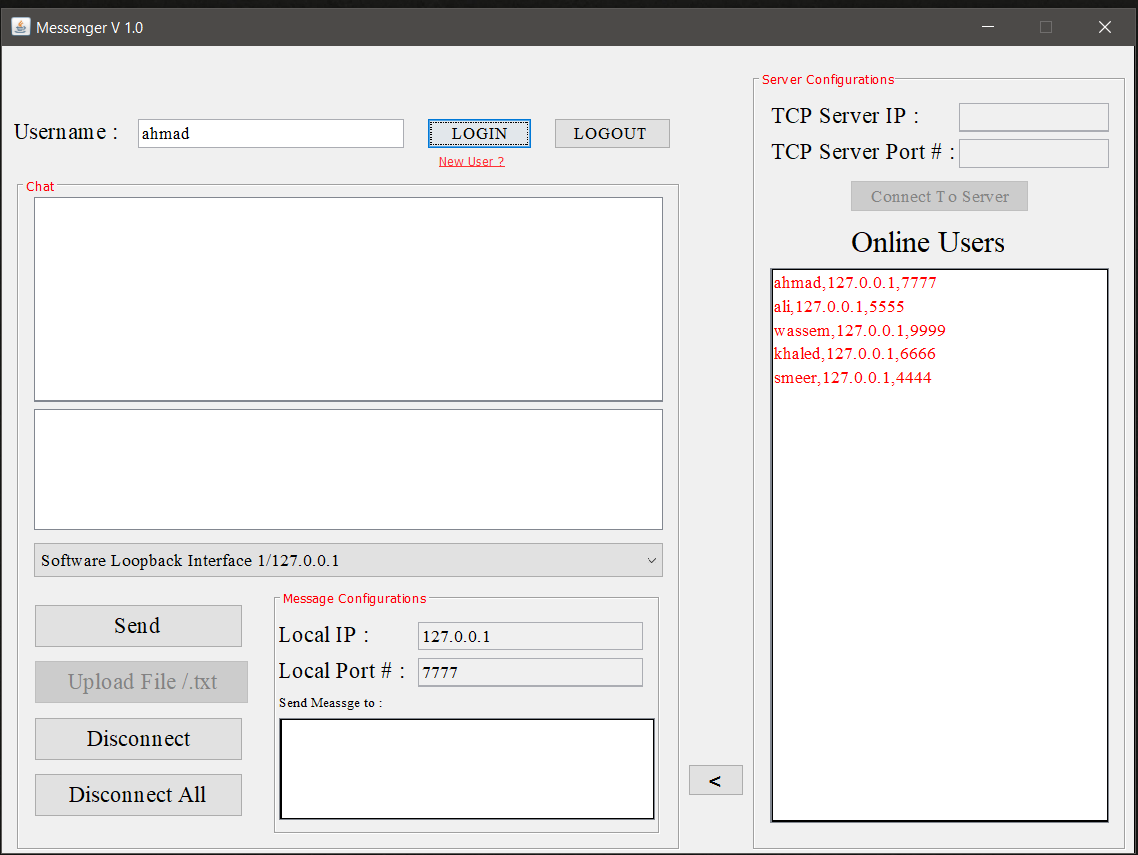


Will get the all interfaces network **IP** addresses and set them in compo box .

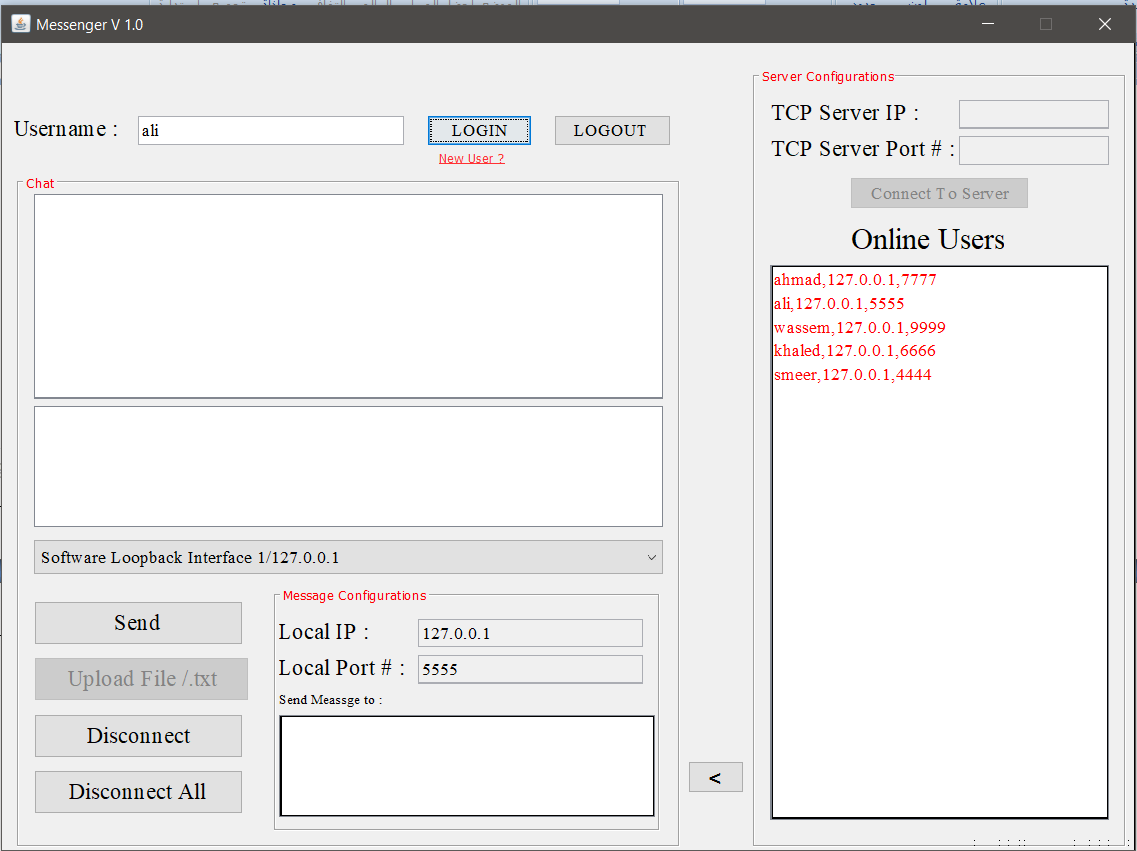
1. Testing :

* Send for one user from list :

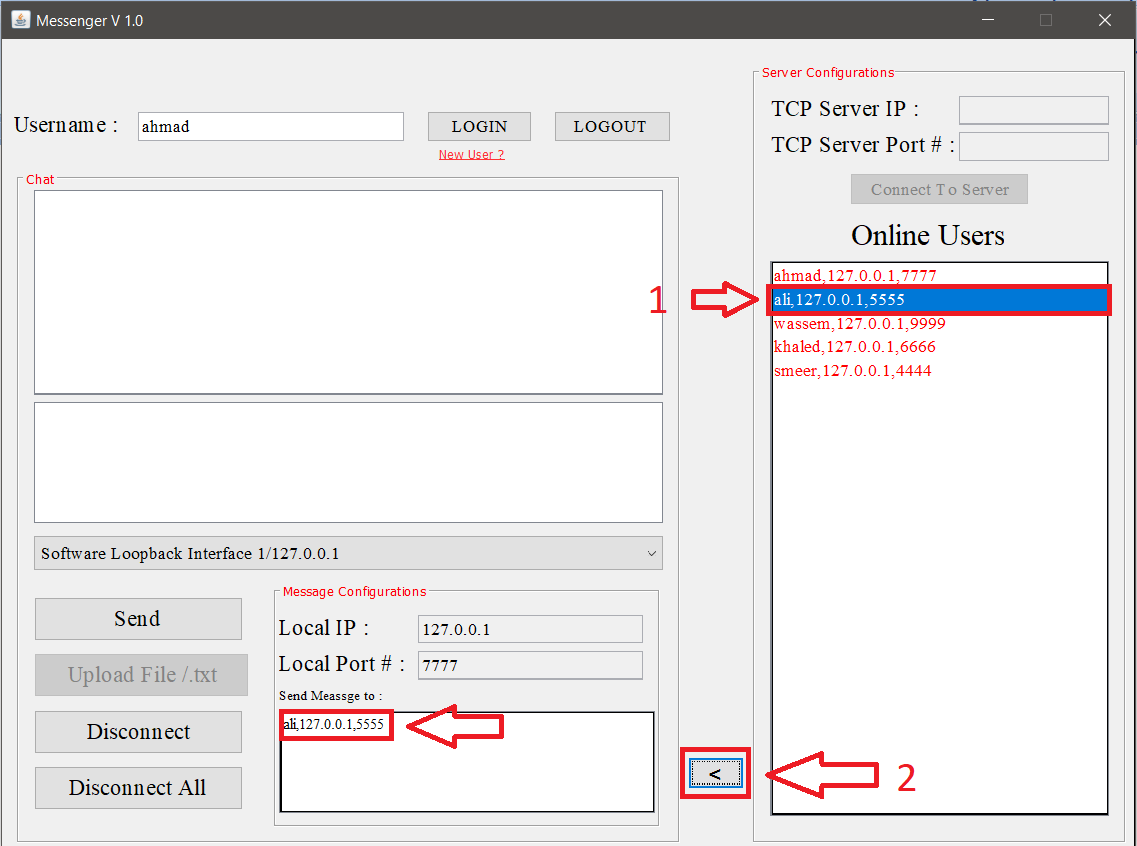
Assume Ahmad will sign in ( already exists in the data base ) and he want to send message to Ali.



