*Multi-Client LAN Chat Messenger Application*

Amith Bhat Nekkare ,181IT105

Department of Information TechnologyNational Institute of Technology Karnataka, SurathkalMangaluru, India  
[amithbhat01@gmail.com](mailto:amithbhat01@gmail.com)

Samarth S Hadimani,181IT240

Department of Information TechnologyNational Institute of Technology Karnataka, SurathkalMangaluru, India  
connect.samarth@gmail.com

Arya Sharma,181IT207

Department of Information TechnologyNational Institute of Technology Karnataka, SurathkalMangaluru, India  
[arya.sharma@gmail.com](mailto:arya.sharma@gmail.com)

***Abstract*— Socket programming is a field in networking where two or more nodes can be connected to each other so as to allow communication among them. In this paper, we propose an object-oriented LAN Chat Messenger application that can allow people connected through a LAN to communicate among themselves, using socket programming. This has been implemented using a multi-client to server model, instead of a client-to-client model, due to the ease of implementing the operations involves. With this system, users can send text, images and files of all types to each other. To ensure security of data related to user accounts such as username, passwords, etc. , a Caesar cipher has also been employed. To ensure the synchronization of the server sockets with each of the client sockets, we have used multi-threading concepts as well. Our proposed application can also store messages and display it to the receiver if he is not online at the time of sending of message. To ensure that messages are not lost, in our model, the server also does not immediately delete messages once they have been receives; it maintains a copy until some time has passed. This allows the user to re-access it, should he wish to do so.**

**Keywords—LAN Messenger, Client-Server Model, Socket Programming, Caesar cipher, Object-Oriented programming, Multi-threading.**

# Introduction

A LAN Chat Messenger is an useful tool for communication among people who are using the same LAN. They help quite a lot in facilitating communication between people. In this century, instant communication has become a necessity to go about life normally, hence there will always be a relevance for chat applications and related programs in the IT domain[2].

Keeping this in mind, we have built a user-friendly object-oriented LAN chat messenger application, which can be used to send private messages of different types to other users, demonstrating the application of networking and socket programming principles in real-life scenarios[1].

Our project has the obvious application of being used in office and other work environments for easy and quick communication between workers working on a project[2].

II. PROBLEM STATEMENT & OBJECTIVES

1. To develop a secure, object-oriented, continuous and two-way multi-client LAN messenger application, using socket and multi-threading concepts.
2. It will be able to send files and images, along with text messages. It should support all major file and image types.

III. LITERATURE SURVEY

We have referred multiple papers during the creation of our messenger tool. They have been mentioned below:

Paper [1] is “An Internet Messenger System for Local Area Network”, by W.Bandara,A.Karunaratne,R Rajapaksa and

R.Ragel. In this paper, a LAN messenger system has been proposed that exploits all available resources and gives the expected privacy. Their system even allows for private chat, chat rooms, video messaging,file transfer, etc.

Paper [2] is “Development of LAN Chat Messenger (LCM) Using Rational Unified Process (RUP) Methodology with Object Oriented Programming”, by I.M. Abba, M.Cruz, U. Eaganathan, and J. Gabriel. In this paper, an object-oriented LAN Chat Messenger using RUP methodology provides low-cost communication between staff in an organization has been proposed. An advantage of this design is that allows users to chat on networks outside the internet boundaries.

In paper [3], “ Implementation of Chat Application- Ichat”, by R.Chate, M.Shivshankar ,Jyothi B., a LAN Chat Application was built by the authors, using the principles of socket programming and session management.

In paper [4], “A Study of Internet Instant Messaging and

Chat Protocols”, by R.B. Jennings, E.M. Nahum,

D.P. Olshefski, D. Saha, Z.Y.Shae, and C.Waters , a comparison of different protocols commonly used in instant messaging applications and chatbots was performed.

