A PROJECT REPORT ON

**CLOUD COMPUTING USING INTRANET**

**SUBMITTED BY :**

UTKARSH KORE (10)

NIRAJ PALKAR (14)

HARSH TARKAR (15)

OMKAR DONGRE (22)

**PROJECT GUIDE :-**

MRS. POONAM WAVARE

**DEPARTMENT OF COMPUTER ENGINEERING**

2016 – 2017



**VIDYA PRASARAK MANDAL’s POLYETECHNIC ,**

**CHENDANI BUNDER ROAD , THANE – 400 601**

**V.P.M’s POLYTECHNIC , THANE**

**Department of Computer Engineering**

**VISION**

To produce comprehensively trained socially responsible , innovative computer engineers of highest quality to contribute to the nation’s development .

**MISSION**

1. Provide sound technical foundation in Computer Engineering through comprehensive curriculum and application oriented learning .
2. Provide ambience for professional growth and lifelong learning for adapting to challenges in rapidly changing technology.
3. Inculcate social and ethical values.
4. Nurture the interpersonal and entrepreneurial skills so as to develop leadership qualities in information industries diverse culture.

**ACKNOWLEDGEMENT**

The completion of a project is a milestone in student life and its execution is inevitable in the hands of guide. We are highly indebted the project guide “**MRS. POONAM WAVARE**” for her invaluable guidance and appreciation for giving form and substance to this report. It is due to her enduring efforts , patience and enthusiasm which has given a sense of direction and purposefulness to this project and ultimately made it a success.

We would like to tender our sincere thanks to the staff members and H.O.D. “**MRS. S.K.SHUKLA**” for their co-operation.

We would like to express our deep regards and gratitude to the PRINCIPAL “**Prof. D.K.NAYAK**”.

We are also thankful to our parents for promoting and motivating us regarding our project development.

We would wish to thank the non-teaching staffs who have helped us all the way in one way or the other.

Really it is highly impossible to repay the debt of all the people who have directly or indirectly helped us performing the project.

**PREFACE**

We have made sincere attempts and taken every care to present this matter in precise and compact form , the language being as simple as possible.

We are sure that the information contained in this volume would certainly prove useful for better insight in the scope and dimension of this project in its true perspective.

The task of completion of the project though being difficulty was made quite simple, interesting and successful due to deep involvement and complete dedication of our group members:

* Utkarsh A. Kore
* Niraj K. Palkar
* Harsh L. Tarkar
* Omkar S. Dongre

**ABSTRACT**

The project that is to be developed is a communication between a cloud server and number of clients. Communication between different clients is done through sockets. This can be implemented using the sockets and ServerSocket classes. The requested choice is sent to the server and the communication takes place .The server is anything that has some resource that can be shared. A client is simply and other entity that wants to gain access to a particular server. The requests of clients will be analyzed by packet filter and log files will generated on the server side . Clients can do file sharing with proper authentication .

**TABLE OF CONTENTS :**

Chapter 1

**INTRODUCTION**

* 1. Introduction
  2. Feasibility

Chapter 2

**PROJECT STATEMENT**

2.1 Problem Statement

2.2 Problem Description

Chapter 3

**SYSTEM REQUIREMENTS AND SPECIFICATIONS**

3.1 System Requirement

3.1.1 Software Requirement

3.1.2 Hardware Requirement

3.1.3 Modules

3.2 System Functionality

Chapter 4

**SYSTEM DEVELOPMENT**

4.1 System Architecture

4.2 Flowchart

4.2.1 Flowchart for Server and Client System

4.3 Data Flow Diagram

4.3.1 DFD Level 0

4.3.2 DFD Level 1

Chapter 5

**IMPLEMENTATION**

Chapter 6

**TEST CASES**

Chapter 7

**SUMMARY , FEATURES & ADVANTAGES**

Chapter 8

**FUTURE SCOPE**

Chapter 9

**BIBLIOGRAPHY**

Chapter 1

**INTRODUCTION**

* 1. **INTRODUCTION :**

Present day industries are increasingly shifting towards Cloud platforms. Cloud is a type of Internet based computing that provides shared computer processing resources and data to computer and other devices on demand.

The advantages of cloud can be summarized as:

1. Cost Saving
2. Reliability
3. Manageability
4. Strategic Edge

Now, an intranet is a private network that is contained within an enterprise. It may consists of many interlinked LAN’s and also use leased lines in WAN. Typically, an intranet includes connections through one or more gateway computers to the outside internet. The main purpose of intranet is to share company information and computing resources among employees.

Intrusion Detection System is a device or software application that monitors a network or system for a malicious activity or policy violence. Any detected activity or violation is typically reported either to an admin or collected centrally using **Security Information and Event Management system.**

* 1. **Feasibility :**
     1. **Software Feasibility -**

The software system is able to perform adequately constructing the proposed system. The technology needed for the system exists and is simple to understand and firm has enough experience using that technology. The success rate of the software goes up to 90% .

* + 1. **Economical Feasibility –**

The system analysts estimate the cost for new system as INR 10,000 one time investment for developing, updating hardware and user training. Also, firm needs to pay INR 4,000 each year for software maintenance and supplies. On the other hand the system will provide approximately INR 30,000.

Chapter 2

**PROJECT STATEMENT**

**2.1 Problem Statement :**

To provide an efficient security in real time prevention and detection of Intrusions for Cloud services which are flexible and easy to implement , configure and administrate .

**2.2 Problem Description :**

**Cloud Computing Using Intranet** helps in preventing and detecting the Intrusion attacks and to develop the filter based Model which will consume less computational resources and memory space by designing intelligent and flexible architecture for filtering suspicious activities and detecting attacks . Using the model each client can send the service request and get response over the Intranet . When the client requests for service , it will be passed through the packet filter and then the request will be passed to the server by which log files will be generated and activities will be analyzed.

Chapter 3

**SYSTEM REQUIREMENT AND SPECIFICATION**

**3.1 System Requirement :**

3.1.1 Software Requirement-

* Microsoft Windows XP or Higher
* Java(JDK 1.7.0) or Higher version
* NetBeans IDE
* Microsoft Access 2007 or Higher

3.1.2 Hardware Requirements –

* Computers
* LAN Connection
* Processor Pentium or Higher
* RAM 550 MB or Higher
* Speed 500MHz or Higher
* Input Devices (Keyboard and Mouse)

3.1.3 Modules –

* Server
* Client

Chapter 4

**SYSTEM DEVELOPMENT**

**4.1 Architecture :**

Intranet

**C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0292020.wmf C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0292020.wmf C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0292020.wmf**

Requests

Requests

Requests

Packet Filter And Packet Analyzer(IDS)

Log Files

Database

**4.2 Flow Chart :**

**Client Server Client**

Socket

Server Socket

Client

Socket

Client

Socket

Request/Reply

Socket

Packet Filter **Filtered Message**

**NETWORK**

# Request/Reply

Client Server Communication :

SERVER

CLIENT

Connection Success

**4.3 Data Flow Diagrams :**

Client asks for services from cloud

Check for space

Client

Cloud storage

Cloud

Verification

Directory

Allocated space

**Fig . Level 0 DFD**

Client

Cloud storage

Management

Check request

Client asks for services from cloud

Directory

Allocated space

1

2

Client info database

Cloud Storage

Check for valid info

Check for space

3

Provided Services database

Update storage info

**Fig . Level 1 DFD**

Chapter 5

**IMPLEMENTATION :**

**Server Code –**

package chat\_server;

import java.io.\*;

import java.net.\*;

import java.nio.file.\*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.\*;

import java.util.logging.FileHandler;

import java.util.logging.Level;

import java.util.logging.Logger;

import java.util.logging.SimpleFormatter;

import javax.crypto.Cipher;

import javax.crypto.SecretKey;

import javax.crypto.spec.SecretKeySpec;

import javax.swing.JFileChooser;

import javax.xml.bind.DatatypeConverter;

/\*import jpcap.\*;

import jpcap.NetworkInterface;

import jpcap.packet.DatalinkPacket;

import jpcap.packet.EthernetPacket;

import jpcap.packet.ICMPPacket;

import jpcap.packet.IPPacket;

import jpcap.packet.Packet;

import jpcap.packet.TCPPacket;

import jpcap.packet.UDPPacket;

\*/

public class server\_frame extends javax.swing.JFrame

{

ArrayList clientOutputStreams,cOutputStreams;

ArrayList<String> users;

String encrypt,decrypt;

File sendf;

// rec r=new rec();

int l\_inc,f\_inc,s\_inc;

public class ClientHandler implements Runnable

{

BufferedReader reader;

Socket sock;

PrintWriter client;

public ClientHandler(Socket clientSocket, PrintWriter user)

{

client = user;

try

{

sock = clientSocket;

InputStreamReader isReader = new InputStreamReader(sock.getInputStream());

reader = new BufferedReader(isReader);

OutputStream os = sock.getOutputStream();

cOutputStreams.add(os);

}

catch (Exception ex)

{

ta\_operations.append("Unexpected error... \n");

}

}

@Override

public void run()

{

String message, connect = "Connect", disconnect = "Disconnect", chat = "Chat", receiveCloud="ReceiveCloud",sendCloud="SendCloud",refresh="Refresh" ,cloud="Cloud";

String[] data;

try

{

while ((message = reader.readLine()) != null)

{

decrypt = decryptText(message);

createlog(decrypt + "\n");

data = decrypt.split(":");

//ta\_operations.append("Received: " + data[0] + data[1] + "\n");

if(!data[2].equals(refresh) && !data[2].equals(sendCloud) && !data[2].equals(cloud))

{

ta\_chat.append(data[0] + ": " + data[1]);

ta\_chat.append("\n");

refreshOl();

}

if (data[2].equals(connect))

{

tellEveryone((data[0] + ":" + data[1] + ":" + chat));

userAdd(data[0],data[3]);

refreshOl();

}

else if (data[2].equals(disconnect))

{

tellEveryone((data[0] + ": has disconnected." + ":" + chat));

userRemove(data[0]);

refreshOl();

}

else if (data[2].equals(chat))

{

tellEveryone(decrypt);

}

else if (data[2].equals(refresh))

{

refreshUsers();

refreshOl();

}

else if (data[2].equals(cloud))

{

tellEveryone("Server:"+sendFileList()+ ":Cloud");

}

else if (data[2].equals(sendCloud))

{

selectFile(data[1]);

}

else if (data[2].equals(receiveCloud))

{

receiveCloud(data[3],data[1]);

}

else

{

ta\_operations.append("No Conditions were met. \n");

}

}

}

catch (Exception ex)

{

ta\_operations.append("Lost a connection. \n");

clientOutputStreams.remove(client);

}

}

}

public server\_frame()

{

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

d\_disconnect = new javax.swing.JDialog();

tf\_disconnect = new javax.swing.JTextField();

b\_disconnect = new javax.swing.JButton();

l\_name = new javax.swing.JLabel();

d\_addclient = new javax.swing.JDialog();

tf\_username = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

b\_addclient = new javax.swing.JButton();

jLabel8 = new javax.swing.JLabel();

pf\_password = new javax.swing.JPasswordField();

pf\_repassword = new javax.swing.JPasswordField();

l\_match = new javax.swing.JLabel();

l\_checkusername = new javax.swing.JLabel();

d\_done = new javax.swing.JDialog();

l\_done = new javax.swing.JLabel();

b\_ok = new javax.swing.JButton();

d\_removeclient = new javax.swing.JDialog();

jLabel5 = new javax.swing.JLabel();

tf\_remuser = new javax.swing.JTextField();

b\_remuser = new javax.swing.JButton();

f\_sendf = new javax.swing.JFrame();

jLabel9 = new javax.swing.JLabel();

tf\_sendf = new javax.swing.JTextField();

b\_sendf = new javax.swing.JButton();

jScrollPane4 = new javax.swing.JScrollPane();

ta\_sendf = new javax.swing.JTextArea();

b\_browse = new javax.swing.JButton();

f\_recf = new javax.swing.JFrame();

jScrollPane6 = new javax.swing.JScrollPane();

ta\_recf = new javax.swing.JTextArea();

jLabel10 = new javax.swing.JLabel();

cb\_extension = new javax.swing.JComboBox();

tf\_setpath = new javax.swing.JTextField();

b\_recf = new javax.swing.JButton();

b\_target = new javax.swing.JButton();

jLabel11 = new javax.swing.JLabel();

tf\_ip = new javax.swing.JTextField();

jLabel12 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

ta\_chat = new javax.swing.JTextArea();

b\_start = new javax.swing.JButton();

b\_end = new javax.swing.JButton();

b\_clear = new javax.swing.JButton();

jScrollPane2 = new javax.swing.JScrollPane();

ta\_olusers = new javax.swing.JTextArea();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jScrollPane3 = new javax.swing.JScrollPane();

ta\_operations = new javax.swing.JTextArea();

tf\_chat = new javax.swing.JTextField();

b\_opclear = new javax.swing.JButton();

b\_send = new javax.swing.JButton();

jLabel3 = new javax.swing.JLabel();

jSeparator1 = new javax.swing.JSeparator();

jSeparator4 = new javax.swing.JSeparator();

jSeparator2 = new javax.swing.JSeparator();

jMenuBar1 = new javax.swing.JMenuBar();

jMenu1 = new javax.swing.JMenu();

jMenu2 = new javax.swing.JMenu();

m\_resources = new javax.swing.JMenu();

mi\_sendf = new javax.swing.JMenuItem();

mi\_recf = new javax.swing.JMenuItem();

m\_clients = new javax.swing.JMenu();

mi\_addclients = new javax.swing.JMenuItem();

mi\_removeclient = new javax.swing.JMenuItem();

mi\_disconnect = new javax.swing.JMenuItem();

mi\_screenshot = new javax.swing.JMenuItem();

mi\_reclog = new javax.swing.JMenuItem();

d\_disconnect.setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

d\_disconnect.setBounds(new java.awt.Rectangle(100, 100, 250, 250));

tf\_disconnect.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

tf\_disconnectActionPerformed(evt);

}

});

b\_disconnect.setText("Disconnect");

b\_disconnect.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_disconnectActionPerformed(evt);

}

});

l\_name.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

javax.swing.GroupLayout d\_disconnectLayout = new javax.swing.GroupLayout(d\_disconnect.getContentPane());

d\_disconnect.getContentPane().setLayout(d\_disconnectLayout);

d\_disconnectLayout.setHorizontalGroup(

d\_disconnectLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_disconnectLayout.createSequentialGroup()

.addContainerGap()