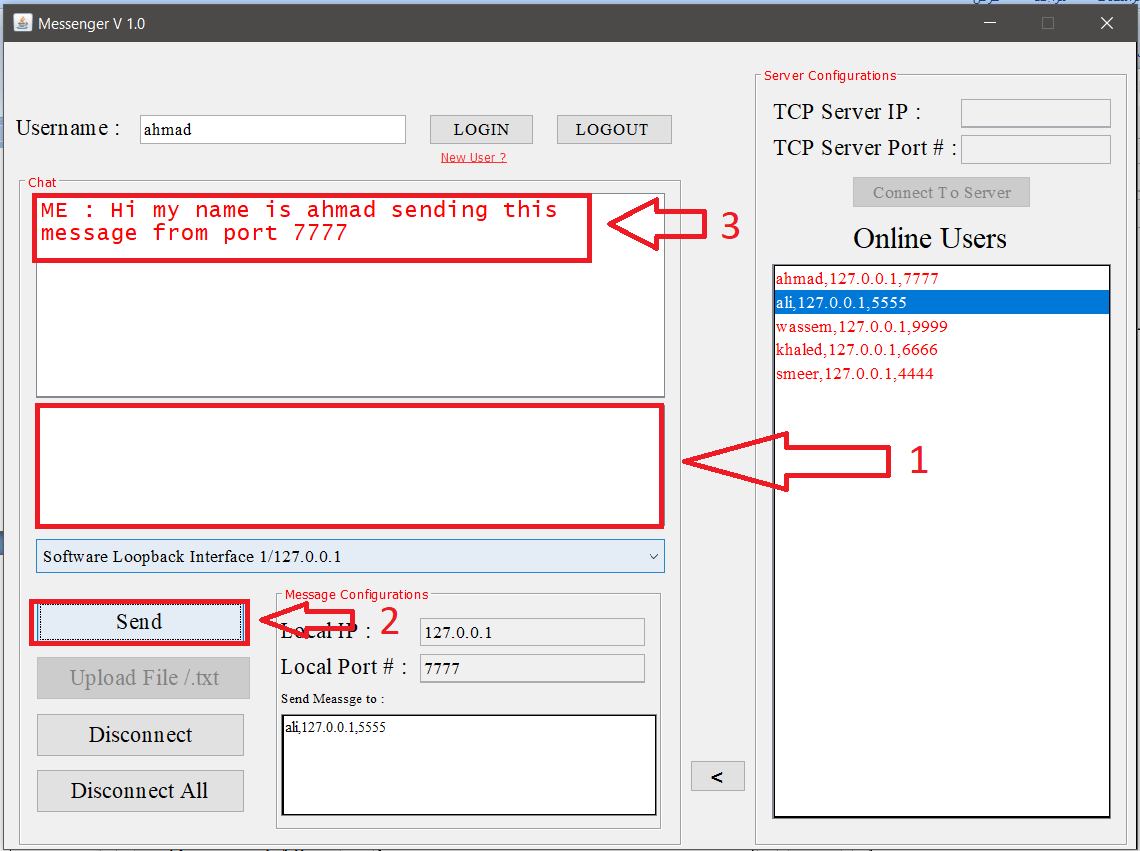
Such that Ali is also signed in :



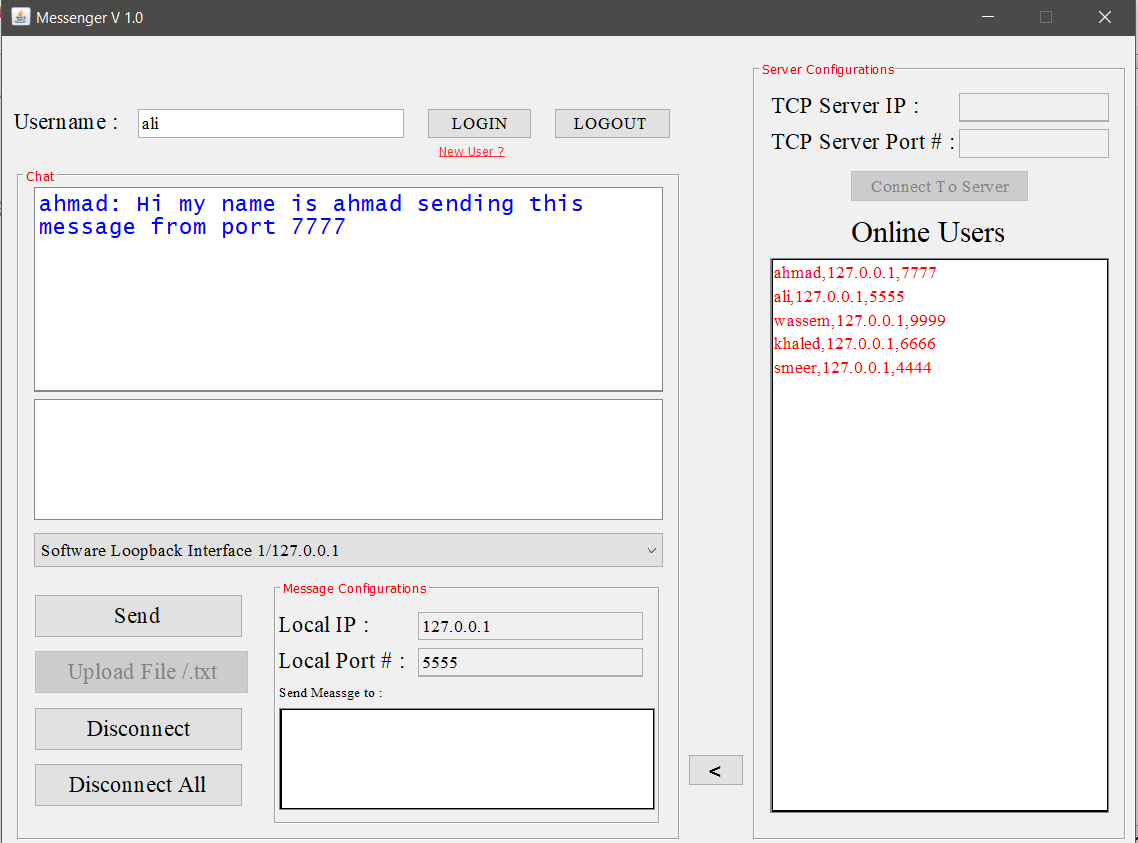
1. Select Ali from Online User List and press in the Arrow button will set Ali as someone who I want to send message for .



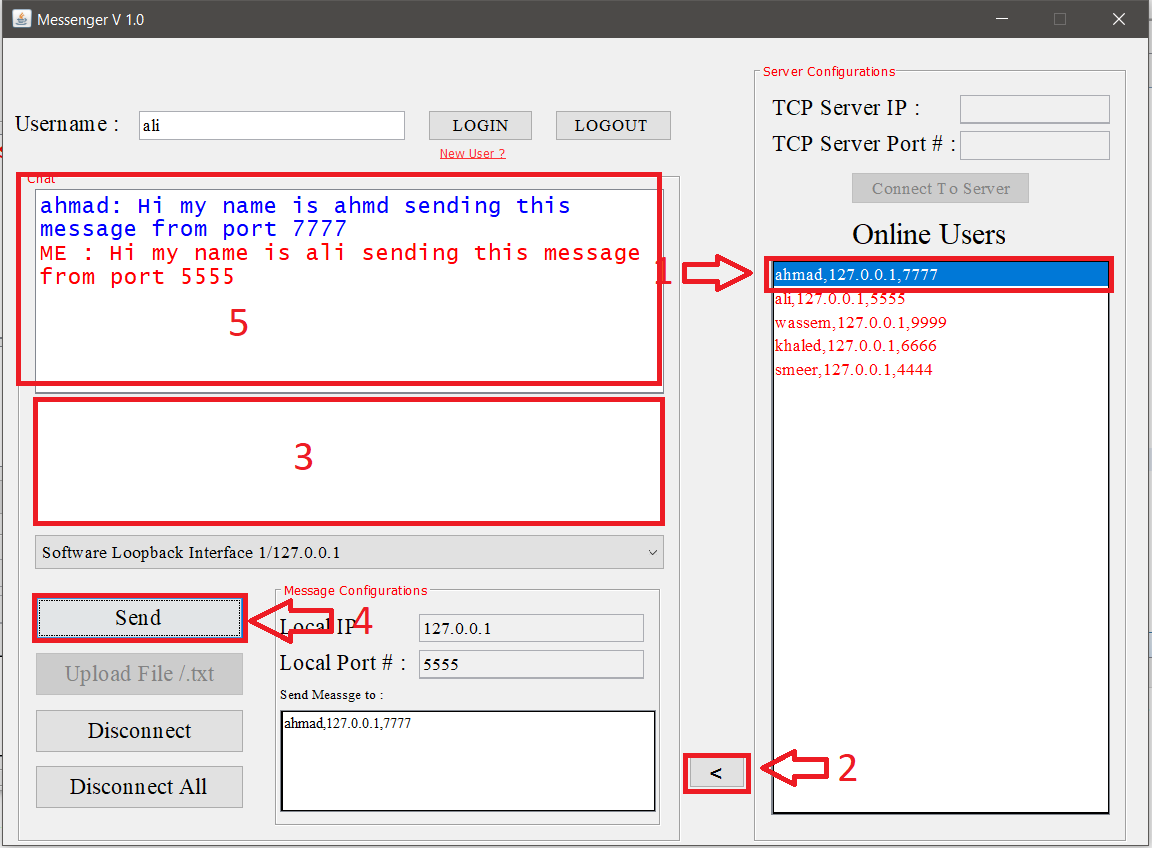
1. then I will type my message and send it .