IV. SYSTEM DESIGN & ARCHITECTURE

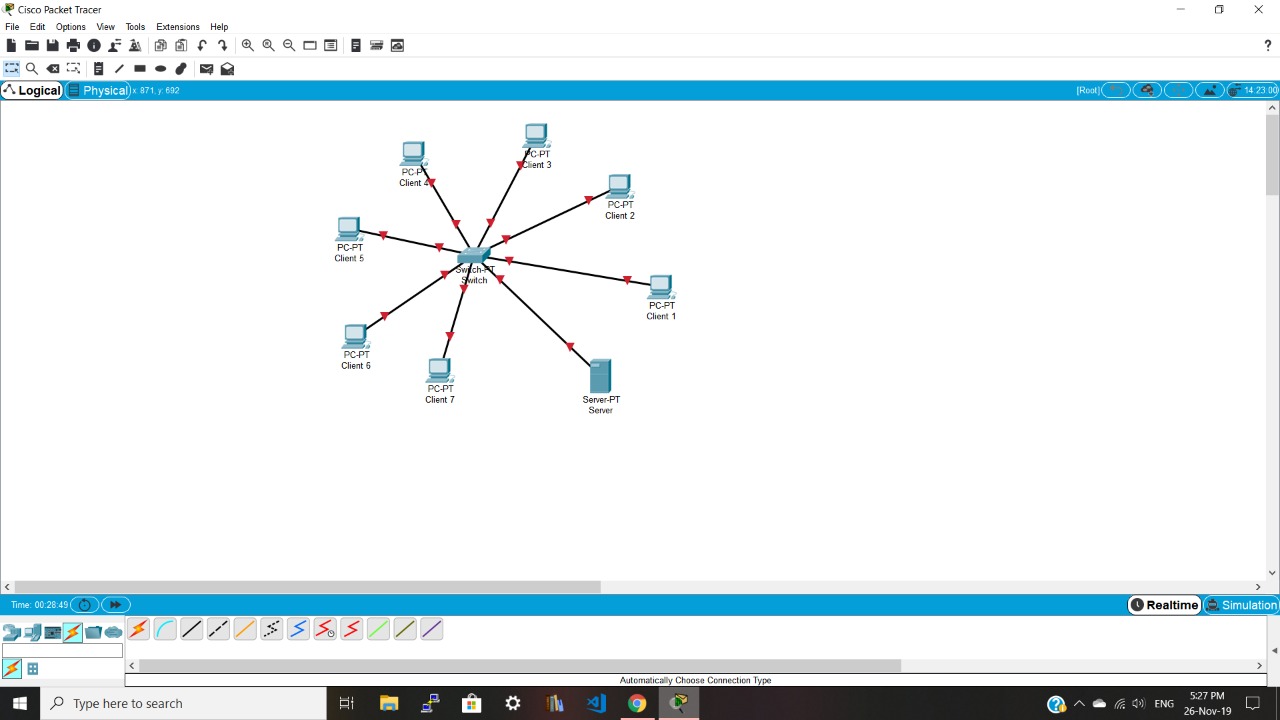


Fig 1. System Design

In our messenger tool, we envision a multi-client server chat application connected in the manner shown above[3]. The message(may be text, file or image) may be sent from one client to another client connected to the main server. The message will reach the server, where it will be stored till the client logs into his system(if he hasn’t already). The message will then be sent to him, where he can view it.

In our system, the server will store a limited number of messages sent by any client to another, ie , it will not delete a message immediately after it has been sent to the receiving client.

Also, we have added encryption to the passwords stored in the server, to ensure security of the client account.

V. IMPLEMENTATION

A. System Specifications

* Hardware Requirements

- Processors: Will run on Intel Pentium IV and newer

- Hard Disk : 1 TB and above

- RAM : 2 GB and above

* Software Requirements

- OS : Ubuntu 10.04 and upwards

- Technologies used : Core Java

B. Methodology

In this project, we have built a multi-client server chat application. When the user opens the application, the client program connects to the server by using a unique port number and IP address to create a TCP socket[4]. If there is a problem in the connection or socket, an error is shown. Now, the user is greeted by a welcome interface, which asks them to log in to the messenger, or to sign up, if they do not have an account.

Upon signing up, they are prompted to enter a username and password. If another account has already used the entered username, the user is prompted to enter another username/password. After the entry of the correct username and password, it is encrypted using a Caesar encryption algorithm at clientside, and sent to the server, where it is stored in the username and password database.

The Caesar cipher is a type of substitution cipher, where each letter of the alphabet will be shifted by a constant value. For example, with a shift of 2, A becomes C, C becomes E, etc.

Now, the welcome interface is called again, and the user will now log in. The username and password entered by the user is compared with the ones called from the server. If one set match completely, the user is allowed to access the account. Else, the username/password match error is displayed to the user, and he is asked to try again.