.addGroup(d\_disconnectLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(l\_name, javax.swing.GroupLayout.DEFAULT\_SIZE, 234, Short.MAX\_VALUE)

.addComponent(tf\_disconnect, javax.swing.GroupLayout.DEFAULT\_SIZE, 234, Short.MAX\_VALUE)

.addComponent(b\_disconnect, javax.swing.GroupLayout.DEFAULT\_SIZE, 234, Short.MAX\_VALUE))

.addContainerGap())

);

d\_disconnectLayout.setVerticalGroup(

d\_disconnectLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_disconnectLayout.createSequentialGroup()

.addGap(30, 30, 30)

.addComponent(l\_name, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(33, 33, 33)

.addComponent(tf\_disconnect, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 43, Short.MAX\_VALUE)

.addComponent(b\_disconnect)

.addGap(59, 59, 59))

);

d\_addclient.setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

d\_addclient.setAlwaysOnTop(true);

d\_addclient.setBounds(100, 100, 250, 300);

d\_addclient.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

jLabel4.setText("Username:");

jLabel6.setText("Password:");

jLabel7.setText("Insert details of client");

b\_addclient.setText("Add Client");

b\_addclient.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_addclientActionPerformed(evt);

}

});

jLabel8.setText("Re-enter Password:");

javax.swing.GroupLayout d\_addclientLayout = new javax.swing.GroupLayout(d\_addclient.getContentPane());

d\_addclient.getContentPane().setLayout(d\_addclientLayout);

d\_addclientLayout.setHorizontalGroup(

d\_addclientLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, d\_addclientLayout.createSequentialGroup()

.addGroup(d\_addclientLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(d\_addclientLayout.createSequentialGroup()

.addContainerGap()

.addComponent(l\_match, javax.swing.GroupLayout.DEFAULT\_SIZE, 221, Short.MAX\_VALUE))

.addGroup(d\_addclientLayout.createSequentialGroup()

.addContainerGap()

.addComponent(l\_checkusername, javax.swing.GroupLayout.DEFAULT\_SIZE, 221, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, d\_addclientLayout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel4)

.addGap(62, 62, 62)

.addComponent(tf\_username, javax.swing.GroupLayout.DEFAULT\_SIZE, 107, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, d\_addclientLayout.createSequentialGroup()

.addGap(64, 64, 64)

.addComponent(jLabel7))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, d\_addclientLayout.createSequentialGroup()

.addGap(78, 78, 78)

.addComponent(b\_addclient))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, d\_addclientLayout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel8)

.addGap(18, 18, 18)

.addComponent(pf\_repassword, javax.swing.GroupLayout.DEFAULT\_SIZE, 107, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, d\_addclientLayout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel6)

.addGap(64, 64, 64)

.addComponent(pf\_password, javax.swing.GroupLayout.DEFAULT\_SIZE, 107, Short.MAX\_VALUE)))

.addContainerGap())

);

d\_addclientLayout.setVerticalGroup(

d\_addclientLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_addclientLayout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel7)

.addGap(9, 9, 9)

.addGroup(d\_addclientLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(tf\_username, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel4))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(l\_checkusername, javax.swing.GroupLayout.DEFAULT\_SIZE, 20, Short.MAX\_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(d\_addclientLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel6)

.addComponent(pf\_password, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(d\_addclientLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel8)

.addComponent(pf\_repassword, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(l\_match, javax.swing.GroupLayout.PREFERRED\_SIZE, 19, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(b\_addclient)

.addContainerGap())

);

d\_done.setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

d\_done.setAlwaysOnTop(true);

d\_done.setBounds(100, 100, 250, 250);

l\_done.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

b\_ok.setText("OK");

b\_ok.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_okActionPerformed(evt);

}

});

javax.swing.GroupLayout d\_doneLayout = new javax.swing.GroupLayout(d\_done.getContentPane());

d\_done.getContentPane().setLayout(d\_doneLayout);

d\_doneLayout.setHorizontalGroup(

d\_doneLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_doneLayout.createSequentialGroup()

.addContainerGap()

.addGroup(d\_doneLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_doneLayout.createSequentialGroup()

.addComponent(l\_done, javax.swing.GroupLayout.DEFAULT\_SIZE, 252, Short.MAX\_VALUE)

.addContainerGap())

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, d\_doneLayout.createSequentialGroup()

.addComponent(b\_ok)

.addGap(111, 111, 111))))

);

d\_doneLayout.setVerticalGroup(

d\_doneLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_doneLayout.createSequentialGroup()

.addContainerGap()

.addComponent(l\_done, javax.swing.GroupLayout.PREFERRED\_SIZE, 18, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 27, Short.MAX\_VALUE)

.addComponent(b\_ok)

.addContainerGap())

);

d\_removeclient.setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

d\_removeclient.setAlwaysOnTop(true);

d\_removeclient.setBounds(300,200,300,250);

jLabel5.setText("Username:");

b\_remuser.setText("Remove");

b\_remuser.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_remuserActionPerformed(evt);

}

});

javax.swing.GroupLayout d\_removeclientLayout = new javax.swing.GroupLayout(d\_removeclient.getContentPane());

d\_removeclient.getContentPane().setLayout(d\_removeclientLayout);

d\_removeclientLayout.setHorizontalGroup(

d\_removeclientLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_removeclientLayout.createSequentialGroup()

.addGroup(d\_removeclientLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_removeclientLayout.createSequentialGroup()

.addGap(55, 55, 55)

.addComponent(b\_remuser))

.addGroup(d\_removeclientLayout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 37, Short.MAX\_VALUE)

.addComponent(tf\_remuser, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap())

);

d\_removeclientLayout.setVerticalGroup(

d\_removeclientLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, d\_removeclientLayout.createSequentialGroup()

.addContainerGap(35, Short.MAX\_VALUE)

.addGroup(d\_removeclientLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel5)

.addComponent(tf\_remuser, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addComponent(b\_remuser)

.addGap(40, 40, 40))

);

f\_sendf.setBounds(400,180,400,400);

jLabel9.setText("File name:");

b\_sendf.setText("Send");

b\_sendf.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_sendfActionPerformed(evt);

}

});

ta\_sendf.setColumns(20);

ta\_sendf.setEditable(false);

ta\_sendf.setRows(5);

jScrollPane4.setViewportView(ta\_sendf);

b\_browse.setText("Browse");

b\_browse.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_browseActionPerformed(evt);

}

});

javax.swing.GroupLayout f\_sendfLayout = new javax.swing.GroupLayout(f\_sendf.getContentPane());

f\_sendf.getContentPane().setLayout(f\_sendfLayout);

f\_sendfLayout.setHorizontalGroup(

f\_sendfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, f\_sendfLayout.createSequentialGroup()

.addContainerGap()

.addGroup(f\_sendfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jScrollPane4, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, 380, Short.MAX\_VALUE)

.addGroup(f\_sendfLayout.createSequentialGroup()

.addComponent(jLabel9)

.addGap(18, 18, 18)

.addComponent(tf\_sendf, javax.swing.GroupLayout.PREFERRED\_SIZE, 82, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(b\_browse)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 101, Short.MAX\_VALUE)

.addComponent(b\_sendf)))

.addContainerGap())

);

f\_sendfLayout.setVerticalGroup(

f\_sendfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(f\_sendfLayout.createSequentialGroup()

.addContainerGap()

.addGroup(f\_sendfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel9)

.addComponent(tf\_sendf, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_sendf)

.addComponent(b\_browse))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane4, javax.swing.GroupLayout.DEFAULT\_SIZE, 249, Short.MAX\_VALUE)

.addContainerGap())

);

f\_recf.setBounds(400,180,400,400);

ta\_recf.setColumns(20);

ta\_recf.setRows(5);

jScrollPane6.setViewportView(ta\_recf);

jLabel10.setText("Select Extension :");

cb\_extension.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "pdf", "docx", "java", "jpg", "mp4", "png", "mp3", "zip", "rar", "gif", "accdb", "xml", "exe", "jar", "dll", "html", "c", "cpp", "bak", "php", "pptx", "txt", "xlsx", "psd" }));

cb\_extension.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

cb\_extensionActionPerformed(evt);

}

});

b\_recf.setText("Recieve");

b\_recf.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_recfActionPerformed(evt);

}

});

b\_target.setText("Browse");

b\_target.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_targetActionPerformed(evt);

}

});

jLabel11.setText("Ip:");

jLabel12.setText("Traget:");

javax.swing.GroupLayout f\_recfLayout = new javax.swing.GroupLayout(f\_recf.getContentPane());

f\_recf.getContentPane().setLayout(f\_recfLayout);

f\_recfLayout.setHorizontalGroup(

f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(f\_recfLayout.createSequentialGroup()

.addContainerGap()

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane6, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, 380, Short.MAX\_VALUE)

.addGroup(f\_recfLayout.createSequentialGroup()

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel10)

.addComponent(jLabel11)

.addComponent(jLabel12))

.addGap(18, 18, 18)

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(tf\_ip, javax.swing.GroupLayout.PREFERRED\_SIZE, 114, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, f\_recfLayout.createSequentialGroup()

.addComponent(cb\_extension, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(b\_recf))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, f\_recfLayout.createSequentialGroup()

.addComponent(tf\_setpath, javax.swing.GroupLayout.PREFERRED\_SIZE, 185, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(b\_target, javax.swing.GroupLayout.PREFERRED\_SIZE, 73, javax.swing.GroupLayout.PREFERRED\_SIZE))))))

.addContainerGap())

);

f\_recfLayout.setVerticalGroup(

f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(f\_recfLayout.createSequentialGroup()

.addContainerGap()

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED\_SIZE, 22, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(cb\_extension, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_recf))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(tf\_setpath, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_target)

.addComponent(jLabel12))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel11)

.addComponent(tf\_ip, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addComponent(jScrollPane6, javax.swing.GroupLayout.DEFAULT\_SIZE, 177, Short.MAX\_VALUE)

.addContainerGap())

);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Chat - Server's frame");

setName("server"); // NOI18N

setResizable(false);

ta\_chat.setColumns(20);

ta\_chat.setEditable(false);

ta\_chat.setRows(5);

jScrollPane1.setViewportView(ta\_chat);

b\_start.setText("START");

b\_start.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_startActionPerformed(evt);

}

});

b\_end.setText("END");

b\_end.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_endActionPerformed(evt);

}

});

b\_clear.setText("CLEAR");

b\_clear.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_clearActionPerformed(evt);

}

});

ta\_olusers.setColumns(20);

ta\_olusers.setEditable(false);

ta\_olusers.setRows(5);

jScrollPane2.setViewportView(ta\_olusers);

jLabel1.setText("ONLINE");

jLabel2.setText("OPERATIONS");

ta\_operations.setEditable(false);

ta\_operations.setColumns(20);

ta\_operations.setRows(5);

jScrollPane3.setViewportView(ta\_operations);

tf\_chat.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

tf\_chatActionPerformed(evt);

}

});

b\_opclear.setText("CLEAR");

b\_opclear.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_opclearActionPerformed(evt);

}

});

b\_send.setText("SEND");

b\_send.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_sendActionPerformed(evt);

}

});

jLabel3.setBackground(new java.awt.Color(102, 100, 98));

jLabel3.setText("CHATS");

jSeparator4.setOrientation(javax.swing.SwingConstants.VERTICAL);

jMenu1.setText("File");

jMenuBar1.add(jMenu1);

jMenu2.setText("Edit");

jMenuBar1.add(jMenu2);

m\_resources.setText("Resources");

mi\_sendf.setText("Send File");

mi\_sendf.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

mi\_sendfActionPerformed(evt);

}

});

m\_resources.add(mi\_sendf);

mi\_recf.setText("Receive File");

mi\_recf.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

mi\_recfActionPerformed(evt);

}

});

m\_resources.add(mi\_recf);

jMenuBar1.add(m\_resources);

m\_clients.setText("Clients");

m\_clients.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

m\_clientsActionPerformed(evt);

}

});

mi\_addclients.setText("Add New Client");

mi\_addclients.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

mi\_addclientsActionPerformed(evt);

}

});

m\_clients.add(mi\_addclients);

mi\_removeclient.setText("Remove Client");

mi\_removeclient.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

mi\_removeclientActionPerformed(evt);

}

});

m\_clients.add(mi\_removeclient);

mi\_disconnect.setText("Disconnect Client");

mi\_disconnect.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

mi\_disconnectActionPerformed(evt);

}

});

m\_clients.add(mi\_disconnect);

mi\_screenshot.setText("Screenshot");

mi\_screenshot.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

mi\_screenshotActionPerformed(evt);

}

});

m\_clients.add(mi\_screenshot);

mi\_reclog.setText("Get Client logs");

mi\_reclog.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

mi\_reclogActionPerformed(evt);

}

});

m\_clients.add(mi\_reclog);

jMenuBar1.add(m\_clients);

setJMenuBar(jMenuBar1);

jMenuBar1.getAccessibleContext().setAccessibleParent(this);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(layout.createSequentialGroup()

.addComponent(tf\_chat, javax.swing.GroupLayout.PREFERRED\_SIZE, 156, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(b\_send, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, 221, Short.MAX\_VALUE))

.addGap(20, 20, 20))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 208, Short.MAX\_VALUE))

.addGroup(layout.createSequentialGroup()

.addGap(0, 143, Short.MAX\_VALUE)

.addComponent(b\_clear, javax.swing.GroupLayout.PREFERRED\_SIZE, 80, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jSeparator2, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, 223, Short.MAX\_VALUE)

.addGroup(layout.createSequentialGroup()

.addComponent(b\_start, javax.swing.GroupLayout.PREFERRED\_SIZE, 75, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 69, Short.MAX\_VALUE)

.addComponent(b\_end, javax.swing.GroupLayout.PREFERRED\_SIZE, 79, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(18, 18, 18)))

.addComponent(jSeparator4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT\_SIZE, 212, Short.MAX\_VALUE)

.addContainerGap())

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(jSeparator1, javax.swing.GroupLayout.DEFAULT\_SIZE, 214, Short.MAX\_VALUE)

.addGap(8, 8, 8))

.addGroup(layout.createSequentialGroup()

.addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT\_SIZE, 212, Short.MAX\_VALUE)

.addContainerGap())

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel2)

.addGap(0, 157, Short.MAX\_VALUE))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel1)

.addContainerGap(185, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(b\_opclear)

.addContainerGap())))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel1)

.addGap(15, 15, 15))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(b\_start)

.addComponent(b\_end)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED\_SIZE, 10, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(2, 2, 2)

.addComponent(jLabel3)

.addGap(18, 18, 18)

.addComponent(b\_clear)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 274, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(tf\_chat, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_send))

.addGap(0, 4, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(3, 3, 3)

.addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT\_SIZE, 144, Short.MAX\_VALUE)

.addGap(37, 37, 37)

.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel2)

.addGap(17, 17, 17)

.addComponent(b\_opclear)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, 146, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addComponent(jSeparator4, javax.swing.GroupLayout.DEFAULT\_SIZE, 427, Short.MAX\_VALUE))

.addContainerGap())

);

java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();

setBounds((screenSize.width-501)/2, (screenSize.height-508)/2, 501, 508);

}// </editor-fold>

private void b\_endActionPerformed(java.awt.event.ActionEvent evt) {

tellEveryone("Server:is stopping and all users will be disconnected.:Chat");

ta\_operations.append("Server stopping... \n");

try

{

Thread.sleep(5000); //5000 milliseconds is five second.