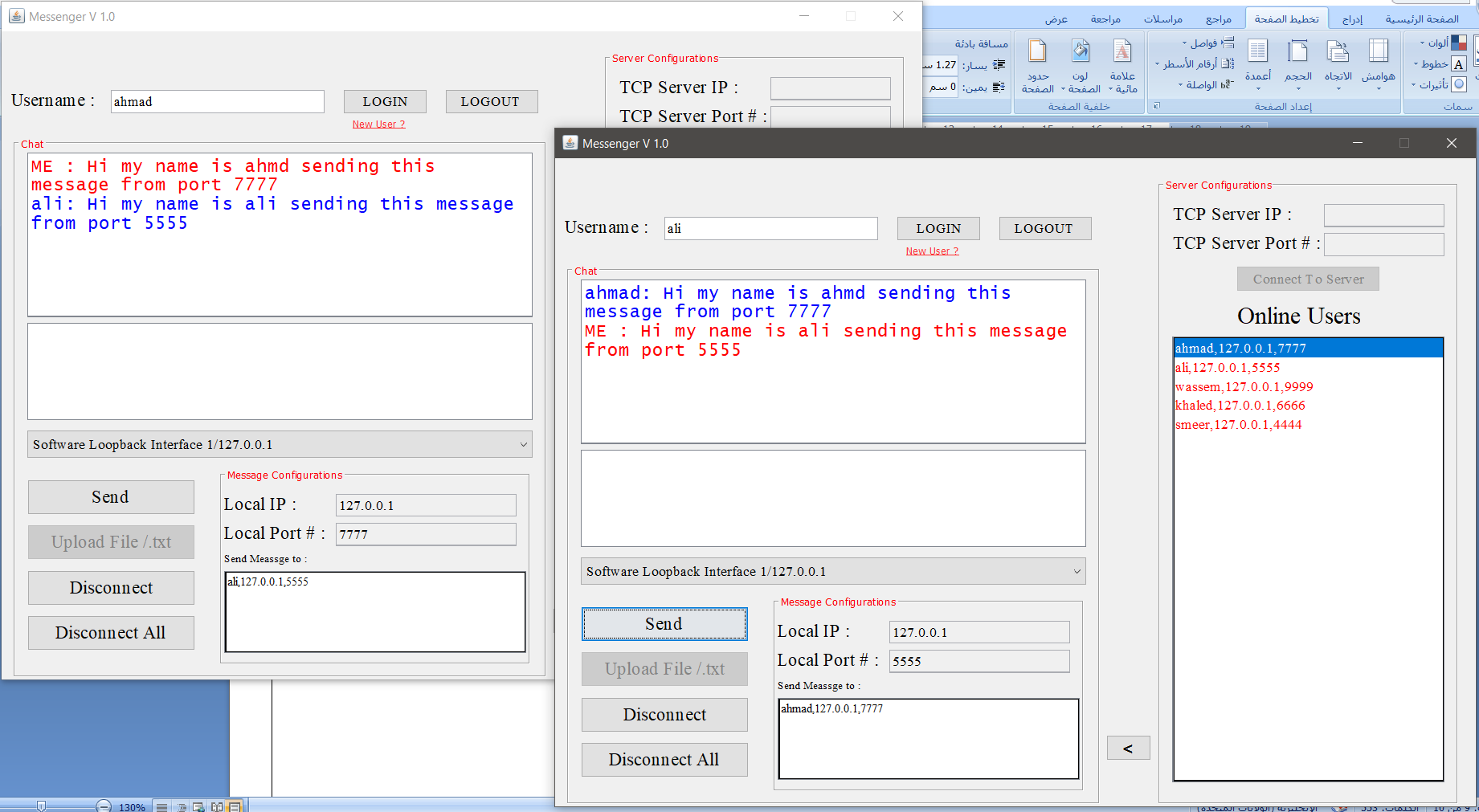
1. what Ali did receive ? :



1. Ali now will replay at Ahmad :

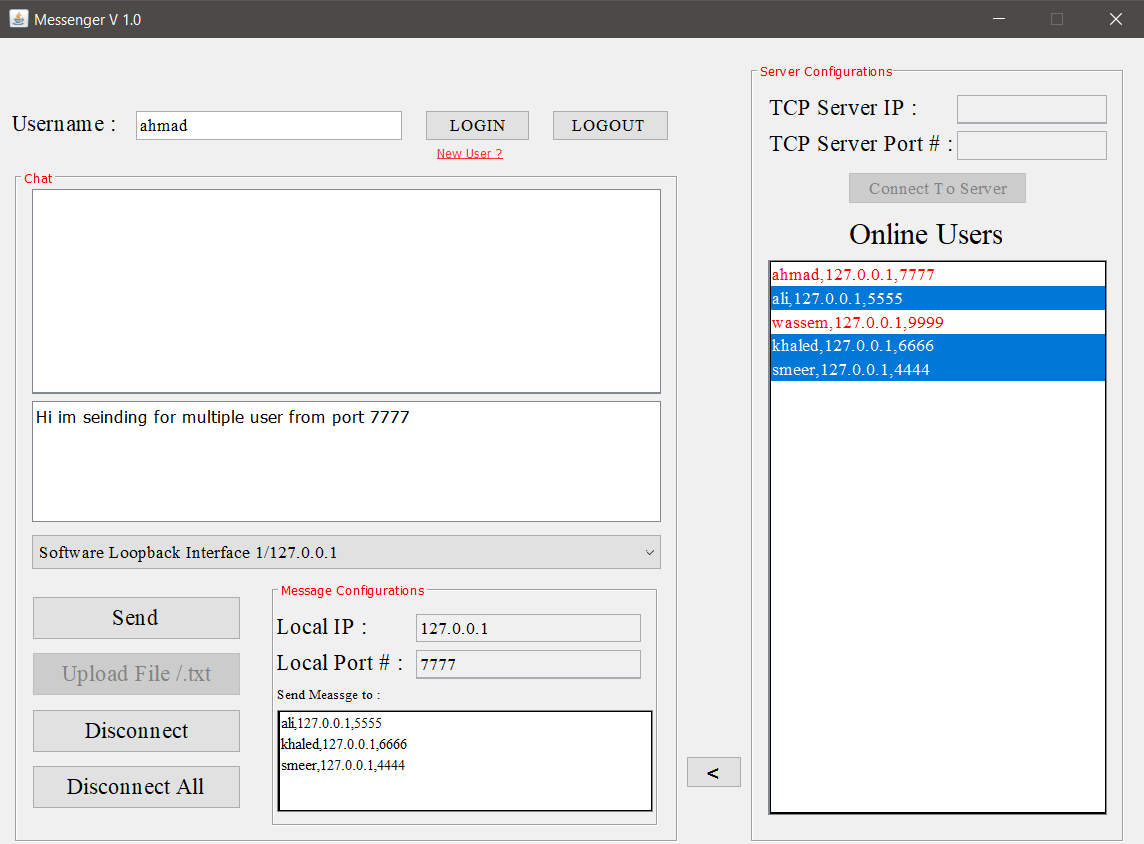


1. what Ahmad did receive ? :

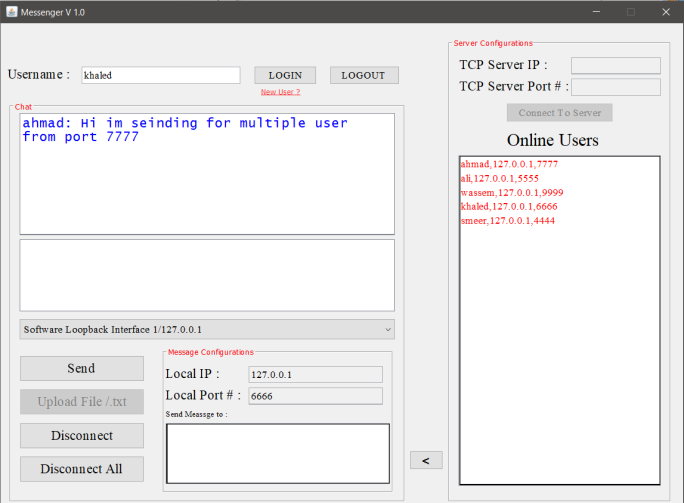
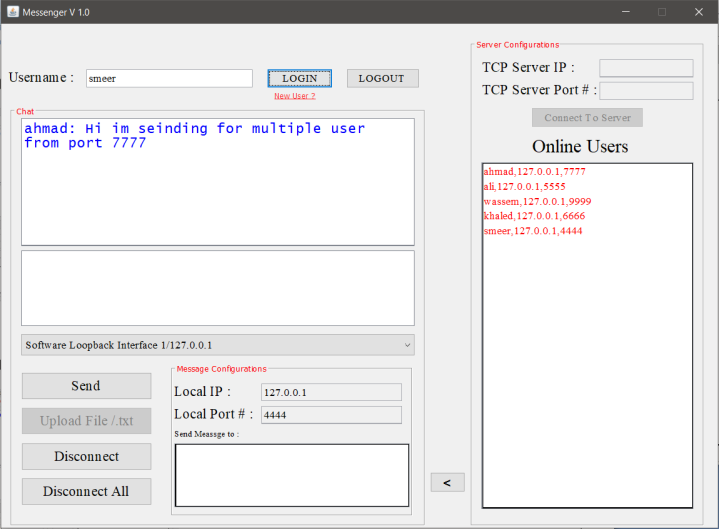