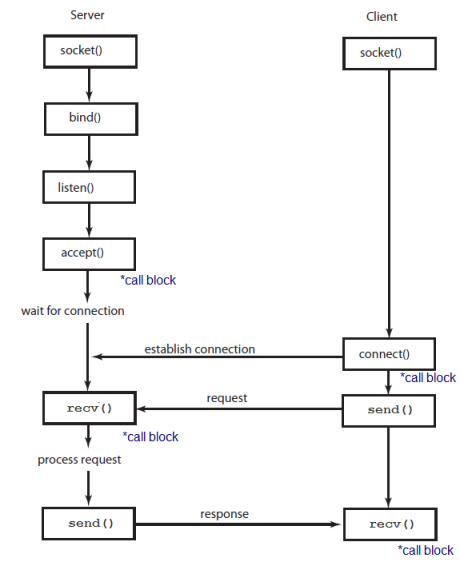


Fig 2. Connection Establishment for Socket

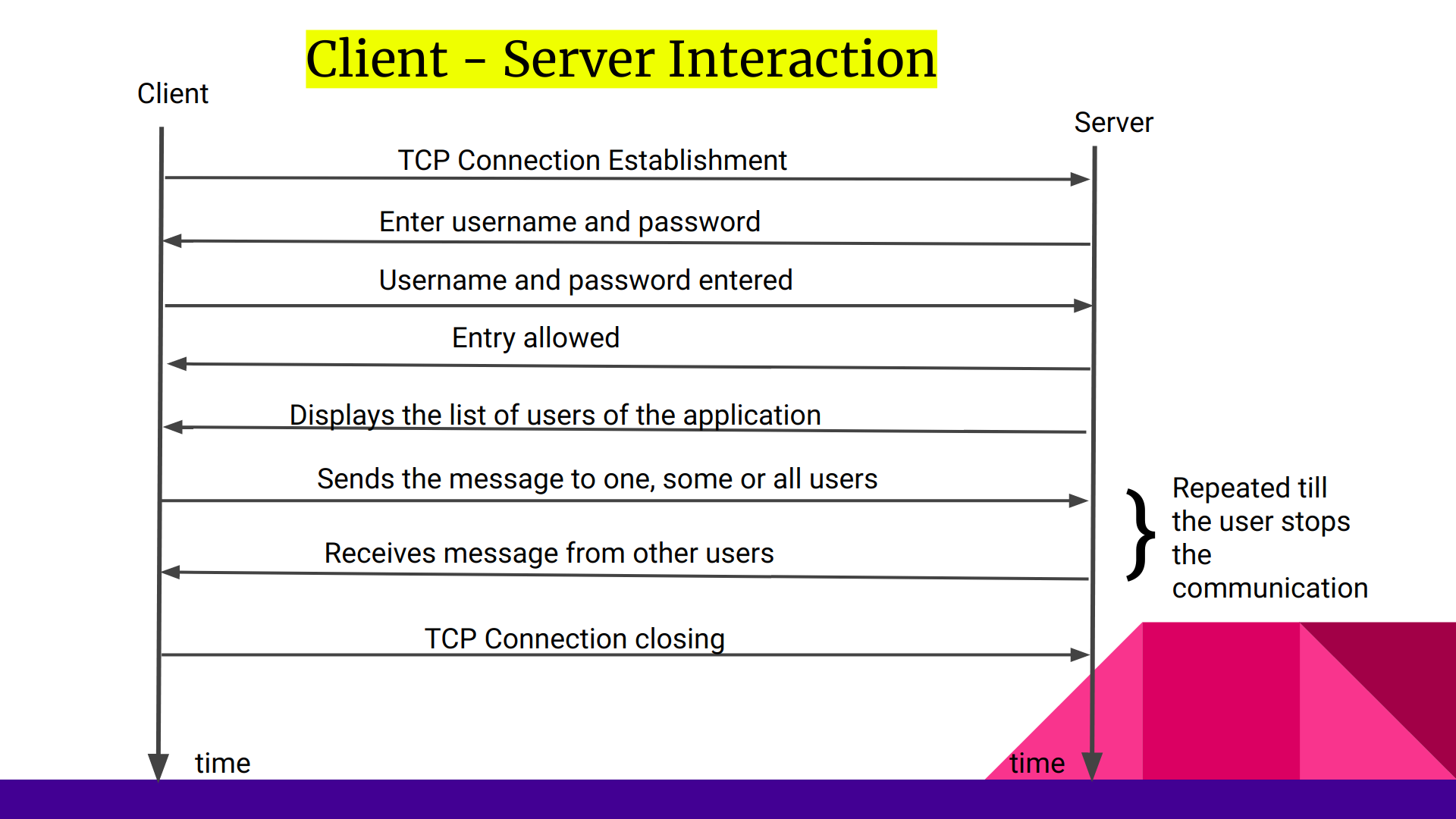


Fig 3. Client-Server Interaction

Once the user enters the main account, they have four options : to send a message[ to one, two, or many receivers], to send a file or image, to check their inbox for a message or to check their inbox for a file/ image.