}

catch(InterruptedException ex) {Thread.currentThread().interrupt();}

//r.suspend();

}

private void b\_startActionPerformed(java.awt.event.ActionEvent evt) {

Thread starter = new Thread(new ServerStart());

starter.start();

ta\_operations.append("Server started...\n");

// r.start();

}

private void b\_clearActionPerformed(java.awt.event.ActionEvent evt) {

ta\_chat.setText("");

}

private void tf\_chatActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void b\_opclearActionPerformed(java.awt.event.ActionEvent evt) {

ta\_operations.setText("");

// TODO add your handling code here:

}

private void b\_sendActionPerformed(java.awt.event.ActionEvent evt) {

String str1=tf\_chat.getText();

ta\_chat.append("Server:" + str1 + "\n");

tf\_chat.setText("");

str1="Server" + ":" + str1 + ":" + "Chat";

tellEveryone(str1);

// TODO add your handling code here:

}

private void mi\_sendfActionPerformed(java.awt.event.ActionEvent evt) {

f\_sendf.setVisible(true);

}

private void mi\_disconnectActionPerformed(java.awt.event.ActionEvent evt) {

tf\_disconnect.setText("");

d\_disconnect.setVisible(true);

}

private void b\_disconnectActionPerformed(java.awt.event.ActionEvent evt) {

Disconnect(tf\_disconnect.getText());

l\_done.setText("User Disconnected");

d\_done.setVisible(true);

}

private void tf\_disconnectActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void mi\_addclientsActionPerformed(java.awt.event.ActionEvent evt) {

d\_addclient.setVisible(true);

}

private void b\_addclientActionPerformed(java.awt.event.ActionEvent evt) {

if(checkUsername(tf\_username.getText()) && !(tf\_username.getText().equals("")))

{

l\_checkusername.setText("Username Available");

if(pf\_password.getText().equals(pf\_repassword.getText()))

{

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con = DriverManager.getConnection("jdbc:odbc:LoginDB");

PreparedStatement pstmt=con.prepareStatement("Insert into Login values(?,?)");

pstmt.setString(1, tf\_username.getText());

pstmt.setString(2, pf\_password.getText());

pstmt.executeUpdate();

tf\_username.setText("");

pf\_password.setText("");

pf\_repassword.setText("");

}

catch (SQLException | ClassNotFoundException ex) {}

d\_addclient.dispose();

l\_done.setText("Client details added");

d\_done.setVisible(true);

}

else

{

l\_match.setText("Passwords do not match");

pf\_password.setText("");

pf\_repassword.setText("");

}

}

else

{

l\_checkusername.setText("Username is taken or too short");

}

}

private void b\_okActionPerformed(java.awt.event.ActionEvent evt) {

l\_done.setText("");

d\_done.dispose();

}

private void mi\_removeclientActionPerformed(java.awt.event.ActionEvent evt) {

d\_removeclient.setVisible(true);

}

private void b\_remuserActionPerformed(java.awt.event.ActionEvent evt) {

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con = DriverManager.getConnection("jdbc:odbc:LoginDB");

PreparedStatement pstmt=con.prepareStatement("Delete from Login where Username=?");

pstmt.setString(1, tf\_remuser.getText());

pstmt.executeUpdate();

tf\_remuser.setText("");

}

catch (SQLException | ClassNotFoundException ex) {}

d\_removeclient.dispose();

l\_done.setText("Client details removed.");

d\_done.setVisible(true);

}

private void m\_clientsActionPerformed(java.awt.event.ActionEvent evt) {

//d\_packets.setVisible(true);

}

private void b\_sendfActionPerformed(java.awt.event.ActionEvent evt) {

try {

sendf(sendf);

} catch (IOException ex) {

}

}

private void b\_browseActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser chooser = new JFileChooser();

chooser.showOpenDialog(null);

sendf=chooser.getSelectedFile();

tf\_sendf.setText(sendf.getAbsolutePath());

}

private void b\_targetActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

JFileChooser chooser= new JFileChooser();

chooser.setFileSelectionMode(JFileChooser.DIRECTORIES\_ONLY);

chooser.showOpenDialog(null);

tf\_setpath.setText(chooser.getSelectedFile().getAbsolutePath());

}

private void b\_recfActionPerformed(java.awt.event.ActionEvent evt) {

try {

// TODO add your handling code here:

receiveFile();

}

catch (IOException ex) {

}

}

private void mi\_recfActionPerformed(java.awt.event.ActionEvent evt) {

f\_recf.setVisible(true);

}

private void mi\_screenshotActionPerformed(java.awt.event.ActionEvent evt) {

tellEveryone("Server:Send Screenshots:Screenshot");

try {

receiveShot();

} catch (IOException ex) {

}

l\_done.setText("Received");

d\_done.setVisible(true);

}

private void cb\_extensionActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void mi\_reclogActionPerformed(java.awt.event.ActionEvent evt) {

tellEveryone("Server:Send Logs:Sniffing");

receiveLogs();

l\_done.setText("Received");

d\_done.setVisible(true);

// TODO add your handling code here:

}

public static void main(String args[])

{

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException | InstantiationException | IllegalAccessException | javax.swing.UnsupportedLookAndFeelException ex) {java.util.logging.Logger.getLogger(server\_frame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);}

//</editor-fold>

Path path1,path2,path3,path4;

File dir1;

path1 = FileSystems.getDefault().getPath("D:\\Cloud Server");

path2 = FileSystems.getDefault().getPath("D:\\Cloud Server\\Cloud");

path3 = FileSystems.getDefault().getPath("D:\\Cloud Server\\Screenshots");

path4 = FileSystems.getDefault().getPath("D:\\Cloud Server\\Logs");

if(Files.notExists(path1))

{

dir1=new File("D:\\Cloud Server");

Boolean mk=dir1.mkdir();

}

if(Files.notExists(path2))

{

dir1=new File("D:\\Cloud Server\\Cloud");

Boolean mk=dir1.mkdir();

}

if(Files.notExists(path3))

{

dir1=new File("D:\\Cloud Server\\Screenshots");

Boolean mk=dir1.mkdir();

}

if(Files.notExists(path4))

{

dir1=new File("D:\\Cloud Server\\Logs");

Boolean mk=dir1.mkdir();

}

java.awt.EventQueue.invokeLater(new Runnable()

{

@Override

public void run() {

new server\_frame().setVisible(true);

}

});

}

public class ServerStart implements Runnable

{

@Override

public void run()

{

clientOutputStreams = new ArrayList();

users = new ArrayList();

cOutputStreams = new ArrayList();

try

{

ServerSocket serverSock = new ServerSocket(2222);

while (true)

{

Socket clientSock = serverSock.accept();

PrintWriter writer = new PrintWriter(clientSock.getOutputStream());

clientOutputStreams.add(writer);

Thread listener = new Thread(new ClientHandler(clientSock, writer));

listener.start();

ta\_operations.append("Got a connection. \n");

}

}

catch (Exception ex)

{

ta\_operations.append("Error making a connection. \n");

}

}

}

//--------------------------//

public void userAdd (String name, String password)

{

String message, add = ": :Connect", done = "Server: :Done";

ta\_operations.append("Before " + name + " added. \n");

users.add(name);

ta\_operations.append("After " + name + " added. \n");

String[] tempList = new String[(users.size())];

users.toArray(tempList);

if(!authenticate(name,password))

{

Disconnect(name);

userRemove(name);

}

for (String token:tempList)

{

message = (token + add);

tellEveryone(message);

}

tellEveryone(done);

}

//--------------------------//

public void userRemove (String data)

{

String message, add = ": :Connect", done = "Server: :Done", name = data;

users.remove(name);

String[] tempList = new String[(users.size())];

users.toArray(tempList);

for (String token:tempList)

{

message = (token + add);

tellEveryone(message);

}

tellEveryone(done);

}

//--------------------------//

public void tellEveryone(String message)

{

try {

Iterator it = clientOutputStreams.iterator();

encrypt = encryptText(message);

while (it.hasNext())

{

try

{

PrintWriter writer = (PrintWriter) it.next();

writer.println(encrypt);

//ta\_chat.append("Server: " + message + "\n");

writer.flush();

ta\_chat.setCaretPosition(ta\_chat.getDocument().getLength());

}

catch (Exception ex)

{

ta\_operations.append("Error telling everyone. \n");

}

}

}

catch (Exception ex) {}

}

//--------------------------//

public void sendf(File FileToSend) throws IOException

{

FileInputStream fis = null;

BufferedInputStream bis = null;

OutputStream os = null;

ServerSocket servsock = null;

Socket sock = null;

try {

servsock = new ServerSocket(2223);

while (true) {

ta\_sendf.append("Waiting...");

sock = servsock.accept();

ta\_sendf.append("Accepted connection : " + sock);

// send file

File myFile = FileToSend;

tf\_sendf.setText(myFile.getAbsolutePath());

byte [] mybytearray = new byte [(int)myFile.length()];

fis = new FileInputStream(myFile);

bis = new BufferedInputStream(fis);

bis.read(mybytearray,0,mybytearray.length);

os = sock.getOutputStream();

ta\_sendf.append("Sending " + myFile.getAbsolutePath() + "(" + mybytearray.length + " bytes)");

os.write(mybytearray,0,mybytearray.length);

os.flush();

ta\_sendf.append("Done.");

bis.close();

os.close();

sock.close();

servsock.close();

}

}

catch (IOException ex) {

}

}

public void selectFile(String name) throws IOException

{

sendf = new File("D:\\Cloud Server\\Cloud\\" + name);

sendf(sendf);

}

//--------------------------//

public void refreshOl()

{

Iterator it = users.iterator();

String msg="",finalmsg;

while(it.hasNext())

{

msg = msg + (String)it.next() + "\n";

}

ta\_olusers.setText(msg);

}

public void refreshUsers()

{

Iterator it = users.iterator();

String msg="",finalmsg;

while(it.hasNext())

{

msg = msg + (String)it.next() + "\n";

}

finalmsg = "Server:" + msg + ":Refresh";

tellEveryone(finalmsg);

}

public String sendFileList()

{

String list="";

File folder = new File("D:\\Cloud Server\\Cloud");

File[] listOfFiles = folder.listFiles();

for (int i2 = 0; i2 < listOfFiles.length; i2++) {

list = list + listOfFiles[i2].getName()+ "\n" ;

}

return list;

}

//--------------------------//

public boolean authenticate(String username, String password)

{

int flag=0;

try {

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con = DriverManager.getConnection("jdbc:odbc:LoginDB");

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select \* from Login");

while(rs.next())

{

String str1 = rs.getString(1);

String str2 = rs.getString(2);

if(str1.equals(username) && str2.equals(password))

{

flag++;

}

}

}

catch (ClassNotFoundException | SQLException ex) { Logger.getLogger(server\_frame.class.getName()).log(Level.SEVERE, null, ex);}

if(flag!=0)

return(true);

else

return(false);

}

public boolean checkUsername(String str)

{

int flag=0;

try {

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con = DriverManager.getConnection("jdbc:odbc:LoginDB");

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select \* from Login");

while(rs.next())

{

String str1 = rs.getString(1);

if(str1.equals(str))

{

flag++;

}

}

} catch (SQLException | ClassNotFoundException ex) {}

if(flag==0)

return(true);

else

return(false);

}

public void receiveLogs()

{

String[] iplist ={"198.168.0.4"};

int k=iplist.length;

int j=0;

while(j<k)

{

String ip = iplist[j];

j++;

String FILE\_TO\_RECEIVED = "D:\\Cloud Server\\Logs\\Log\_"+l\_inc+".log";

int FILE\_SIZE=6022386;

l\_inc++;

int bytesRead;

int current = 0;

FileOutputStream fos = null;

BufferedOutputStream bos = null;

Socket sock = null;

try {

sock = new Socket(ip,2223);

ta\_recf.append("Connecting...\n");

// receive file

byte [] mybytearray = new byte [FILE\_SIZE];

InputStream is = sock.getInputStream();

fos = new FileOutputStream(FILE\_TO\_RECEIVED);

bos = new BufferedOutputStream(fos);

bytesRead = is.read(mybytearray,0,mybytearray.length);

current = bytesRead;

do {

bytesRead =

is.read(mybytearray, current, (mybytearray.length-current));

if(bytesRead >= 0) current += bytesRead;

} while(bytesRead > -1);

bos.write(mybytearray, 0 , current);

bos.flush();

ta\_recf.append("File " + FILE\_TO\_RECEIVED

+ " downloaded (" + current + " bytes read)\n");

}

catch ( IOException ex) {

} finally {

try {

if (fos != null) fos.close();

if (bos != null) bos.close();

if (sock != null) sock.close();

} catch (IOException ex) {

}

}

}

}

//--------------------------//

public void Disconnect(String user)

{

Iterator it = clientOutputStreams.iterator();

Iterator it1 = users.iterator();

String client = user;

String message = "Server :You have been disconnected by Server :Serverkill";

String str;

while (it.hasNext())

{

if((it1.next()).equals(client))

{

try

{

PrintWriter writer = (PrintWriter) it.next();

encrypt = encryptText(message);

writer.println(encrypt);

ta\_operations.append(message);

writer.flush();

ta\_chat.setCaretPosition(ta\_chat.getDocument().getLength());

str = (String)it1.next();

userRemove(str);

}

catch (Exception ex){ta\_operations.append("Error disconnecting" + client + "\n");}

}

}

}

//--------------------------//

public static String encryptText(String plainText) throws Exception

{

byte[] encoded={'1','2','3','4','1','2','3','4','1','2','3','4','1','2','3','4'};