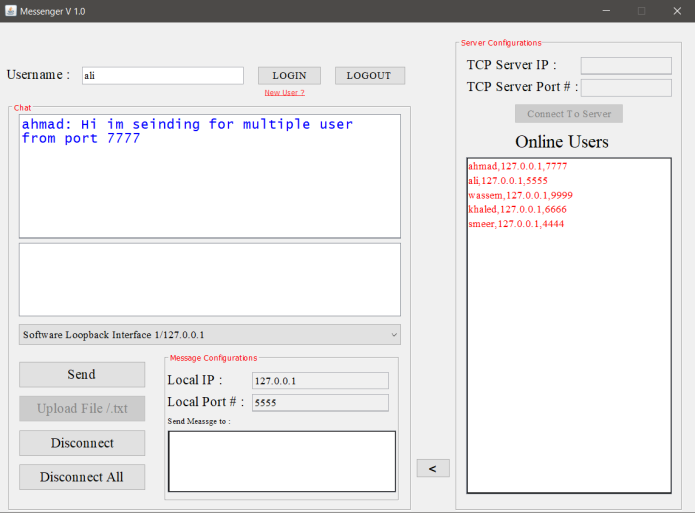
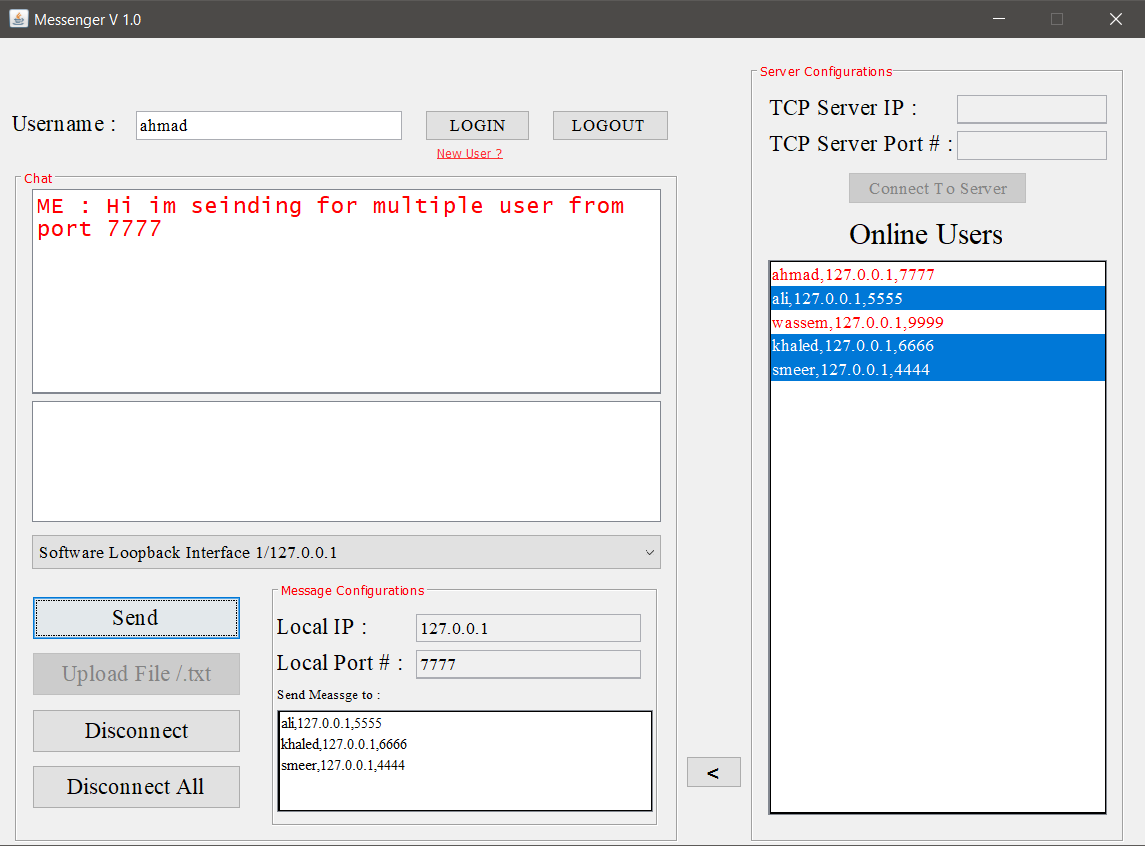
* Send for multiple users from list:

1. Assume some users are signed in and Ahmad want to send a message for them .

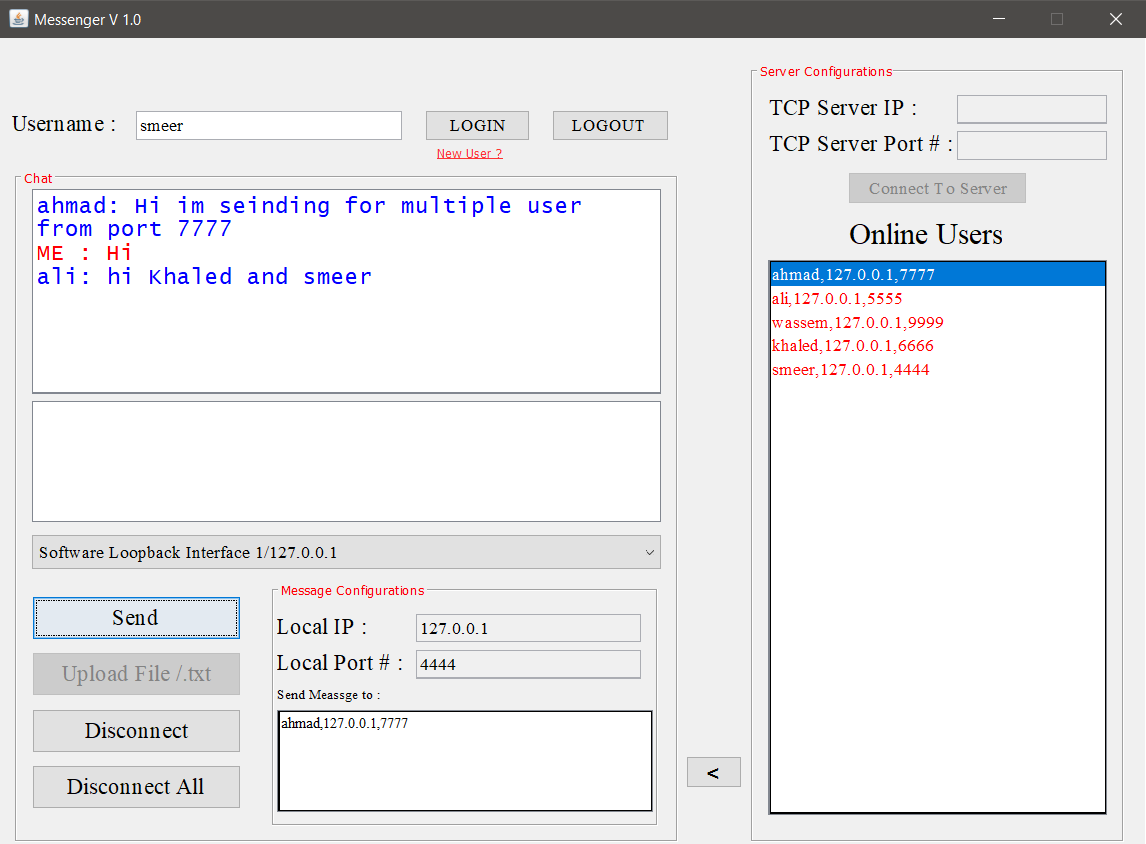
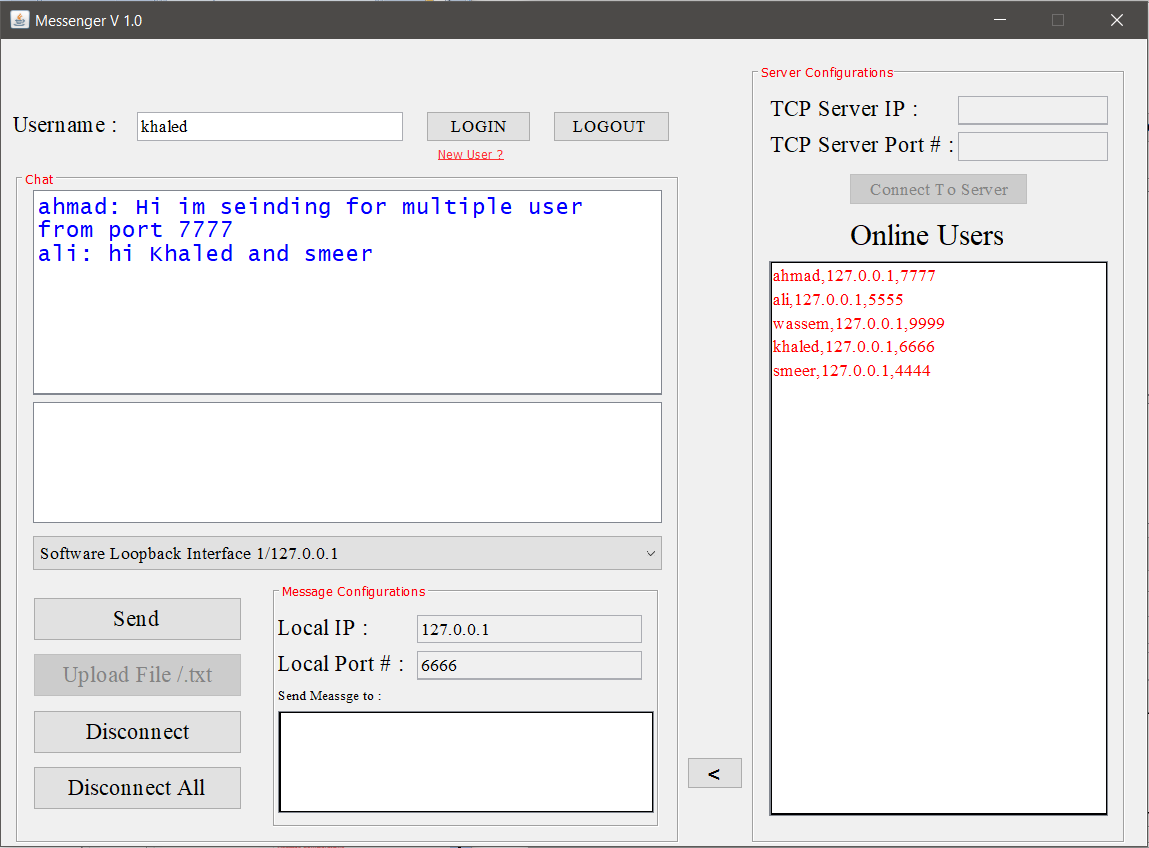