To ensure the synchronization of server sockets with client sockets, we have used multi-threading concepts. In our system, the server maintains a thread for each client to synchronize between them easily.

When the user wants to send a message, he is asked the number of receivers he wants for the message being sent.

Then, a one to one interaction takes place between the objects initialized by the different clients, and the message is transferred. For example, if user1 would like to send a message to user2 , the message sent by user1 to user2 passes through the socket connecting user1 and the server. Here, the object initialized by user1 interacts with user2’s object, and the message is transferred through user2’s object.

Similarly, when user2 checks his inbox, user2’s object transfers the message to the user2 client computer through the socket connecting user2 and the server computer.

When the user wants to send a file/ image, the file is broken down into a stream of bytes, which is then stored in a byte array of sufficient length. This is then transferred over to the server through the socket connection.

Similarly, when the receiver checks his inbox for any files, the file is received as a byte array, which is then reassembled into the complete file. In this manner, we can send all types of files and images, as the specific encoding of each type is not changed.

At any point, the user has the option to exit the chat application.

VI. RESULTS AND ANALYSIS

A program for the above chat application was written using Java, and run on an Ubuntu 16.04 machine. The results obtained were as follows :

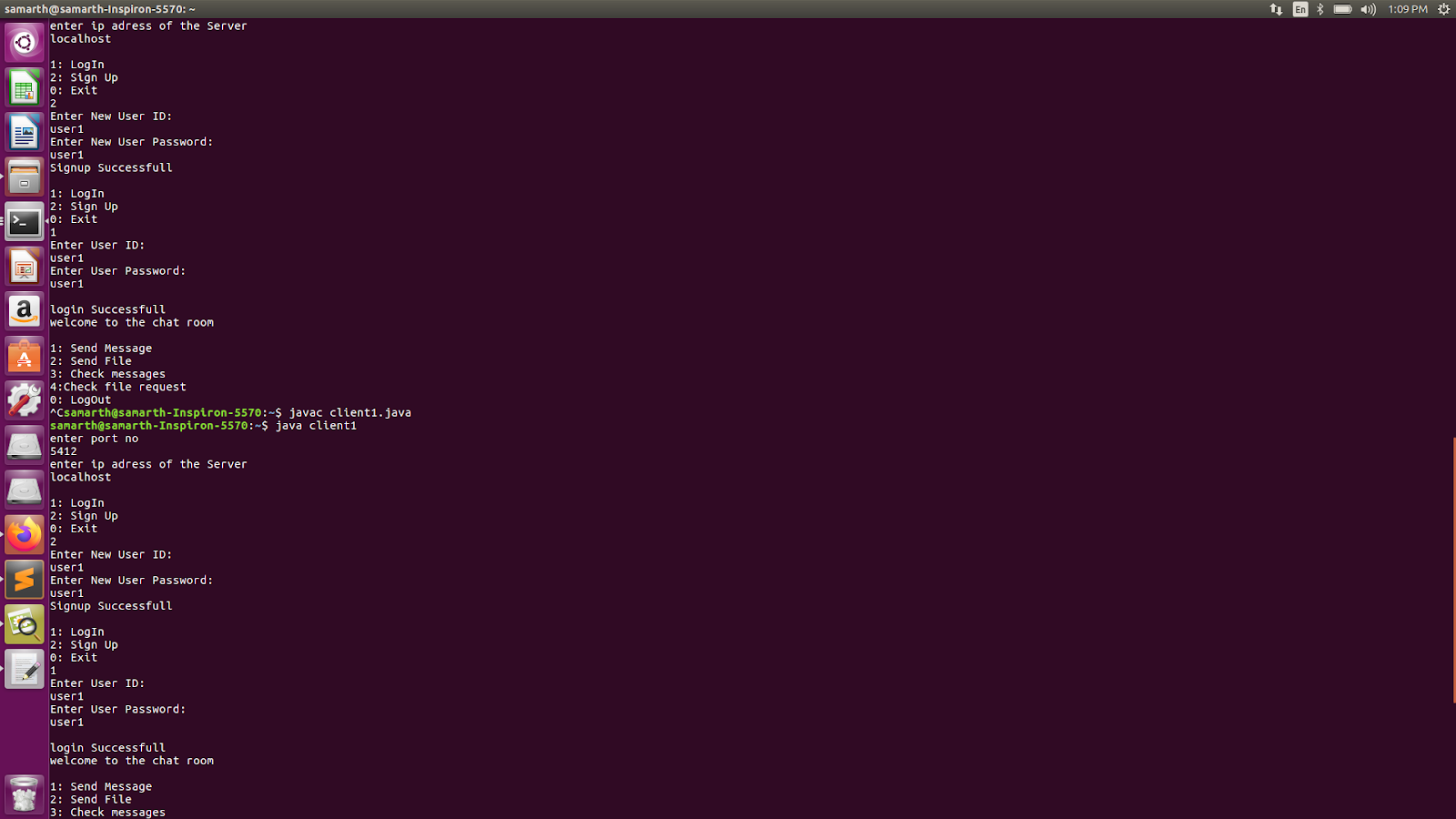


Fig 4. Welcome Interface - Client

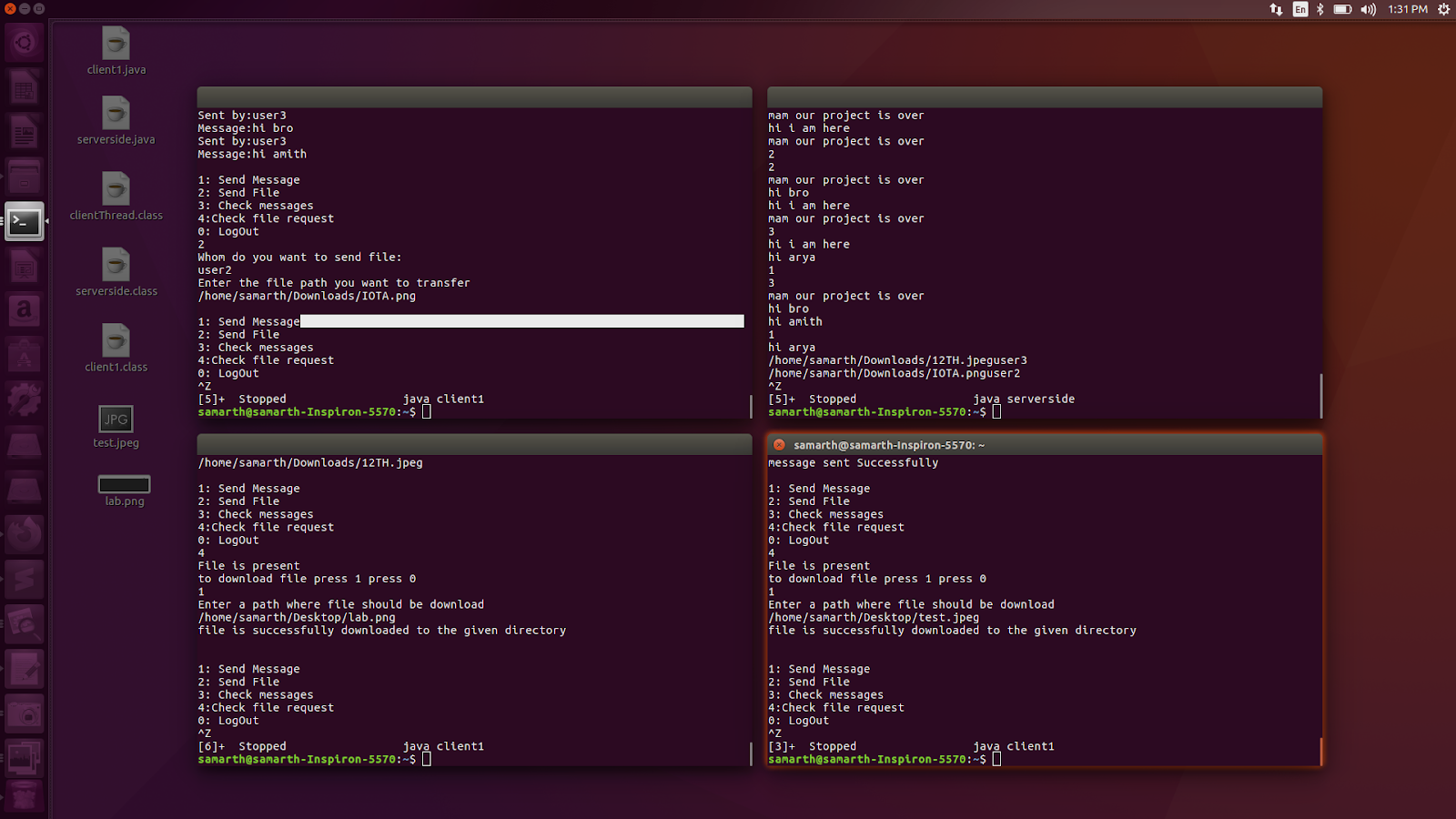


Fig 5. Text Messaging and File Sending – Client

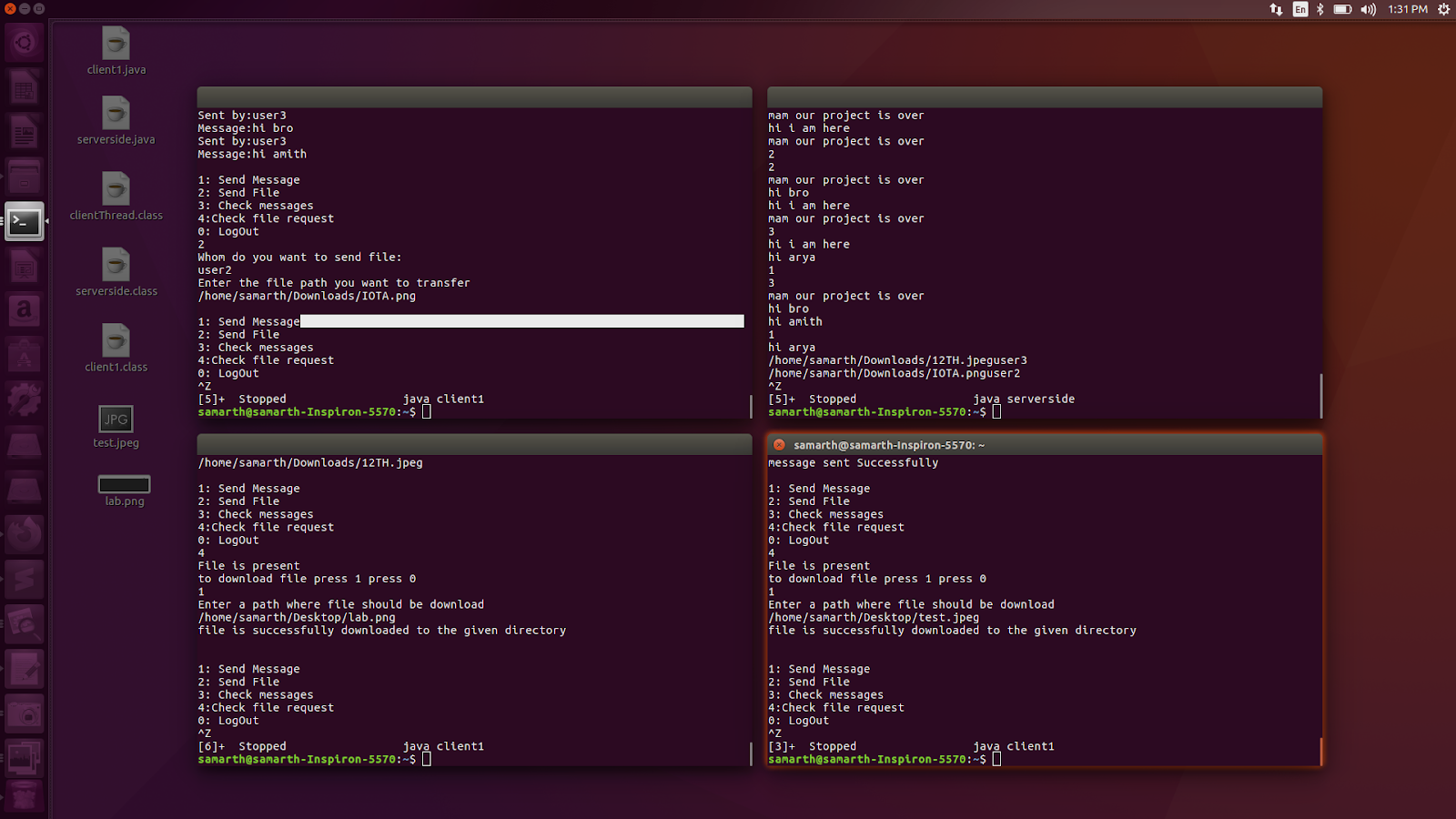


Fig 6. File Receiving- Client

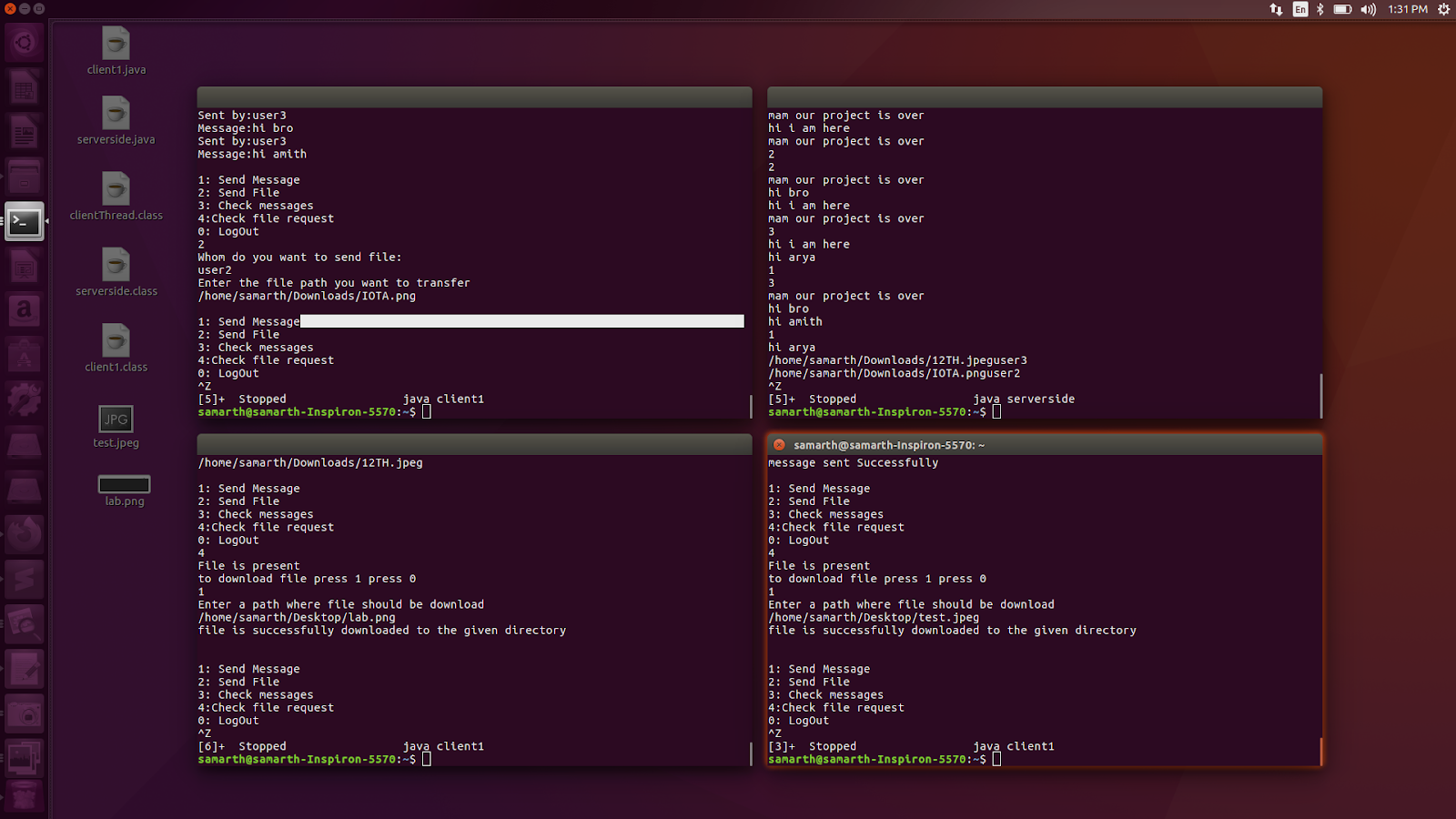
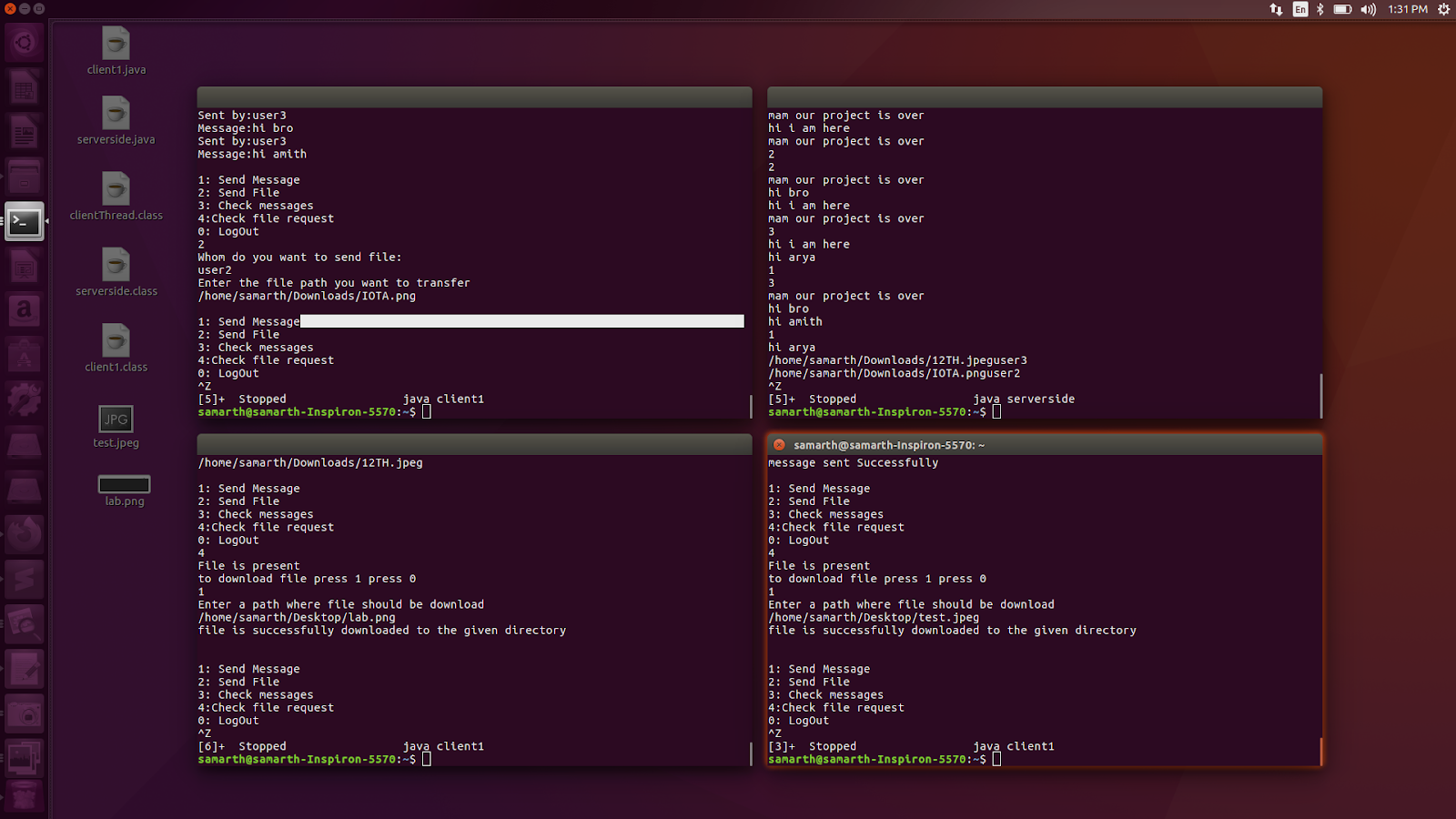