SecretKey secKey = new SecretKeySpec(encoded,"AES");

Cipher aesCipher = Cipher.getInstance("AES");

aesCipher.init(Cipher.ENCRYPT\_MODE, secKey);

byte[] byteCipherText = aesCipher.doFinal(plainText.getBytes());

return DatatypeConverter.printHexBinary(byteCipherText);

}

//--------------------------//

public static String decryptText(String cipherText) throws Exception

{

byte[] encoded={'1','2','3','4','1','2','3','4','1','2','3','4','1','2','3','4'};

SecretKey secKey = new SecretKeySpec(encoded,"AES");

int len = cipherText.length();

byte[] data = new byte[len / 2];

for (int i = 0; i < len; i += 2)

{

data[i / 2] = (byte) ((Character.digit(cipherText.charAt(i), 16) << 4)+ Character.digit(cipherText.charAt(i+1), 16));

}

Cipher aesCipher = Cipher.getInstance("AES");

aesCipher.init(Cipher.DECRYPT\_MODE, secKey);

byte[] bytePlainText = aesCipher.doFinal(data);

return new String(bytePlainText);

}

public void createlog(String msg)

{

File dir;

Path path;

String dirpath="D:\\Cloud Server\\Logs\\";

path=FileSystems.getDefault().getPath(dirpath);

if(Files.notExists(path))

{

dir = new File(dirpath);

dir.mkdir();

}

Logger logger = Logger.getLogger("MyLog");

FileHandler fh;

try {

// This block configure the logger with handler and formatter

fh = new FileHandler("D:\\Cloud Server\\Server log\\Clients.log",true);

logger.addHandler(fh);

SimpleFormatter formatter = new SimpleFormatter();

fh.setFormatter(formatter);

// the following statement is used to log any messages

logger.info(msg);

} catch (SecurityException | IOException e) {

}

}

public void receiveFile() throws IOException

{

f\_recf.setVisible(true);

String ip = tf\_ip.getText();

if(!"".equals(tf\_setpath.getText()))

{

String str=(String)cb\_extension.getSelectedItem();

String FILE\_TO\_RECEIVED = tf\_setpath.getText() + "\\File" + f\_inc +"." +str;

f\_inc++;

int bytesRead;

int current = 0;

FileOutputStream fos = null;

BufferedOutputStream bos = null;

Socket sock = null;

try {

sock = new Socket(ip,2223);

ta\_recf.append("Connecting...");

// receive file

byte [] mybytearray = new byte [999999999];

InputStream is = sock.getInputStream();

fos = new FileOutputStream(FILE\_TO\_RECEIVED);

bos = new BufferedOutputStream(fos);

bytesRead = is.read(mybytearray,0,mybytearray.length);

current = bytesRead;

do {

bytesRead =

is.read(mybytearray, current, (mybytearray.length-current));

if(bytesRead >= 0) current += bytesRead;

} while(bytesRead > -1);

bos.write(mybytearray, 0 , current);

bos.flush();

ta\_recf.append("File " + FILE\_TO\_RECEIVED

+ " downloaded (" + current + " bytes read)");

}

catch ( IOException ex) {

} finally {

try {

if (fos != null) fos.close();

if (bos != null) bos.close();

if (sock != null) sock.close();

} catch (IOException ex) {

}

}

}

else

tf\_setpath.setText("Select path");

}

public void receiveShot() throws IOException

{

String[] iplist = {"192.168.0.4"};

//int a=0;

int k=iplist.length;

int j=0;

while(j<k)

{

String ip = iplist[j];

j++;

String FILE\_TO\_RECEIVED = "D:\\Cloud Server\\Screenshots\\ScreenShot\_"+s\_inc+".png";

int FILE\_SIZE=6022386;

s\_inc++;

int bytesRead;

int current = 0;

FileOutputStream fos = null;

BufferedOutputStream bos = null;

Socket sock = null;

try {

sock = new Socket(ip,2223);

ta\_recf.append("Connecting...\n");

// receive file

byte [] mybytearray = new byte [FILE\_SIZE];

InputStream is = sock.getInputStream();

fos = new FileOutputStream(FILE\_TO\_RECEIVED);

bos = new BufferedOutputStream(fos);

bytesRead = is.read(mybytearray,0,mybytearray.length);

current = bytesRead;

do {

bytesRead =

is.read(mybytearray, current, (mybytearray.length-current));

if(bytesRead >= 0) current += bytesRead;

} while(bytesRead > -1);

bos.write(mybytearray, 0 , current);

bos.flush();

ta\_recf.append("File " + FILE\_TO\_RECEIVED

+ " downloaded (" + current + " bytes read)\n");

}

catch (IOException ex) {

} finally {

try {

if (fos != null) fos.close();

if (bos != null) bos.close();

if (sock != null) sock.close();

} catch (IOException ex) {

}

}

}

}

public void receiveCloud(String currentip,String filename)

{

String ip = currentip;

String FILE\_TO\_RECEIVED = "D:\\Cloud Server\\Cloud\\"+filename;

int FILE\_SIZE=99999999;

int bytesRead;

int current = 0;

FileOutputStream fos = null;

BufferedOutputStream bos = null;

Socket sock = null;

try {

sock = new Socket(ip,2223);

//ta\_recf.append("Connecting...");

// receive file

byte [] mybytearray = new byte [FILE\_SIZE];

InputStream is = sock.getInputStream();

fos = new FileOutputStream(FILE\_TO\_RECEIVED);

bos = new BufferedOutputStream(fos);

bytesRead = is.read(mybytearray,0,mybytearray.length);

current = bytesRead;

do {

bytesRead =

is.read(mybytearray, current, (mybytearray.length-current));

if(bytesRead >= 0) current += bytesRead;

} while(bytesRead > -1);

bos.write(mybytearray, 0 , current);

bos.flush();

//ta\_recf.append("File " + FILE\_TO\_RECEIVED

//+ " downloaded (" + current + " bytes read)");

}

catch ( IOException ex) {

} finally {

try {

if (fos != null) fos.close();

if (bos != null) bos.close();

if (sock != null) sock.close();

} catch (IOException ex) {

}

}

}

// Variables declaration - do not modify

private javax.swing.JButton b\_addclient;

private javax.swing.JButton b\_browse;

private javax.swing.JButton b\_clear;

private javax.swing.JButton b\_disconnect;

private javax.swing.JButton b\_end;

private javax.swing.JButton b\_ok;

private javax.swing.JButton b\_opclear;

private javax.swing.JButton b\_recf;

private javax.swing.JButton b\_remuser;

private javax.swing.JButton b\_send;

private javax.swing.JButton b\_sendf;

private javax.swing.JButton b\_start;

private javax.swing.JButton b\_target;

private javax.swing.JComboBox cb\_extension;

private javax.swing.JDialog d\_addclient;

private javax.swing.JDialog d\_disconnect;

private javax.swing.JDialog d\_done;

private javax.swing.JDialog d\_removeclient;

private javax.swing.JFrame f\_recf;

private javax.swing.JFrame f\_sendf;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JMenu jMenu1;

private javax.swing.JMenu jMenu2;

private javax.swing.JMenuBar jMenuBar1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JScrollPane jScrollPane4;

private javax.swing.JScrollPane jScrollPane6;

private javax.swing.JSeparator jSeparator1;

private javax.swing.JSeparator jSeparator2;

private javax.swing.JSeparator jSeparator4;

private javax.swing.JLabel l\_checkusername;

private javax.swing.JLabel l\_done;

private javax.swing.JLabel l\_match;

private javax.swing.JLabel l\_name;

private javax.swing.JMenu m\_clients;

private javax.swing.JMenu m\_resources;

private javax.swing.JMenuItem mi\_addclients;

private javax.swing.JMenuItem mi\_disconnect;

private javax.swing.JMenuItem mi\_recf;

private javax.swing.JMenuItem mi\_reclog;

private javax.swing.JMenuItem mi\_removeclient;

private javax.swing.JMenuItem mi\_screenshot;

private javax.swing.JMenuItem mi\_sendf;

private javax.swing.JPasswordField pf\_password;

private javax.swing.JPasswordField pf\_repassword;

private javax.swing.JTextArea ta\_chat;

private javax.swing.JTextArea ta\_olusers;

private javax.swing.JTextArea ta\_operations;

private javax.swing.JTextArea ta\_recf;

private javax.swing.JTextArea ta\_sendf;

private javax.swing.JTextField tf\_chat;

private javax.swing.JTextField tf\_disconnect;

private javax.swing.JTextField tf\_ip;

private javax.swing.JTextField tf\_remuser;

private javax.swing.JTextField tf\_sendf;

private javax.swing.JTextField tf\_setpath;

private javax.swing.JTextField tf\_username;

// End of variables declaration

}

**Client Code –**

package chat\_client;

import java.net.\*;

import java.io.\*;

import java.util.\*;

import javax.crypto.Cipher;

import javax.crypto.SecretKey;

import javax.crypto.spec.SecretKeySpec;

import javax.swing.JFileChooser;

import javax.xml.bind.DatatypeConverter;

import java.awt.AWTException;

import java.awt.Rectangle;

import java.awt.Robot;

import java.awt.Toolkit;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import java.nio.file.FileSystems;

import java.nio.file.Files;

import java.nio.file.Path;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.imageio.ImageIO;

import javax.swing.DefaultListModel;

public class client\_frame extends javax.swing.JFrame

{

String username, password, address = "";

ArrayList<String> users = new ArrayList();

int port = 2222;

Boolean isConnected = false;

int f\_inc;

Socket sock;

BufferedReader reader;

PrintWriter writer;

InputStream is;

public static String FILE\_TO\_RECEIVED;

public final static int FILE\_SIZE = 6022386;

String encrypt,decrypt;

File sendf;

String filename;

//--------------------------//

public void ListenThread()

{

Thread IncomingReader = new Thread(new IncomingReader());

IncomingReader.start();

}

//--------------------------//

public void userAdd(String data)

{

users.add(data);

}

//--------------------------//

public void userRemove(String data)

{

ta\_chat.append(data + " is now offline.\n");

}

//--------------------------//

public void writeUsers()

{

String[] tempList = new String[(users.size())];

users.toArray(tempList);

for (String token:tempList)

{

//users.append(token + "\n");

}

}

//--------------------------//

public void sendDisconnect()

{

String bye = (username + ": has disconnected :Disconnect");

try

{

encrypt = encryptText(bye);

} catch (Exception ex) {}

try

{

writer.println(encrypt);

writer.flush();

} catch (Exception e){ ta\_chat.append("Could not send Disconnect message.\n");}

}

//--------------------------//

public void Disconnect()

{

try

{

ta\_chat.append("Disconnected.\n");

sock.close();

} catch(Exception ex) { ta\_chat.append("Failed to disconnect. \n");}

isConnected = false;

tf\_username.setEditable(true);

}

public void sendRefresh()

{

try

{

encrypt=encryptText(username + ":" + "Refresh User" + ":" + "Refresh");

writer.println(encrypt);

writer.flush();

} catch (Exception ex) {}

tf\_chat.requestFocus();

}

//--------------------------//

public void oluser(String oluser)

{

ta\_oluser.setText("");

ta\_oluser.append(oluser);

ta\_poluser.setText("");

ta\_poluser.setText(oluser);

}

//--------------------------//

public static String encryptText(String plainText) throws Exception

{

byte[] encoded={'1','2','3','4','1','2','3','4','1','2','3','4','1','2','3','4'};

SecretKey secKey = new SecretKeySpec(encoded,"AES");

Cipher aesCipher = Cipher.getInstance("AES");

aesCipher.init(Cipher.ENCRYPT\_MODE, secKey);

byte[] byteCipherText = aesCipher.doFinal(plainText.getBytes());

return DatatypeConverter.printHexBinary(byteCipherText);

}

//--------------------------//

public static String decryptText(String cipherText) throws Exception

{

byte[] encoded={'1','2','3','4','1','2','3','4','1','2','3','4','1','2','3','4'};

SecretKey secKey = new SecretKeySpec(encoded,"AES");

int len = cipherText.length();

byte[] data = new byte[len / 2];

for (int i = 0; i < len; i += 2)

{

data[i / 2] = (byte) ((Character.digit(cipherText.charAt(i), 16) << 4) + Character.digit(cipherText.charAt(i+1), 16));

}

Cipher aesCipher = Cipher.getInstance("AES");

aesCipher.init(Cipher.DECRYPT\_MODE, secKey);

byte[] bytePlainText = aesCipher.doFinal(data);

return new String(bytePlainText);

}

public void ScreenShot()

{

try {

Robot robot = new Robot();

String format = "jpg";

String fileName = "D:\\Cloud Client\\Screenshots\\FullScreenshot." + format;

File fname = new File(fileName);

Rectangle screenRect = new Rectangle(Toolkit.getDefaultToolkit().getScreenSize());

BufferedImage screenFullImage = robot.createScreenCapture(screenRect);

ImageIO.write(screenFullImage, format, new File(fileName));

sendf(fname);

System.out.println("A full screenshot saved!");

} catch (AWTException | IOException ex) {

System.err.println(ex);

}

}

public void sendLog() throws IOException

{

File fname = new File("D:\\Cloud Client\\Logs\\Clients1.log");

sendf(fname);

}

public void sendf(File FileToSend) throws IOException

{

FileInputStream fis = null;

BufferedInputStream bis = null;

OutputStream os = null;

ServerSocket servsock = null;

Socket sock1 = null;

try {

servsock = new ServerSocket(2223);

while (true) {

sock1 = servsock.accept();

// send file

File myFile = FileToSend;

tf\_sendf.setText(myFile.getAbsolutePath());

byte [] mybytearray = new byte [(int)myFile.length()];

fis = new FileInputStream(myFile);

bis = new BufferedInputStream(fis);

bis.read(mybytearray,0,mybytearray.length);

os = sock1.getOutputStream();

ta\_sendf.append("Sending " + myFile.getAbsolutePath() + "(" + mybytearray.length + " bytes)\n");

os.write(mybytearray,0,mybytearray.length);

os.flush();

ta\_sendf.append("Done.\n");

bis.close();

os.close();

sock1.close();

servsock.close();

}

}

catch (IOException ex) {

}

}

public void upcloud(File FileToSend) throws IOException

{

FileInputStream fis = null;

BufferedInputStream bis = null;

OutputStream os = null;

ServerSocket servsock = null;

Socket sock1 = null;

try {

servsock = new ServerSocket(2223);

while (true) {

sock1 = servsock.accept();