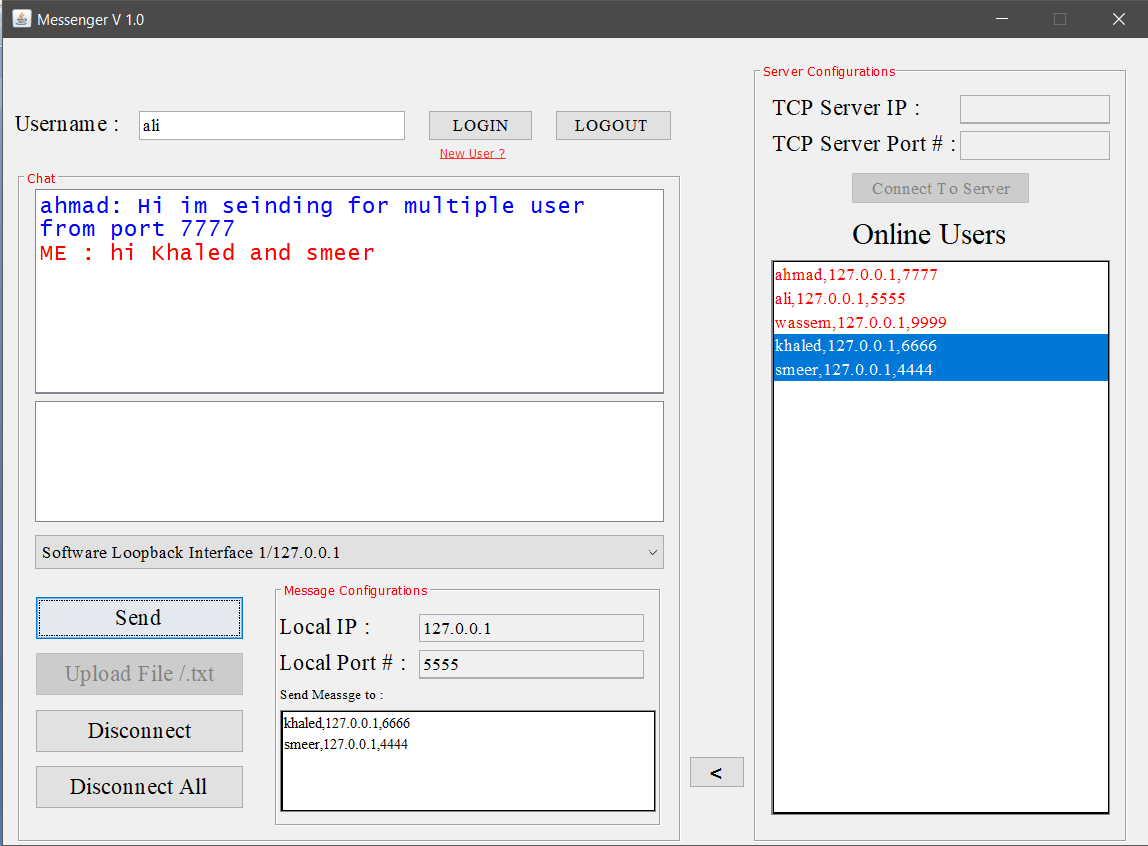
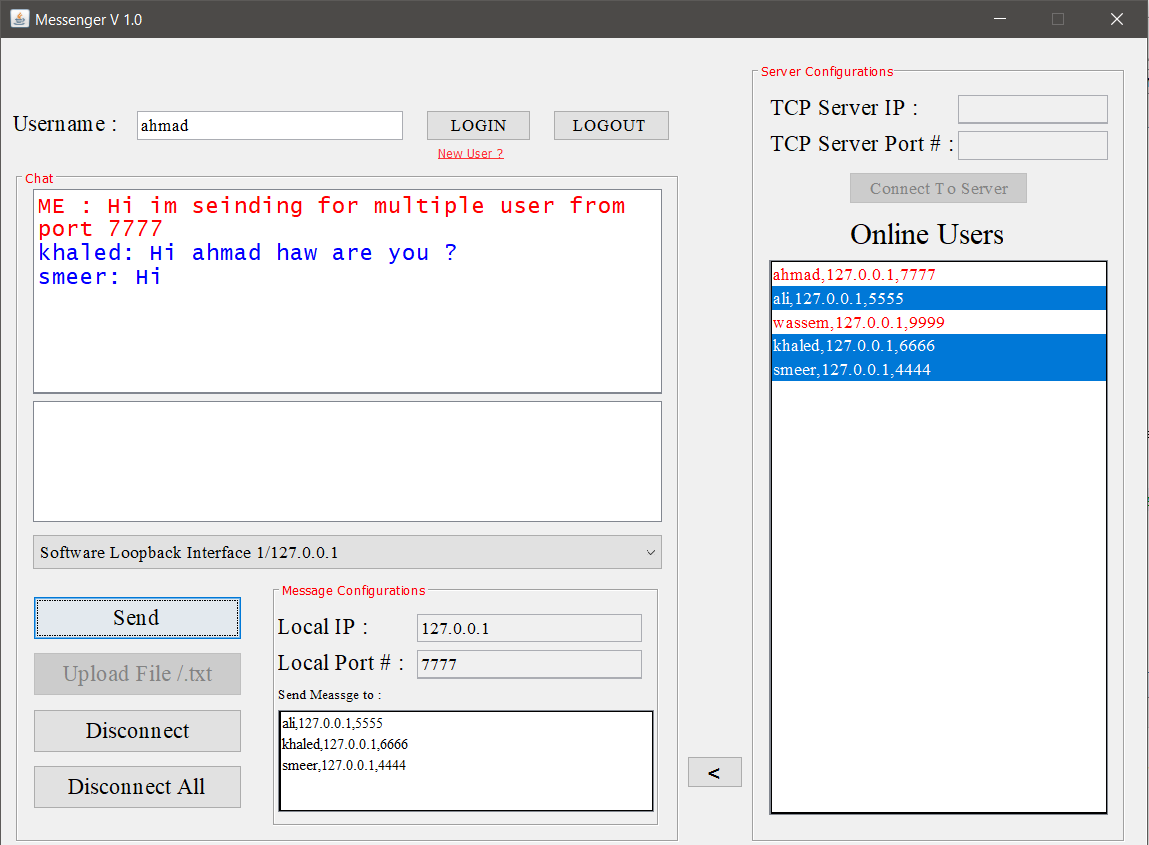
User 1 : User 2:



User 3 : Sender :

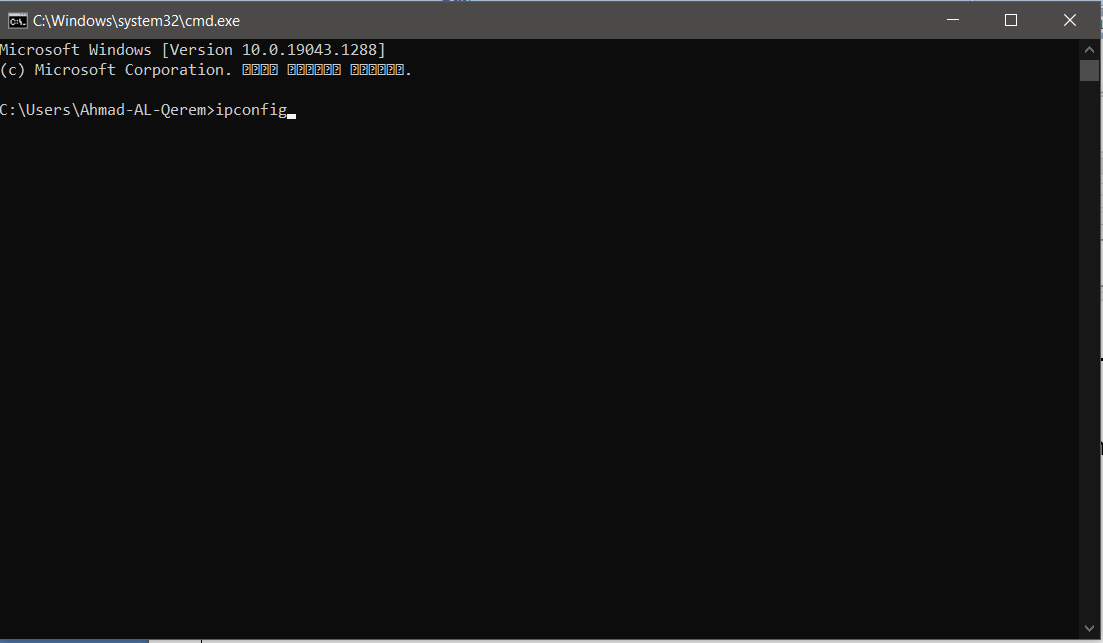
1. Now some users will send for Ahmad and Smeer :

User 1 : User 2:

User 3 : User 4 :

* Send from deferent computer at local area net work ( not local host ):

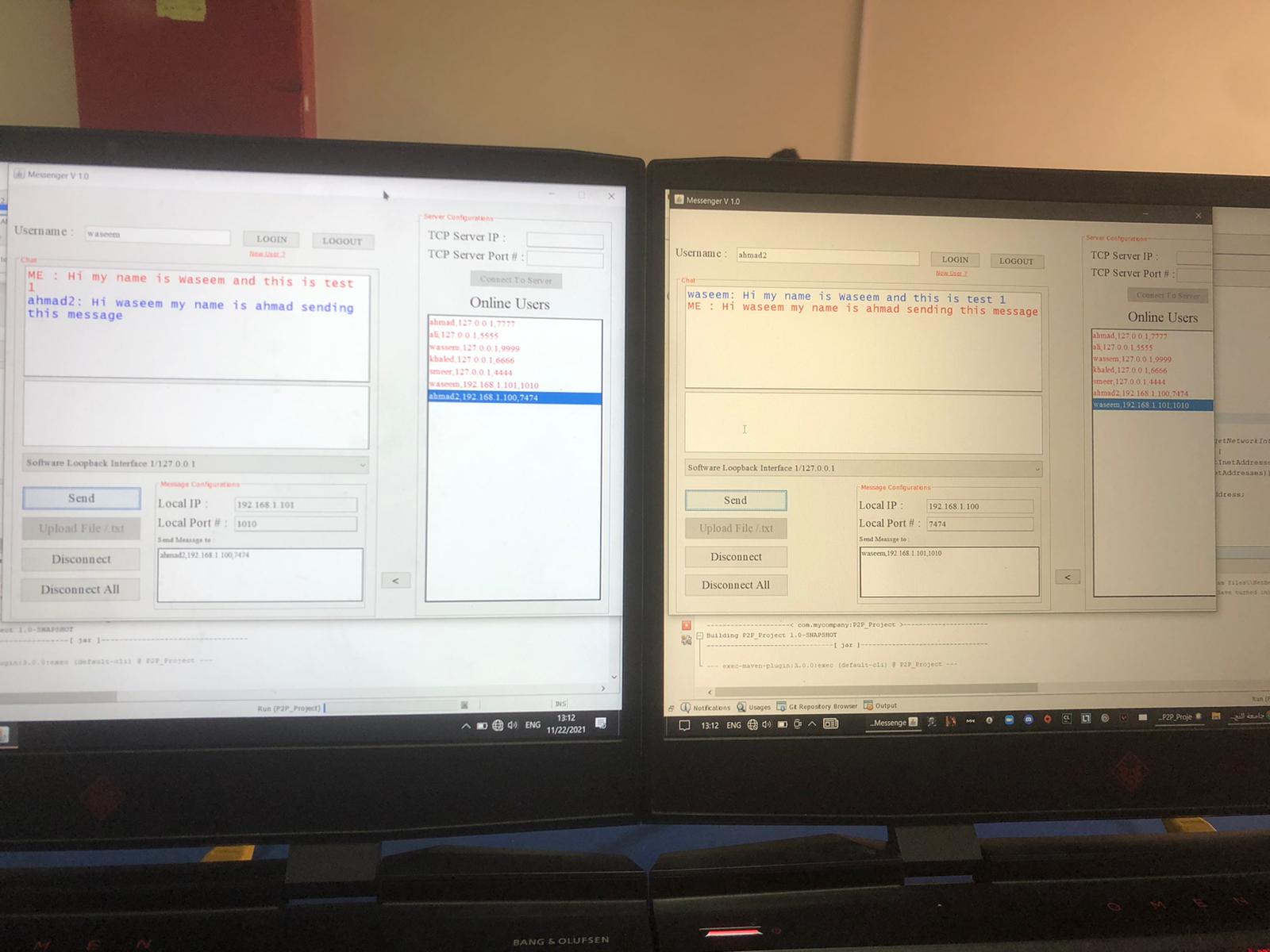
At first I will get my IP address from **CMD** using **ipconfig** command as shown :