Fig 7. Server

Fig 8. Image Receiving - Client

VII. CONCLUSION

In this project, we have demonstrated the effectiveness of using a secure, two way, multi-client LAN chat messaging application as a medium for instant communication.

LAN Chat Messengers are very useful in daily life, as they help in instant communication between people. In this century, instant communication has become a necessity to go about life normally, hence there will always be a relevance for chat applications and related programs in the IT domain. Keeping this in mind, we have built this project.

REFERENCES

[1] W.Bandara,A.Karunaratne,R Rajapaksa and

R.Ragel, An Internet Messenger System for Local Area Network, Proceedings of the Pcradcniya University Research Sessions, Sri Lanka, Vol.114, 3 December 2009.

[2] I.M. Abba, M.Cruz, U. Eaganathan, and J. Gabriel,Development of LAN Chat Messenger (LCM) Using Rational Unified Process (RUP) Methodology with Object Oriented Programming, Proceeding of the International Conference on Artificial Intelligence in Computer Science and ICT(AICS 2013), 25 -26 November 2013, Langkawi, Malaysia.

[3] R.Chate, M.Shivshankar ,Jyothi B. ,Processing and Computing Technologies- Implementation of Chat Application- Ichat, International Journal of Science, Engineering and Technology Research (IJSETR)

Volume 6, Issue 4, April 2017,.

[4] R.B. Jennings, E.M. Nahum, D.P. Olshefski, D. Saha, Z.Y.Shae, and C.Waters , A Study of Internet Instant Messaging and Chat Protocols , IEEE Network, Volume: 20 , Issue: 4 , July-Aug. 2006.

INDIVIDUAL CONTRIBUTION

Amith Bhat Nekkare - Client Side Programming

Samarth S Hadimani - Server Side Programming

Arya Sharma – Encryption and database management