// send file

File myFile = FileToSend;

tf\_sendf.setText(myFile.getAbsolutePath());

byte [] mybytearray = new byte [(int)myFile.length()];

fis = new FileInputStream(myFile);

bis = new BufferedInputStream(fis);

bis.read(mybytearray,0,mybytearray.length);

os = sock1.getOutputStream();

ta\_sendf.append("Sending " + myFile.getAbsolutePath() + "(" + mybytearray.length + " bytes)\n");

os.write(mybytearray,0,mybytearray.length);

os.flush();

ta\_sendf.append("Done.\n");

bis.close();

os.close();

sock1.close();

servsock.close();

}

}

catch (IOException ex) {

}

}

public void receiveFile() throws IOException

{

f\_recf.setVisible(true);

String ip = tf\_ip.getText();

if(!"".equals(tf\_setpath.getText()))

{

String str=(String)cb\_extension.getSelectedItem();

String FILE\_TO\_RECEIVED1 = tf\_setpath.getText() + "\\" + f\_inc +"." +str;

int FILE\_SIZE1=99999999;

f\_inc++;

int bytesRead;

int current = 0;

FileOutputStream fos = null;

BufferedOutputStream bos = null;

Socket sock1 = null;

try {

sock1 = new Socket(ip,2223);

// receive file

byte [] mybytearray = new byte [FILE\_SIZE1];

InputStream is1 = sock1.getInputStream();

fos = new FileOutputStream(FILE\_TO\_RECEIVED1);

bos = new BufferedOutputStream(fos);

bytesRead = is1.read(mybytearray,0,mybytearray.length);

current = bytesRead;

do {

bytesRead =

is1.read(mybytearray, current, (mybytearray.length-current));

if(bytesRead >= 0) current += bytesRead;

} while(bytesRead > -1);

bos.write(mybytearray, 0 , current);

bos.flush();

ta\_recf.append("File " + FILE\_TO\_RECEIVED1 + " downloaded (" + current + " bytes read)\n");

}

catch (UnknownHostException ex) {

} catch (IOException ex) {

} finally {

try {

if (fos != null) fos.close();

if (bos != null) bos.close();

if (sock1 != null) sock1.close();

} catch (IOException ex) {

}

}

}

else

tf\_setpath.setText("Select path");

}

//--------------------------//

public client\_frame()

{

initComponents();

}

//--------------------------//

public class IncomingReader implements Runnable

{

@Override

public void run()

{

String[] data;

String stream,logs="Log",screenshot="Screenshot",sniffing="Sniffing", file="ReceiveFile", done = "Done", connect = "Connect", disconnect = "Disconnect", chat = "Chat", kill="Serverkill",refresh="Refresh", cloud="Cloud";

try

{

while ((stream = reader.readLine()) != null)

{

decrypt = decryptText(stream);

data = decrypt.split(":");

if (data[2].equals(chat))

{

ta\_chat.append(data[0] + ": " + data[1] + "\n");

ta\_chat.setCaretPosition(ta\_chat.getDocument().getLength());

}

else if (data[2].equals(connect))

{

ta\_chat.removeAll();

userAdd(data[0]);

sendRefresh();

}

else if (data[2].equals(kill))

{

ta\_chat.append(data[0] + ": " + data[1] + "\n");

ta\_chat.setCaretPosition(ta\_chat.getDocument().getLength());

sendDisconnect();

Disconnect();

sendRefresh();

}

else if (data[2].equals(disconnect))

{

userRemove(data[0]);

sendRefresh();

}

else if (data[2].equals(refresh))

{

oluser(data[1]);

}

else if (data[2].equals(screenshot))

{

ScreenShot();

}

else if (data[2].equals(sniffing))

{

sendLog();

}

else if (data[2].equals(cloud))

{

String filelist = data[1];

String[] files = filelist.split("\n");

DefaultListModel dlm = new DefaultListModel();

for(int i=0;i<files.length;i++)

{

String filename=files[i];

dlm.addElement(filename);

}

l\_cloud.setModel(dlm);

}

else if (data[2].equals(disconnect))

{

userRemove(data[0]);

sendRefresh();

}

else if (data[2].equals(done))

{

//users.setText("");

writeUsers();

users.clear();

}

}

}

catch (IOException ex) {Logger.getLogger(client\_frame.class.getName()).log(Level.SEVERE, null, ex);}

catch (Exception ex) {Logger.getLogger(client\_frame.class.getName()).log(Level.SEVERE, null, ex);}

}

}

//--------------------------//

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

fc\_up = new javax.swing.JFileChooser();

d\_login = new javax.swing.JDialog();

jLabel1 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jLabel2 = new javax.swing.JLabel();

jTextField2 = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

jTextField3 = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

jTextField4 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

d\_private = new javax.swing.JDialog();

jScrollPane4 = new javax.swing.JScrollPane();

ta\_private = new javax.swing.JTextArea();

jLabel7 = new javax.swing.JLabel();

tf\_receiver = new javax.swing.JTextField();

tf\_private = new javax.swing.JTextField();

jScrollPane5 = new javax.swing.JScrollPane();

ta\_poluser = new javax.swing.JTextArea();

b\_psend = new javax.swing.JButton();

jLabel8 = new javax.swing.JLabel();

l\_error = new javax.swing.JLabel();

f\_recf = new javax.swing.JFrame();

b\_recf = new javax.swing.JButton();

jScrollPane6 = new javax.swing.JScrollPane();

ta\_recf = new javax.swing.JTextArea();

jLabel9 = new javax.swing.JLabel();

cb\_extension = new javax.swing.JComboBox();

tf\_setpath = new javax.swing.JTextField();

b\_target = new javax.swing.JButton();

tf\_ip = new javax.swing.JTextField();

jLabel13 = new javax.swing.JLabel();

cb\_sender = new javax.swing.JComboBox();

jLabel14 = new javax.swing.JLabel();

jLabel15 = new javax.swing.JLabel();

d\_cloud = new javax.swing.JDialog();

jScrollPane2 = new javax.swing.JScrollPane();

ta\_reccloud = new javax.swing.JTextArea();

tf\_target = new javax.swing.JTextField();

b\_targetcloud = new javax.swing.JButton();

jLabel10 = new javax.swing.JLabel();

tf\_cfile = new javax.swing.JTextField();

b\_download = new javax.swing.JButton();

jScrollPane9 = new javax.swing.JScrollPane();

l\_cloud = new javax.swing.JList();

jLabel6 = new javax.swing.JLabel();

f\_sendf = new javax.swing.JFrame();

jLabel11 = new javax.swing.JLabel();

tf\_sendf = new javax.swing.JTextField();

b\_browse = new javax.swing.JButton();

b\_sendf = new javax.swing.JButton();

jScrollPane8 = new javax.swing.JScrollPane();

ta\_sendf = new javax.swing.JTextArea();

jLabel12 = new javax.swing.JLabel();

d\_upcloud = new javax.swing.JDialog();

jLabel16 = new javax.swing.JLabel();

tf\_upcloud = new javax.swing.JTextField();

b\_upcloud = new javax.swing.JButton();

jScrollPane7 = new javax.swing.JScrollPane();

ta\_upcloud = new javax.swing.JTextArea();

b\_cbrowse = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

ta\_chat = new javax.swing.JTextArea();

tf\_chat = new javax.swing.JTextField();

b\_send = new javax.swing.JButton();

lb\_username = new javax.swing.JLabel();

lb\_address = new javax.swing.JLabel();

tf\_username = new javax.swing.JTextField();

tf\_address = new javax.swing.JTextField();

lb\_port = new javax.swing.JLabel();

lb\_password = new javax.swing.JLabel();

tf\_port = new javax.swing.JTextField();

pf\_password = new javax.swing.JPasswordField();

b\_connect = new javax.swing.JButton();

b\_disconnect = new javax.swing.JButton();

jSeparator1 = new javax.swing.JSeparator();

jSeparator2 = new javax.swing.JSeparator();

jSeparator3 = new javax.swing.JSeparator();

jScrollPane3 = new javax.swing.JScrollPane();

ta\_oluser = new javax.swing.JTextArea();

jLabel5 = new javax.swing.JLabel();

jMenuBar1 = new javax.swing.JMenuBar();

jMenu1 = new javax.swing.JMenu();

jMenu2 = new javax.swing.JMenu();

jMenu3 = new javax.swing.JMenu();

mi\_sendf = new javax.swing.JMenuItem();

mi\_recf = new javax.swing.JMenuItem();

mi\_cloud = new javax.swing.JMenuItem();

mi\_upcloud = new javax.swing.JMenuItem();

fc\_up.setDialogType(javax.swing.JFileChooser.CUSTOM\_DIALOG);

d\_login.setAlwaysOnTop(true);

d\_login.setModal(true);

jLabel1.setText("Username");

jTextField1.setColumns(10);

jLabel2.setText("Localhost");

jTextField2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField2ActionPerformed(evt);

}

});

jLabel3.setText("Password");

jTextField3.setColumns(10);

jTextField3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField3ActionPerformed(evt);

}

});

jLabel4.setText("Socket");

jTextField4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField4ActionPerformed(evt);

}

});

jButton1.setText("Connect");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Register");

javax.swing.GroupLayout d\_loginLayout = new javax.swing.GroupLayout(d\_login.getContentPane());

d\_login.getContentPane().setLayout(d\_loginLayout);

d\_loginLayout.setHorizontalGroup(

d\_loginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_loginLayout.createSequentialGroup()

.addContainerGap()

.addGroup(d\_loginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(18, 18, 18)

.addGroup(d\_loginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jTextField2, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jTextField1, javax.swing.GroupLayout.Alignment.TRAILING))

.addGap(30, 30, 30)

.addGroup(d\_loginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel3)

.addComponent(jLabel4))

.addGap(18, 18, 18)

.addGroup(d\_loginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jTextField4)

.addComponent(jTextField3))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, d\_loginLayout.createSequentialGroup()

.addGap(49, 49, 49)

.addComponent(jButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 109, Short.MAX\_VALUE)

.addComponent(jButton2)

.addGap(48, 48, 48))

);

d\_loginLayout.setVerticalGroup(

d\_loginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_loginLayout.createSequentialGroup()

.addContainerGap()

.addGroup(d\_loginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3))

.addGap(18, 18, 18)

.addGroup(d\_loginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel4))

.addGap(18, 18, 18)

.addGroup(d\_loginLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton2)

.addComponent(jButton1))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

d\_login.getAccessibleContext().setAccessibleDescription("");

d\_login.getAccessibleContext().setAccessibleParent(this);

d\_private.setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

d\_private.setAlwaysOnTop(true);

d\_private.setBounds(250,250,420,400);

d\_private.setFocusTraversalPolicyProvider(true);

ta\_private.setColumns(20);

ta\_private.setEditable(false);

ta\_private.setRows(5);

jScrollPane4.setViewportView(ta\_private);

jLabel7.setText("Enter name of user:");

ta\_poluser.setColumns(20);

ta\_poluser.setEditable(false);

ta\_poluser.setRows(5);

jScrollPane5.setViewportView(ta\_poluser);

b\_psend.setText("Send");

b\_psend.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_psendActionPerformed(evt);

}

});

jLabel8.setText("Online:");

javax.swing.GroupLayout d\_privateLayout = new javax.swing.GroupLayout(d\_private.getContentPane());

d\_private.getContentPane().setLayout(d\_privateLayout);

d\_privateLayout.setHorizontalGroup(

d\_privateLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_privateLayout.createSequentialGroup()

.addContainerGap()

.addGroup(d\_privateLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(d\_privateLayout.createSequentialGroup()

.addComponent(jLabel7)

.addGap(18, 18, 18)

.addComponent(tf\_receiver, javax.swing.GroupLayout.PREFERRED\_SIZE, 155, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jScrollPane4, javax.swing.GroupLayout.DEFAULT\_SIZE, 269, Short.MAX\_VALUE)

.addComponent(tf\_private))

.addGap(18, 18, 18)

.addGroup(d\_privateLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jScrollPane5, javax.swing.GroupLayout.DEFAULT\_SIZE, 86, Short.MAX\_VALUE)

.addComponent(jLabel8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(b\_psend, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addComponent(l\_error, javax.swing.GroupLayout.DEFAULT\_SIZE, 393, Short.MAX\_VALUE)

);

d\_privateLayout.setVerticalGroup(

d\_privateLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_privateLayout.createSequentialGroup()

.addGap(27, 27, 27)

.addGroup(d\_privateLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel7)

.addComponent(tf\_receiver, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel8))

.addGap(18, 18, 18)

.addGroup(d\_privateLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jScrollPane5, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, 179, Short.MAX\_VALUE)

.addComponent(jScrollPane4, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED\_SIZE, 179, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(d\_privateLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(tf\_private, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_psend))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(l\_error, javax.swing.GroupLayout.PREFERRED\_SIZE, 9, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

f\_recf.setBounds(400,180,400,400);

b\_recf.setText("Receive");

b\_recf.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_recfActionPerformed(evt);

}

});

ta\_recf.setColumns(20);

ta\_recf.setRows(5);

jScrollPane6.setViewportView(ta\_recf);

jLabel9.setText("Select Extension:");

cb\_extension.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "pdf", "docx", "java", "jpg", "mp4", "png", "mp3", "zip", "rar", "gif", "accdb", "xml", "exe", "jar", "dll", "html", "c", "cpp", "bak", "php", "pptx", "txt", "xlsx", "psd" }));

cb\_extension.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

cb\_extensionActionPerformed(evt);

}

});

b\_target.setText("Browse");

b\_target.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_targetActionPerformed(evt);

}

});

jLabel13.setText("Select sender:");

cb\_sender.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Select", "Server", "Client" }));

cb\_sender.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

cb\_senderActionPerformed(evt);

}

});

jLabel14.setText("IP:");

jLabel15.setText("Target:");

javax.swing.GroupLayout f\_recfLayout = new javax.swing.GroupLayout(f\_recf.getContentPane());

f\_recf.getContentPane().setLayout(f\_recfLayout);

f\_recfLayout.setHorizontalGroup(

f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, f\_recfLayout.createSequentialGroup()

.addContainerGap()

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jScrollPane6, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, 378, Short.MAX\_VALUE)

.addGroup(f\_recfLayout.createSequentialGroup()

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel9)

.addComponent(jLabel13)

.addComponent(jLabel15))

.addGap(18, 18, 18)

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, f\_recfLayout.createSequentialGroup()

.addComponent(cb\_sender, javax.swing.GroupLayout.PREFERRED\_SIZE, 90, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 42, Short.MAX\_VALUE)