And I got **IP 192.168.1.100**

The same in the other computer which have **192.168.1.101** now sign up with new user with these **IP** .

Then we tested the **P2P** project :



* Note : the two devices are connected with Ethernet cable .

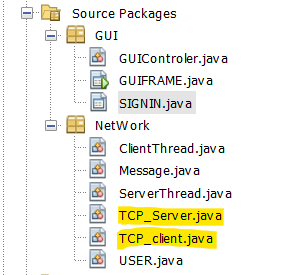
Part 2 :

1. Online User List :

In this part we created a server to listen for all users who sign in or up and send a list of all online users to the other users.

* Client side :

We added more classes as shown :

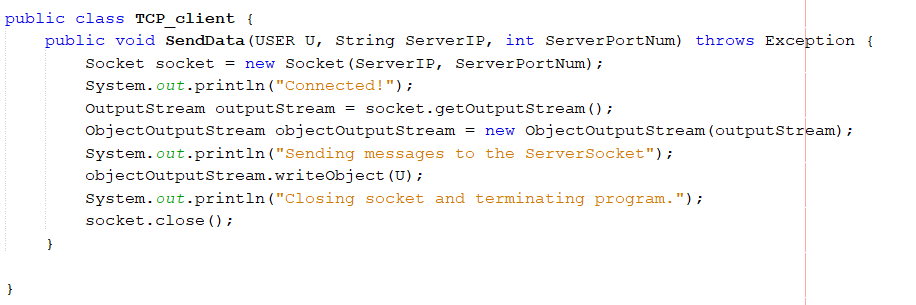


Such that TCP\_Server.java work as listener in parallel at background for real server .receiving a list of online users and show this list at his GUI .

As shown in the figure below :

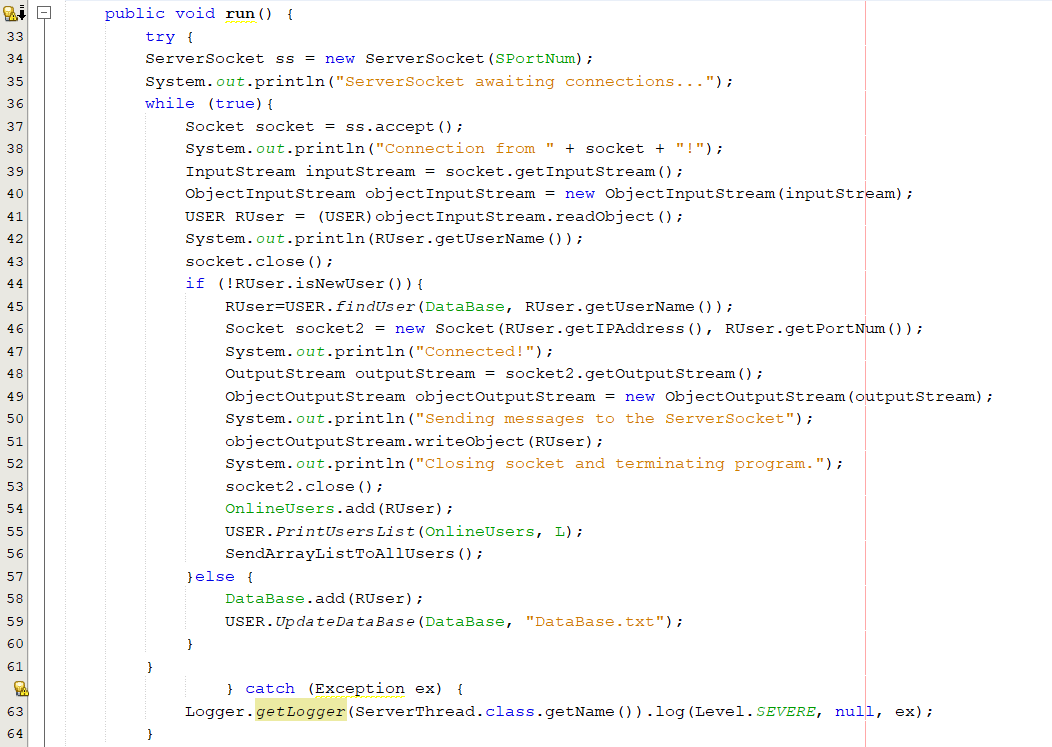
 act as receiver in client side . such that client will receive his own data ( IP address , Port number …etc ) from server side. Then will receive an ArrayList of online users and display them at his own list .

TCP\_client.java contain sender function as shown below :

the function send user that want login at socket with IP " ServerIP " at port " ServerPortNum ".then the server will check his data base if this user exist .

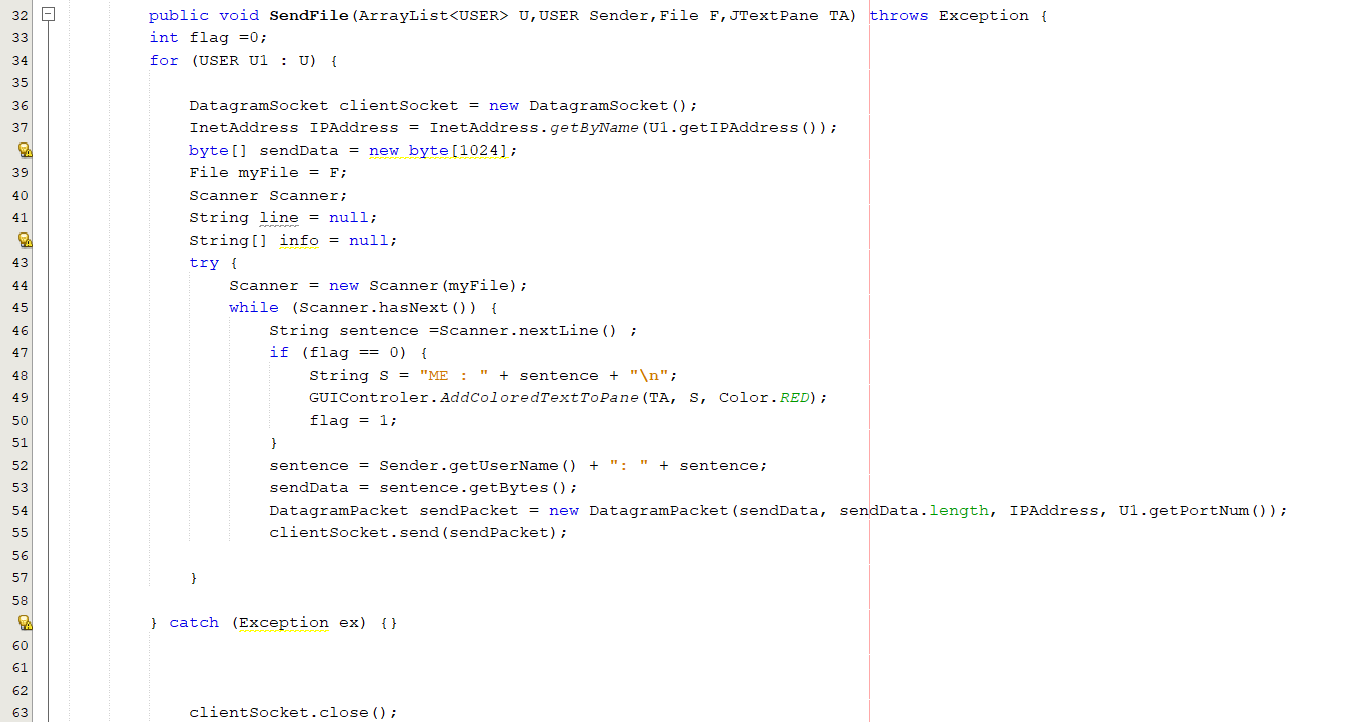
When find this user it will add him to online user list and send this list to all online user .

* Server Side :



Will run in back ground keep listing for clients who logging in .