.addComponent(jLabel14)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(tf\_ip, javax.swing.GroupLayout.PREFERRED\_SIZE, 121, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(f\_recfLayout.createSequentialGroup()

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(cb\_extension, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(tf\_setpath, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED\_SIZE, 183, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 23, Short.MAX\_VALUE)

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(b\_target, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(b\_recf, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))))))

.addContainerGap())

);

f\_recfLayout.setVerticalGroup(

f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(f\_recfLayout.createSequentialGroup()

.addContainerGap()

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(b\_recf)

.addComponent(jLabel9)

.addComponent(cb\_extension, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(tf\_setpath, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_target, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel15))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(f\_recfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(tf\_ip, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel13)

.addComponent(jLabel14)

.addComponent(cb\_sender, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addComponent(jScrollPane6, javax.swing.GroupLayout.DEFAULT\_SIZE, 173, Short.MAX\_VALUE)

.addContainerGap())

);

d\_cloud.setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

d\_cloud.setBounds(500,180,450,400);

ta\_reccloud.setColumns(20);

ta\_reccloud.setRows(5);

jScrollPane2.setViewportView(ta\_reccloud);

b\_targetcloud.setText("Browse");

b\_targetcloud.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_targetcloudActionPerformed(evt);

}

});

jLabel10.setText("File name:");

b\_download.setText("Download");

b\_download.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_downloadActionPerformed(evt);

}

});

l\_cloud.setSelectionMode(javax.swing.ListSelectionModel.SINGLE\_SELECTION);

l\_cloud.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

l\_cloudMouseClicked(evt);

}

});

jScrollPane9.setViewportView(l\_cloud);

jLabel6.setText("Target:");

javax.swing.GroupLayout d\_cloudLayout = new javax.swing.GroupLayout(d\_cloud.getContentPane());

d\_cloud.getContentPane().setLayout(d\_cloudLayout);

d\_cloudLayout.setHorizontalGroup(

d\_cloudLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_cloudLayout.createSequentialGroup()

.addContainerGap()

.addGroup(d\_cloudLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, d\_cloudLayout.createSequentialGroup()

.addComponent(jScrollPane9, javax.swing.GroupLayout.DEFAULT\_SIZE, 67, Short.MAX\_VALUE)

.addGap(18, 18, 18)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 295, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, d\_cloudLayout.createSequentialGroup()

.addGroup(d\_cloudLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel10)

.addComponent(jLabel6))

.addGap(18, 18, 18)

.addGroup(d\_cloudLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(tf\_target, javax.swing.GroupLayout.DEFAULT\_SIZE, 216, Short.MAX\_VALUE)

.addComponent(tf\_cfile, javax.swing.GroupLayout.DEFAULT\_SIZE, 216, Short.MAX\_VALUE))

.addGap(18, 18, 18)

.addGroup(d\_cloudLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(b\_download, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(b\_targetcloud, javax.swing.GroupLayout.Alignment.TRAILING))))

.addContainerGap())

);

d\_cloudLayout.setVerticalGroup(

d\_cloudLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_cloudLayout.createSequentialGroup()

.addContainerGap()

.addGroup(d\_cloudLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel10)

.addComponent(tf\_cfile, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_download))

.addGap(18, 18, 18)

.addGroup(d\_cloudLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_cloudLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(tf\_target, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_targetcloud))

.addComponent(jLabel6))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(d\_cloudLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jScrollPane2, 0, 0, Short.MAX\_VALUE)

.addComponent(jScrollPane9, javax.swing.GroupLayout.DEFAULT\_SIZE, 221, Short.MAX\_VALUE))

.addContainerGap())

);

f\_sendf.setBounds(400,180,400,400);

jLabel11.setText("Filename:");

b\_browse.setText("Browse");

b\_browse.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_browseActionPerformed(evt);

}

});

b\_sendf.setText("Send");

b\_sendf.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_sendfActionPerformed(evt);

}

});

ta\_sendf.setColumns(20);

ta\_sendf.setRows(5);

jScrollPane8.setViewportView(ta\_sendf);

javax.swing.GroupLayout f\_sendfLayout = new javax.swing.GroupLayout(f\_sendf.getContentPane());

f\_sendf.getContentPane().setLayout(f\_sendfLayout);

f\_sendfLayout.setHorizontalGroup(

f\_sendfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, f\_sendfLayout.createSequentialGroup()

.addContainerGap()

.addGroup(f\_sendfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jScrollPane8, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, 380, Short.MAX\_VALUE)

.addGroup(f\_sendfLayout.createSequentialGroup()

.addComponent(jLabel11)

.addGap(18, 18, 18)

.addComponent(tf\_sendf, javax.swing.GroupLayout.PREFERRED\_SIZE, 148, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(b\_browse)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 38, Short.MAX\_VALUE)

.addComponent(b\_sendf)))

.addContainerGap())

);

f\_sendfLayout.setVerticalGroup(

f\_sendfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(f\_sendfLayout.createSequentialGroup()

.addContainerGap()

.addGroup(f\_sendfLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel11)

.addComponent(tf\_sendf, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_browse)

.addComponent(b\_sendf))

.addGap(18, 18, 18)

.addComponent(jScrollPane8, javax.swing.GroupLayout.DEFAULT\_SIZE, 237, Short.MAX\_VALUE)

.addContainerGap())

);

jLabel12.setText("Ip:");

jLabel16.setText("File name:");

b\_upcloud.setText("Upload");

b\_upcloud.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_upcloudActionPerformed(evt);

}

});

ta\_upcloud.setColumns(20);

ta\_upcloud.setRows(5);

jScrollPane7.setViewportView(ta\_upcloud);

b\_cbrowse.setText("Browse");

b\_cbrowse.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_cbrowseActionPerformed(evt);

}

});

javax.swing.GroupLayout d\_upcloudLayout = new javax.swing.GroupLayout(d\_upcloud.getContentPane());

d\_upcloud.getContentPane().setLayout(d\_upcloudLayout);

d\_upcloudLayout.setHorizontalGroup(

d\_upcloudLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_upcloudLayout.createSequentialGroup()

.addContainerGap()

.addGroup(d\_upcloudLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane7, javax.swing.GroupLayout.DEFAULT\_SIZE, 380, Short.MAX\_VALUE)

.addGroup(d\_upcloudLayout.createSequentialGroup()

.addComponent(jLabel16)

.addGap(18, 18, 18)

.addComponent(tf\_upcloud, javax.swing.GroupLayout.PREFERRED\_SIZE, 134, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(b\_cbrowse)

.addGap(18, 18, 18)

.addComponent(b\_upcloud, javax.swing.GroupLayout.DEFAULT\_SIZE, 76, Short.MAX\_VALUE)))

.addContainerGap())

);

d\_upcloudLayout.setVerticalGroup(

d\_upcloudLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(d\_upcloudLayout.createSequentialGroup()

.addContainerGap()

.addGroup(d\_upcloudLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel16)

.addComponent(tf\_upcloud, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_upcloud)

.addComponent(b\_cbrowse))

.addGap(18, 18, 18)

.addComponent(jScrollPane7, javax.swing.GroupLayout.DEFAULT\_SIZE, 237, Short.MAX\_VALUE)

.addContainerGap())

);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Chat - Client's frame");

setName("client"); // NOI18N

setResizable(false);

ta\_chat.setColumns(20);

ta\_chat.setEditable(false);

ta\_chat.setRows(5);

jScrollPane1.setViewportView(ta\_chat);

b\_send.setText("SEND");

b\_send.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_sendActionPerformed(evt);

}

});

lb\_username.setText("Username :");

lb\_address.setText("Address : ");

tf\_username.setText("Hatsu");

tf\_username.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

tf\_usernameActionPerformed(evt);

}

});

tf\_address.setText("192.168.0.103");

tf\_address.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

tf\_addressActionPerformed(evt);

}

});

lb\_port.setText("Port :");

lb\_password.setText("Password : ");

tf\_port.setText("2222");

tf\_port.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

tf\_portActionPerformed(evt);

}

});

pf\_password.setText("1234");

b\_connect.setText("Connect");

b\_connect.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_connectActionPerformed(evt);

}

});

b\_disconnect.setText("Disconnect");

b\_disconnect.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

b\_disconnectActionPerformed(evt);

}

});

jSeparator3.setOrientation(javax.swing.SwingConstants.VERTICAL);

ta\_oluser.setColumns(20);

ta\_oluser.setEditable(false);

ta\_oluser.setRows(10);

ta\_oluser.setAutoscrolls(false);

jScrollPane3.setViewportView(ta\_oluser);

jLabel5.setText("Online Users");

jMenu1.setText("File");

jMenuBar1.add(jMenu1);

jMenu2.setText("Edit");

jMenuBar1.add(jMenu2);

jMenu3.setText("Resources");

mi\_sendf.setText("Send File");

mi\_sendf.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

mi\_sendfActionPerformed(evt);

}

});

jMenu3.add(mi\_sendf);

mi\_recf.setText("Receive File");

mi\_recf.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

mi\_recfActionPerformed(evt);

}

});

jMenu3.add(mi\_recf);

mi\_cloud.setText("Cloud");

mi\_cloud.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

mi\_cloudActionPerformed(evt);

}

});

jMenu3.add(mi\_cloud);

mi\_upcloud.setText("Upload to cloud");

mi\_upcloud.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

mi\_upcloudActionPerformed(evt);

}

});

jMenu3.add(mi\_upcloud);

jMenuBar1.add(jMenu3);

setJMenuBar(jMenuBar1);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jSeparator1, javax.swing.GroupLayout.DEFAULT\_SIZE, 480, Short.MAX\_VALUE)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(layout.createSequentialGroup()

.addComponent(lb\_address, javax.swing.GroupLayout.PREFERRED\_SIZE, 62, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(tf\_address, javax.swing.GroupLayout.PREFERRED\_SIZE, 91, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(lb\_port, javax.swing.GroupLayout.PREFERRED\_SIZE, 53, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(72, 72, 72))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addComponent(lb\_username, javax.swing.GroupLayout.PREFERRED\_SIZE, 62, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(10, 10, 10)

.addComponent(tf\_username, javax.swing.GroupLayout.PREFERRED\_SIZE, 89, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(lb\_password)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(tf\_port, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(pf\_password, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(21, 21, 21))

.addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED\_SIZE, 371, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jSeparator3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(16, 16, 16)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(b\_connect, javax.swing.GroupLayout.PREFERRED\_SIZE, 85, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_disconnect, javax.swing.GroupLayout.Alignment.TRAILING))

.addGap(0, 0, Short.MAX\_VALUE))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(tf\_chat, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT\_SIZE, 351, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 351, Short.MAX\_VALUE))

.addGap(10, 10, 10)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane3, 0, 0, Short.MAX\_VALUE)

.addComponent(jLabel5)

.addComponent(b\_send, javax.swing.GroupLayout.DEFAULT\_SIZE, 119, Short.MAX\_VALUE))))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(lb\_address)

.addComponent(lb\_port)

.addComponent(tf\_address, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_connect)

.addComponent(tf\_port, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(8, 8, 8)

.addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED\_SIZE, 10, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(lb\_username)

.addComponent(lb\_password)

.addComponent(tf\_username, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(pf\_password, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(b\_disconnect)))

.addComponent(jSeparator3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED\_SIZE, 10, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(4, 4, 4)

.addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 262, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, 127, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 25, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(tf\_chat, javax.swing.GroupLayout.DEFAULT\_SIZE, 24, Short.MAX\_VALUE)

.addComponent(b\_send, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addContainerGap())

);

java.awt.Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();

setBounds((screenSize.width-516)/2, (screenSize.height-507)/2, 516, 507);

}// </editor-fold>

private void b\_sendActionPerformed(java.awt.event.ActionEvent evt) {

String nothing = "";

if ((tf\_chat.getText()).equals(nothing))

{

tf\_chat.setText("");

tf\_chat.requestFocus();

}

else

{

try

{

encrypt=encryptText(username + ":" + tf\_chat.getText() + ":" + "Chat");

writer.println(encrypt);

writer.flush(); // flushes the buffer

}

catch (Exception ex) {ta\_chat.append("Message was not sent. \n");}

tf\_chat.setText("");

tf\_chat.requestFocus();

}

}

private void b\_disconnectActionPerformed(java.awt.event.ActionEvent evt) {

sendDisconnect();

Disconnect();

}

private void b\_connectActionPerformed(java.awt.event.ActionEvent evt) {

if (isConnected == false)

{

username = tf\_username.getText();

password = pf\_password.getText();

tf\_username.setEditable(false);

pf\_password.setEditable(false);

address = tf\_address.getText();

port = Integer.parseInt(tf\_port.getText());

try

{

sock = new Socket(address, port);

InputStreamReader streamreader = new InputStreamReader(sock.getInputStream());

reader = new BufferedReader(streamreader);

writer = new PrintWriter(sock.getOutputStream());

is = (sock.getInputStream());

encrypt=encryptText(username + ":has connected.:Connect:"+password);

writer.println(encrypt);

writer.flush();

isConnected = true;

}

catch (Exception ex)

{

ta\_chat.append("Cannot Connect! Try Again. \n");

tf\_username.setEditable(true);

pf\_password.setEditable(true);

}

ListenThread();

//capture a=new capture();

//a.start();

}

else if (isConnected == true)

{

ta\_chat.append("You are already connected. \n");

}

}

private void tf\_portActionPerformed(java.awt.event.ActionEvent evt) {

}

private void tf\_usernameActionPerformed(java.awt.event.ActionEvent evt) {

}

private void tf\_addressActionPerformed(java.awt.event.ActionEvent evt) {

}

private void mi\_sendfActionPerformed(java.awt.event.ActionEvent evt) {

f\_sendf.setVisible(true);

}

private void mi\_recfActionPerformed(java.awt.event.ActionEvent evt) {

f\_recf.setVisible(true);

}

private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jTextField4ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void b\_psendActionPerformed(java.awt.event.ActionEvent evt) {

}

private void b\_recfActionPerformed(java.awt.event.ActionEvent evt) {

try {

receiveFile();

} catch (IOException ex) {

}

}

private void b\_targetActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser chooser= new JFileChooser();

chooser.setFileSelectionMode(JFileChooser.DIRECTORIES\_ONLY);

chooser.showOpenDialog(null);

tf\_setpath.setText(chooser.getSelectedFile().getAbsolutePath());

}

private void mi\_cloudActionPerformed(java.awt.event.ActionEvent evt) {

try {

String str= username + ":Cloud:Cloud";

encrypt=encryptText(str);

writer.println(encrypt);

writer.flush();

d\_cloud.setVisible(true);

} catch (Exception ex) {

}

}

private void b\_downloadActionPerformed(java.awt.event.ActionEvent evt) {

if(!"".equals(tf\_target.getText()))

{

String str=tf\_cfile.getText();

String FILE\_TO\_RECEIVED2 = tf\_target.getText() + "\\" + str;

int FILE\_SIZE2=6022386;

int bytesRead;

int current = 0;

FileOutputStream fos = null;

BufferedOutputStream bos = null;

Socket sock2 = null;

try {

sock2 = new Socket(address,2223);

byte [] mybytearray = new byte [FILE\_SIZE2];

InputStream is2 = sock2.getInputStream();

fos = new FileOutputStream(FILE\_TO\_RECEIVED2);

bos = new BufferedOutputStream(fos);

bytesRead = is2.read(mybytearray,0,mybytearray.length);

current = bytesRead;

do {

bytesRead =

is2.read(mybytearray, current, (mybytearray.length-current));

if(bytesRead >= 0) current += bytesRead;

} while(bytesRead > -1);

bos.write(mybytearray, 0 , current);

bos.flush();

ta\_reccloud.append("File " + FILE\_TO\_RECEIVED2 + " downloaded (" + current + " bytes read)\n");

}

catch (UnknownHostException ex) {

} catch (IOException ex) {

} finally {

try {

if (fos != null) fos.close();

if (bos != null) bos.close();

if (sock2 != null) sock2.close();

} catch (IOException ex) {

}

}

}

else

tf\_setpath.setText("Select path");

}

private void b\_targetcloudActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser chooser= new JFileChooser();

chooser.setFileSelectionMode(JFileChooser.DIRECTORIES\_ONLY);

chooser.showOpenDialog(null);

tf\_target.setText(chooser.getSelectedFile().getAbsolutePath());

}

private void b\_browseActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser chooser = new JFileChooser();

chooser.showOpenDialog(null);

sendf=chooser.getSelectedFile();

tf\_sendf.setText(sendf.getAbsolutePath());

}

private void b\_sendfActionPerformed(java.awt.event.ActionEvent evt) {

try {

sendf(sendf);

} catch (IOException ex) {

}

}

private void cb\_extensionActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void cb\_senderActionPerformed(java.awt.event.ActionEvent evt) {

String ipselect;

ipselect = (String)cb\_sender.getSelectedItem();

if(ipselect.equals("Server"))

tf\_ip.setText(address);

if(ipselect.equals("Client"))

tf\_ip.setText("");

// TODO add your handling code here:

}

private void l\_cloudMouseClicked(java.awt.event.MouseEvent evt) {

String fname= (String) l\_cloud.getSelectedValue();

tf\_cfile.setText(fname);

try {

encrypt=encryptText(username + ":" + tf\_cfile.getText() + ":SendCloud");

// l\_cloud.setEnable(false);

} catch (Exception ex) {

}

writer.println(encrypt);

writer.flush();

}

private void b\_cbrowseActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser chooser = new JFileChooser();

chooser.showOpenDialog(null);

sendf=chooser.getSelectedFile();

tf\_upcloud.setText(sendf.getAbsolutePath());

filename=chooser.getSelectedFile().getName();

}

private void mi\_upcloudActionPerformed(java.awt.event.ActionEvent evt) {

d\_upcloud.setVisible(true);

}

private void b\_upcloudActionPerformed(java.awt.event.ActionEvent evt) {

try {

encrypt=encryptText(username + ":" + filename + ":" + "ReceiveCloud:" + InetAddress.getLocalHost().getHostAddress());

writer.println(encrypt);

writer.flush();

} catch (Exception ex) {

}

try {

upcloud(sendf);

} catch (IOException ex) {

ex.printStackTrace();

}

}

public static void main(String args[])

{

Path path1,path2,path3,path4;

File dir1;

path1 = FileSystems.getDefault().getPath("D:\\Cloud Client");

path2 = FileSystems.getDefault().getPath("D:\\Cloud Client\\Screenshots");

path3 = FileSystems.getDefault().getPath("D:\\Cloud Client\\Logs");

if(Files.notExists(path1))

{

dir1=new File("D:\\Cloud Client");

Boolean mk=dir1.mkdir();

}

if(Files.notExists(path2))

{

dir1=new File("D:\\Cloud Client\\Screenshots");

Boolean mk=dir1.mkdir();

}

if(Files.notExists(path3))

{

dir1=new File("D:\\Cloud Client\\Logs");

Boolean mk=dir1.mkdir();

}

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException | InstantiationException | IllegalAccessException | javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(client\_frame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

java.awt.EventQueue.invokeLater(new Runnable()

{

@Override

public void run()

{

new client\_frame().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton b\_browse;

private javax.swing.JButton b\_cbrowse;

private javax.swing.JButton b\_connect;

private javax.swing.JButton b\_disconnect;

private javax.swing.JButton b\_download;

private javax.swing.JButton b\_psend;

private javax.swing.JButton b\_recf;

private javax.swing.JButton b\_send;

private javax.swing.JButton b\_sendf;

private javax.swing.JButton b\_target;

private javax.swing.JButton b\_targetcloud;

private javax.swing.JButton b\_upcloud;

private javax.swing.JComboBox cb\_extension;

private javax.swing.JComboBox cb\_sender;

private javax.swing.JDialog d\_cloud;

private javax.swing.JDialog d\_login;

private javax.swing.JDialog d\_private;

private javax.swing.JDialog d\_upcloud;

private javax.swing.JFrame f\_recf;

private javax.swing.JFrame f\_sendf;

private javax.swing.JFileChooser fc\_up;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel14;

private javax.swing.JLabel jLabel15;

private javax.swing.JLabel jLabel16;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JMenu jMenu1;

private javax.swing.JMenu jMenu2;

private javax.swing.JMenu jMenu3;

private javax.swing.JMenuBar jMenuBar1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JScrollPane jScrollPane4;

private javax.swing.JScrollPane jScrollPane5;

private javax.swing.JScrollPane jScrollPane6;

private javax.swing.JScrollPane jScrollPane7;

private javax.swing.JScrollPane jScrollPane8;

private javax.swing.JScrollPane jScrollPane9;

private javax.swing.JSeparator jSeparator1;

private javax.swing.JSeparator jSeparator2;

private javax.swing.JSeparator jSeparator3;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

private javax.swing.JTextField jTextField4;

private javax.swing.JList l\_cloud;

private javax.swing.JLabel l\_error;

private javax.swing.JLabel lb\_address;

private javax.swing.JLabel lb\_password;

private javax.swing.JLabel lb\_port;

private javax.swing.JLabel lb\_username;

private javax.swing.JMenuItem mi\_cloud;

private javax.swing.JMenuItem mi\_recf;

private javax.swing.JMenuItem mi\_sendf;

private javax.swing.JMenuItem mi\_upcloud;

private javax.swing.JPasswordField pf\_password;

private javax.swing.JTextArea ta\_chat;

private javax.swing.JTextArea ta\_oluser;

private javax.swing.JTextArea ta\_poluser;

private javax.swing.JTextArea ta\_private;

private javax.swing.JTextArea ta\_reccloud;

private javax.swing.JTextArea ta\_recf;

private javax.swing.JTextArea ta\_sendf;

private javax.swing.JTextArea ta\_upcloud;

private javax.swing.JTextField tf\_address;

private javax.swing.JTextField tf\_cfile;

private javax.swing.JTextField tf\_chat;

private javax.swing.JTextField tf\_ip;

private javax.swing.JTextField tf\_port;

private javax.swing.JTextField tf\_private;

private javax.swing.JTextField tf\_receiver;

private javax.swing.JTextField tf\_sendf;

private javax.swing.JTextField tf\_setpath;

private javax.swing.JTextField tf\_target;

private javax.swing.JTextField tf\_upcloud;

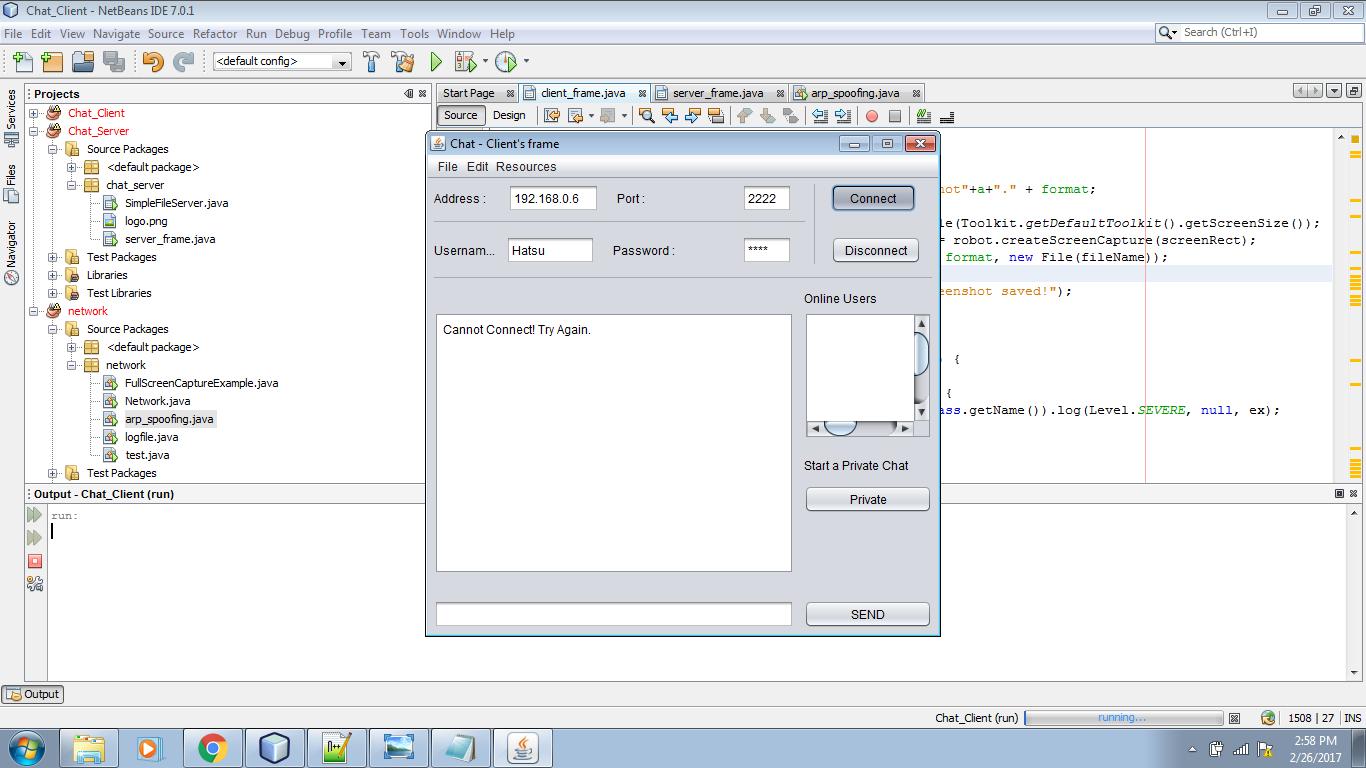
private javax.swing.JTextField tf\_username;

// End of variables declaration

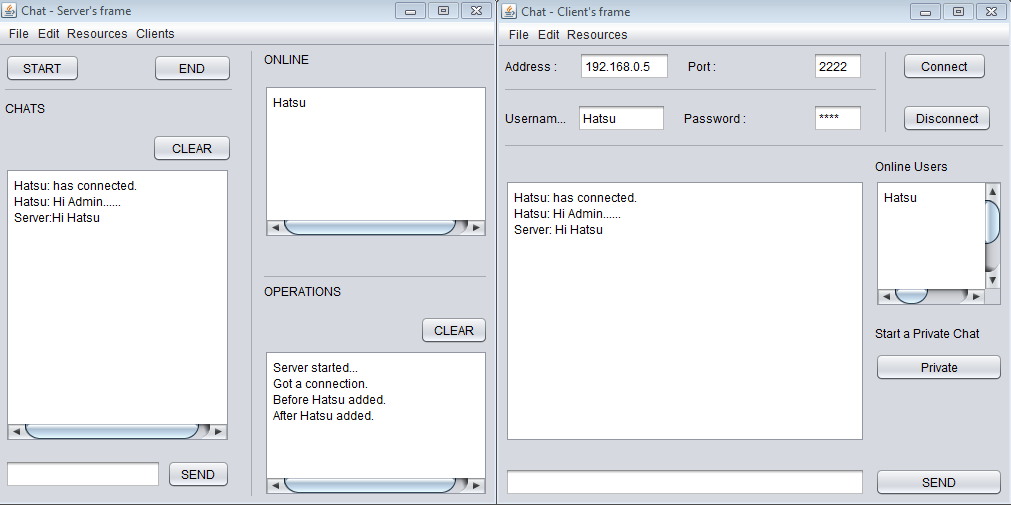
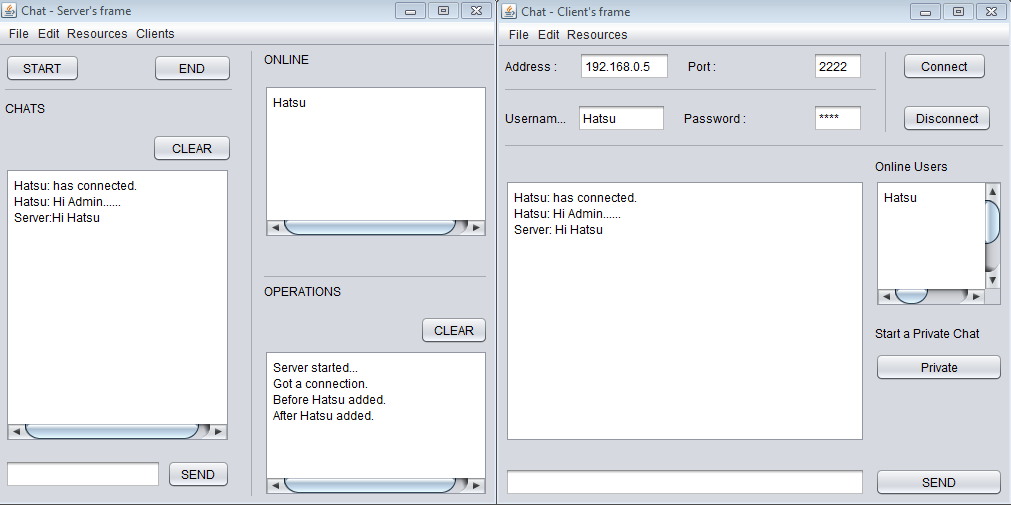
}