1. Send text file :

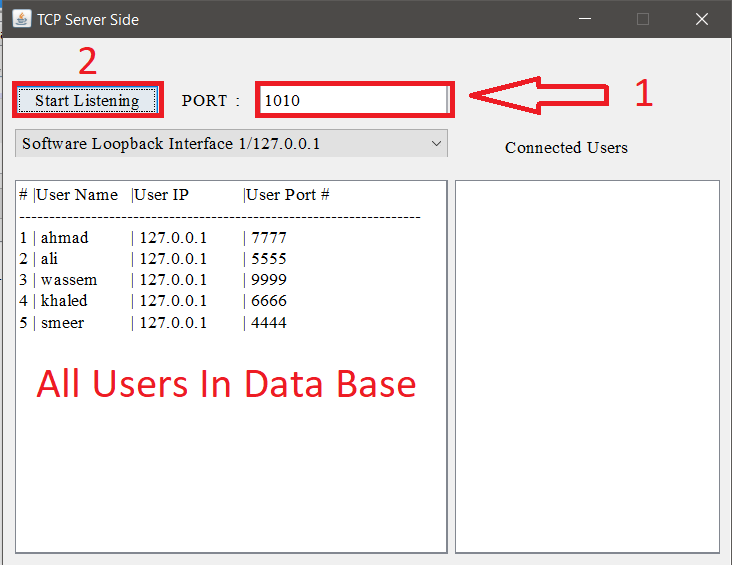


Send file function accept File instead if message .read this file and send its content line by line to the other side .

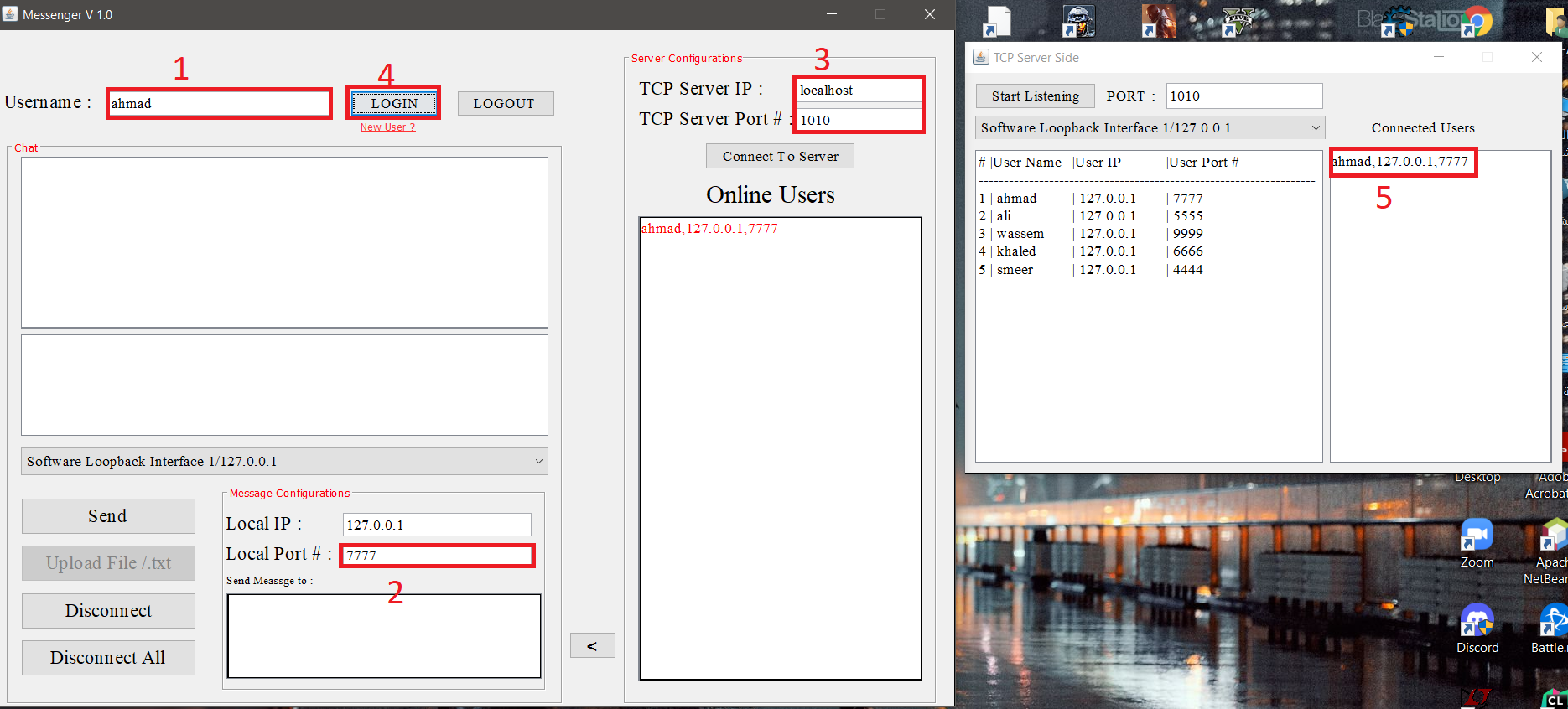
1. Testing :

* Online user :

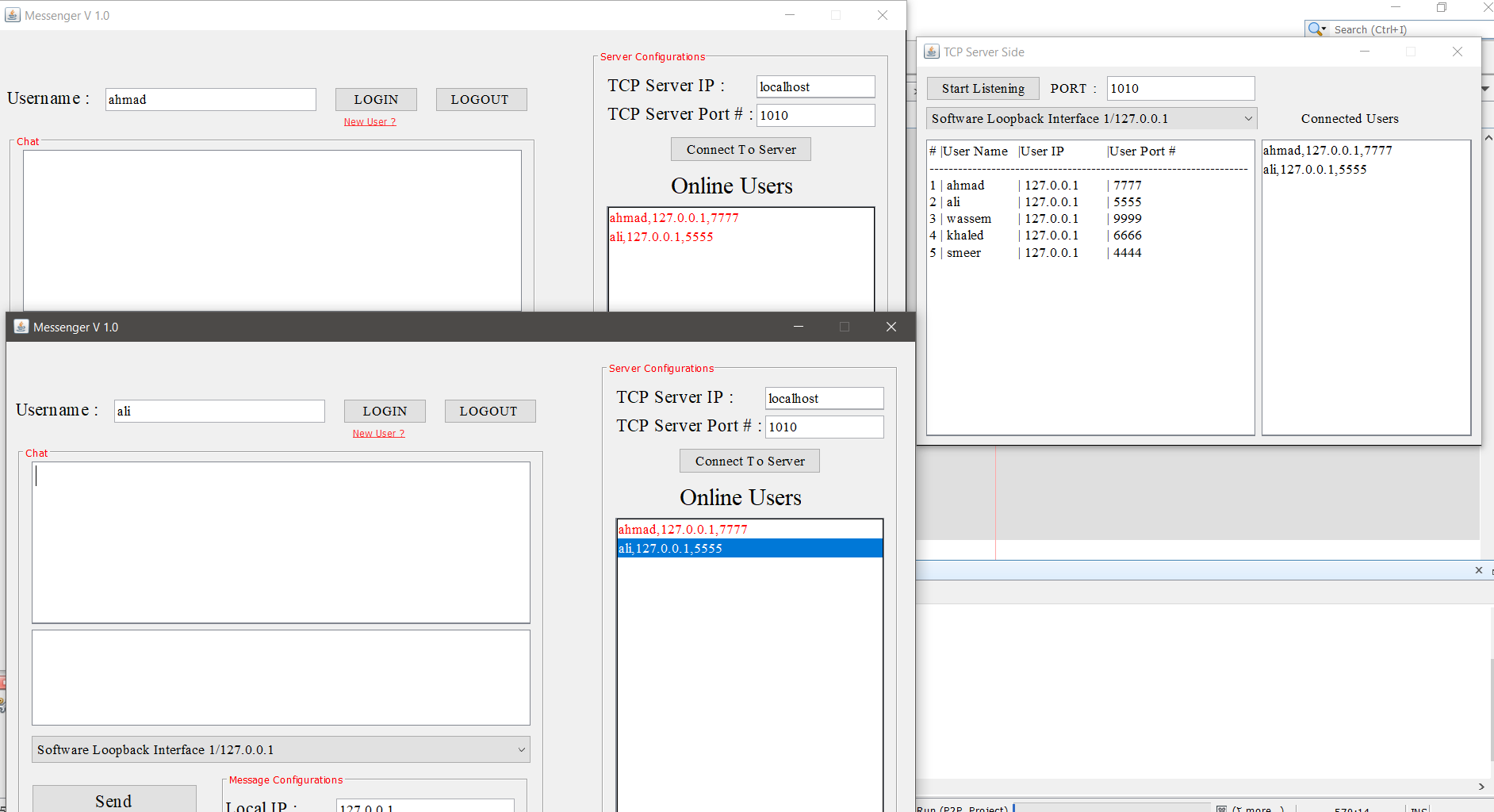
1. Server active and start listing



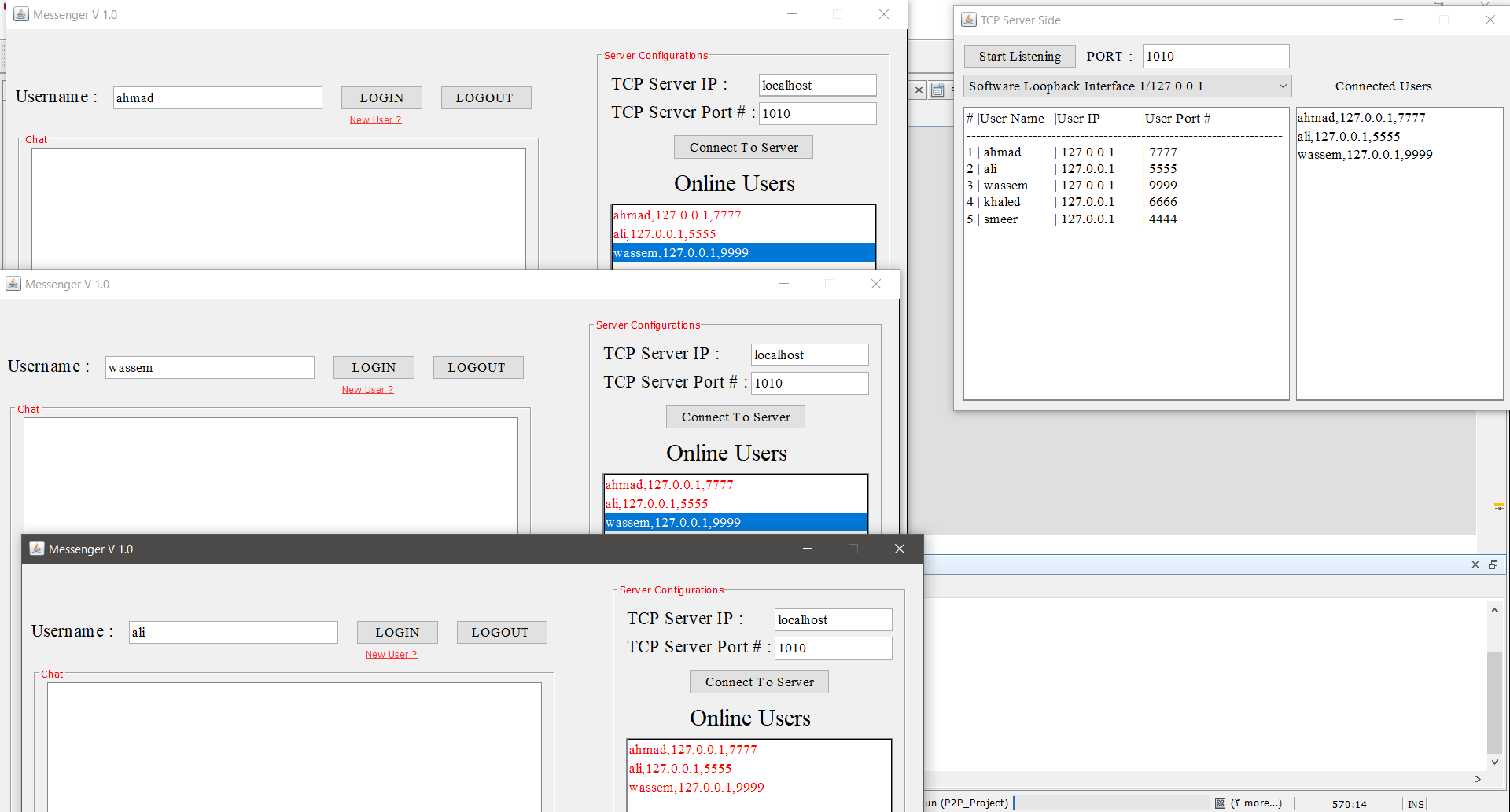
1. Ahmad logged in



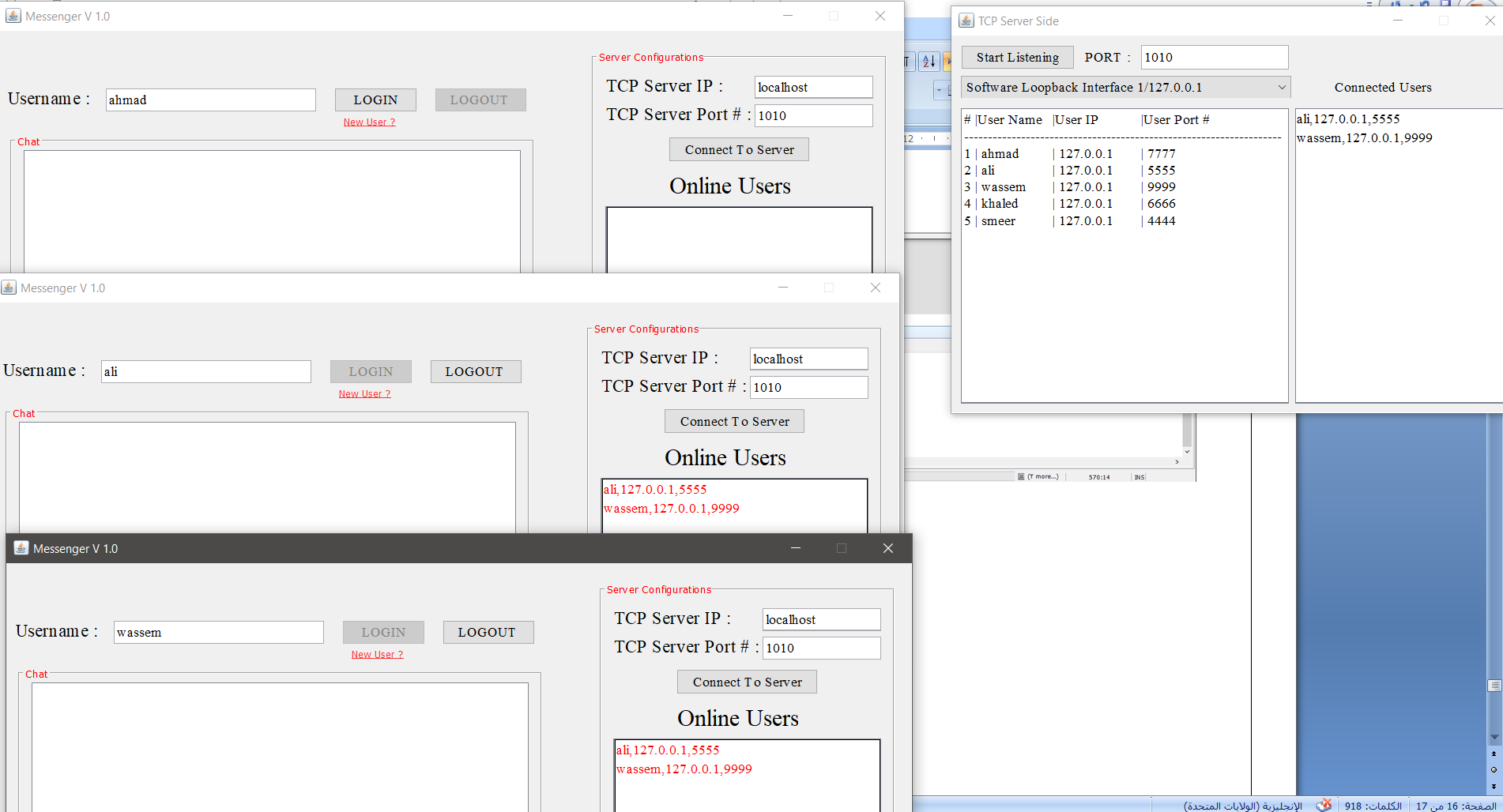
1. Ali logged in



1. Wassem logged in

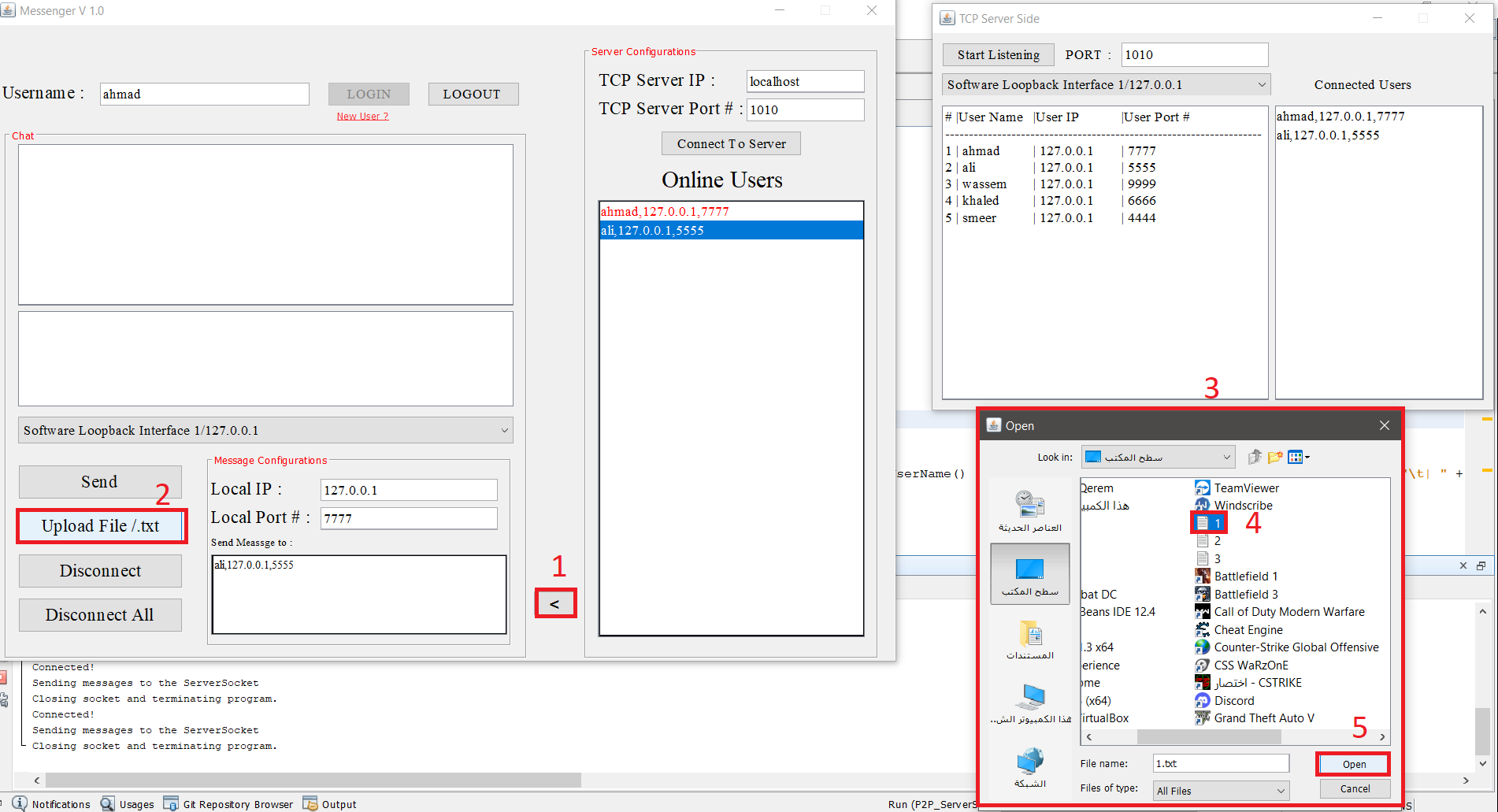


1. Ahmad logged out



* File sender :

Assume ahmad want send file called 1.txt contain "11111111111…" to ali



What ali receive :