**SCREENSHOTS –**

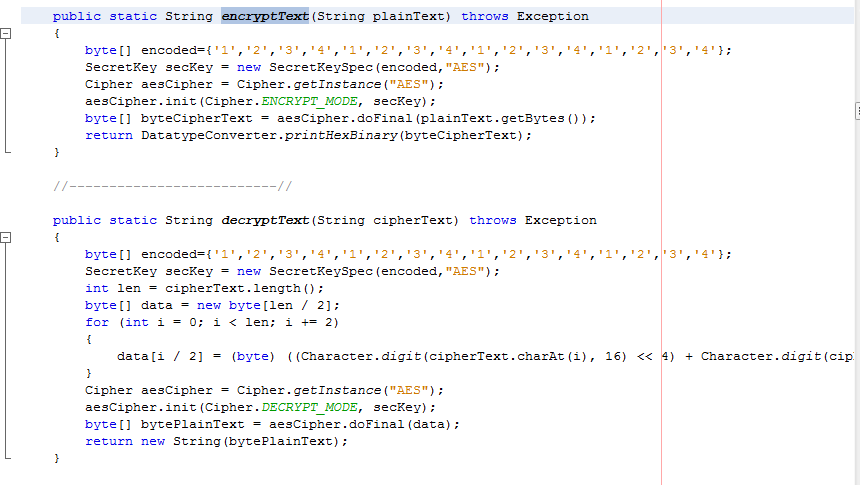
**Login Module :**

****

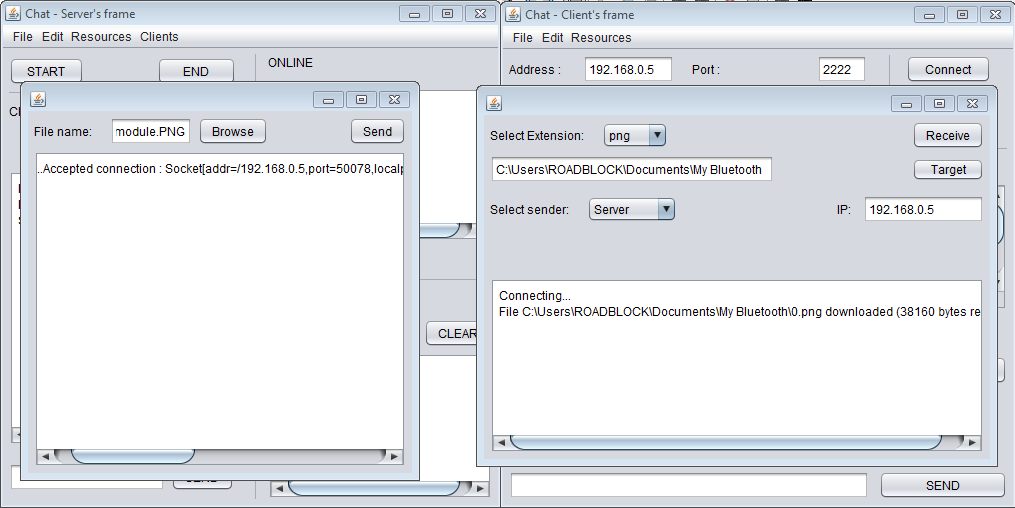
**Chat Module :**

****

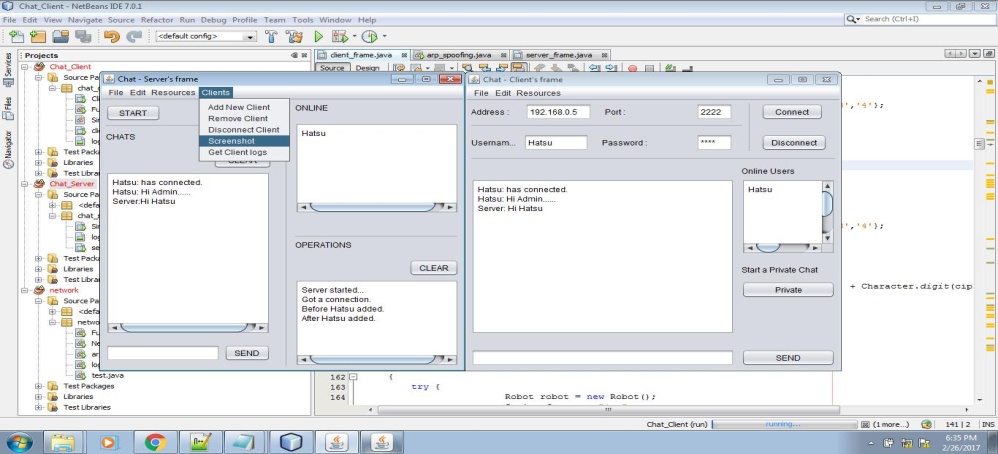
**Encryption & Decryption Module :**

****

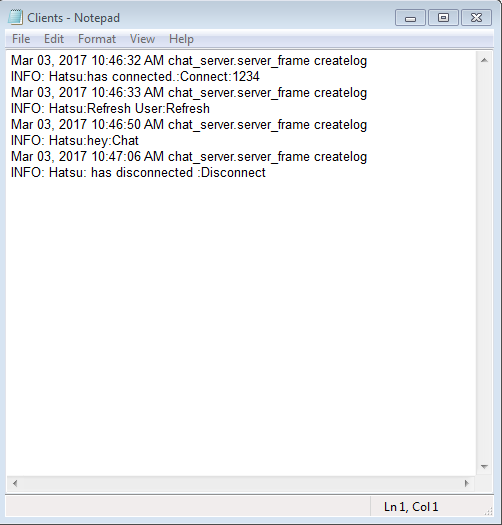
**File Transfer Module :**

****

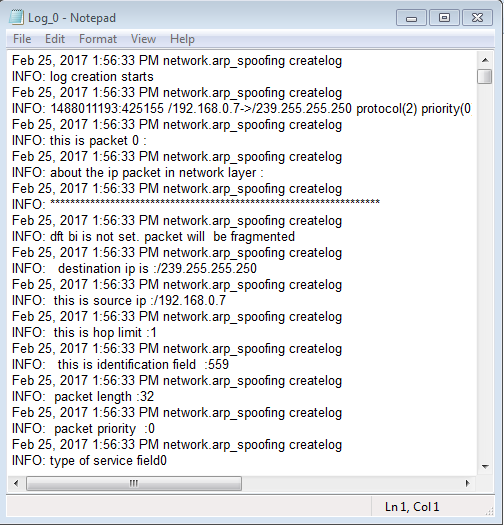
**Screenshot Module :**

****

**Log Files Module :**

****

**Packet Filter Module :**

****

Chapter 6

**TEST CASES**

**6.1 Log in :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Objective** | **Steps to Execute** | **Test Data/Input** | **Expected Result** | **Actual Result** | **Status** |
| TC\_1 | Check log in page by leaving all field as blank | Click Log in Button | N/A | By leaving all fields blank and clicking login button mandatory symbol(\*) should appear | After leaving all fields blank and clicking on login button mandatory symbol(\*) appeared | Pass |
| TC\_2 | Check invalid Username | Enter invalid username | Username :  Roadblock1 | By entering invalid username an error message should occur “User is not authorized” | After entering invalid username an error message occurred as  “Enter valid username” | Pass |
| TC\_3 | Check valid username | Enter valid username | Username :  Admin | It should allow user to proceed | It allows the user to proceed further | Pass |
| TC\_4 | Check password field | N/A | N/A | The password should be displayed in encrypted format (\*\*\*\*) | The password gets entered in encrypted format (\*\*\*\*) | Pass |
| TC\_5 | Check invalid password | Enter invalid password | Password :  \*\*\*\* | Entering wrong password should display message as  “Invalid password” | After entering wrong password , message appeared as  “Enter correct password” | Pass |
| TC\_6 | Check valid password | Enter valid password | Password :  \*\*\* | It should authenticate user | User is authenticated successfully | Pass |
| TC\_7 | Adding new User | Click on New User button | All user details and proper inputs | It should open a newwindow for entering details | It opens a new window where details are to be filled | Pass |

**6.2 Chatting :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Objective** | **Steps To Executive** | **Test Data/Input** | **Expected Result** | **Actual Result** | **Status** |
| TC\_1 | Get Logged in first | Click on Login | Enter valid username and password | By clicking on log in , chatting screen occurs | After clicking on log in , chatting window occurs | Pass |
| TC\_2 | Check how many users are active | Click on online user’s button | N/A | By clicking on online user button , online users should be displayed | After clicking on button , all active users list is displayed | Pass |
| TC\_3 | Check whether all users are able to view the message | N/A | N/A | All messages should be displayed on chat screen | All messages are displayed on the each client side | Pass |
| TC\_4 | Verify all messages are visible to server | N/A | N/A | All messages transferred should be visible at server side | All messages get displayed to server transferred between clients | Pass |
| TC\_5 | Verify if a user is allowed to send message to a specific user | Click on Private Chat Text Field | Enter valid username from online user’s list | By entering the valid username , message will be transferred to that user only | After entering valid username message is transferred to that user only | Pass |
| TC\_6 | Check message delivery time | N/A | N/A | All the messages should be sent instantly to all users | All the messages are sent immediately | Pass |
| TC\_7 | Verify file transferring | Click on file menu to select a file | Select particular file to send | Selected file should get transferred to client/server | Selected file gets transferred to the server/client | Pass |
| TC\_8 | Verify chat disconnectivity | Click on disconnect button | N/A | Clicking on disconnect,should disconnect user | User gets disconnected by clicking on disconnect | Pass |
| TC\_9 | Check if username is visible to other users | Check username field to see username | N/A | Username field should display the username | Username gets displayed in the field | Pass |
| TC\_10 | Check user visibility at server side | Click on the online user field at server side | N/A | All active users should be displayed when button is clicked | After clicking the button online users are displayed | Pass |
| TC\_11 | Check user disconnectivity to server | Click on Disconnect button at client side | N/A | Clicking should disconnect all connections with server | It disconnects user successfully | Pass |
| TC\_12 | Check Window closing function | Click on the Close Button | N/A | Clicking on the Close button should Close the window | Clicking on the Close button closes the window | Pass |
| TC\_13 | Check encryption function | N/A | Enter message | Message sent between client and server should be encrypted | Message sent between client and server is encrypted | Pass |

**6.3 File Sharing :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Objective** | **Steps to Execute** | **Test Data/Input** | **Expected Result** | **Actual Result** | **Status** |
| TC\_1 | Check for the file upload path | Click on file menu to select the file to be sent | Enter or select file to be sent | File should be properly selected | The file gets selected correctly | Pass |
| TC\_2 | Check file functionality with different extensions | N/A | N/A | Selected file with mentioned extension should be sent | File selected with the extension is sent | Pass |
| TC\_3 | Check file functionality with other extension..not mentioned | Select file with other extension and Click OK | Select file with other extension | Proper error message should be displayed with allowed extensions | The error message gets displayed when improper extension is sent | Pass |
| TC\_4 | Check file transfer cancellation | Click on the Cancel button to Cancel transferring | N/A | By clicking the Cancel button file transfer should be stopped | File transfer gets canceled by clicking the cancel button | Pass |
| TC\_5 | Check file selection shows supported files | Click on file menu to select file | Select file supported formats | Supported file formats should be selected | Supported files are selected & displayed | Pass |
| TC\_6 | Check files sent are visible | Check server or client side fields | N/A | File should be displayed to server and user if authorized | Server and authorized users are able to see file transferred | Pass |
| TC\_7 | Check file transfer with extension | Check server and client side | N/A | File should get transferred with proper extension | File gets transferred with the proper extension | Pass |
| TC\_8 | Check if logs are created | Click on View Logs at server side | N/A | File logs should be created after each file transfer | Logs are created successfully and are visible to server and authorized users | Pass |
| TC\_9 | Check if logs are accessible to users | N/A | N/A | Logs should be accessible to server and authorized users only | Logs are visible to server and authorized users | Pass |
| TC\_10 | Check if transferred file is working properly | Click on file to open it | N/A | File should be opened/executed properly | File is working properly | Pass |

**6.4 Log Files :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Objective** | **Steps to Execute** | **Test Data/Input** | **Expected Result** | **Actual Result** | **Status** |
| TC\_1 | Log File Generation | N/A | N/A | Log file should be generated after successful execution of this feature | Log Files have been successfully created | Pass |
| TC\_2 | Log File generation at user defined path | N/A | N/A | Log File should be generated at user defined path | Log File gets generated at user defined path | Pass |
| TC\_3 | Check Log File extension | Check if log files generated is as per specification | N/A | Log File should get created as per user specification | Log Files are created as per user specification | Pass |
| TC\_4 | Check Log File Content | Check if logs are getting updated properly | N/A | Log File should get updated properly with proper time stamp | Log File are updated properly with time stamp | Pass |
| TC\_5 | Check fonts and formats in Logs | Check if fonts and formats are with neat alignment | N/A | Formatting and fonts used in logs should be correct | Formatting and fonts in logs are correct and logs are neatly aligned | Pass |
| TC\_6 | Check whether the file contents get appended | N/A | N/A | Log Files content should get appended and should not be overwritten | Log File contents are properly appended without overwriting | Pass |

**6.4 Packet Filter :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Objective** | **Steps to Execute** | **Test Data/Input** | **Expected Result** | **Actual Result** | **Status** |
| TC\_1 | Check Installation of Packet Sniffer Packages | Check the packages | N/A | All the packages should be properly installed | All the packages are properly installed | Pass |
| TC\_2 | Check proper user login for packet sniffing | Check if user is logged in with proper authentication | Username and password | User should be properly logged in | User is properly logged in | Pass |
| TC\_3 | Check whether packets are sniffed or not | Start the sniffing process | Log Files | Packets should be sniffed with all the details | Packets are sniffed with all the details | Pass |
| TC\_4 | Check if all details are shown | Check the Log Files | N/A | All details should be shown properly | All details are shown properly | Pass |
| TC\_5 | Check whether packet sniffing can be stopped or not | Stop the process by Clicking Stop button | N/A | The packet sniffing should be stopped | The packet sniffing is stopped byt clicking Stop button | Pass |

Chapter 7

**7.1 SUMMERY :**

This project will help to solve many small problems related to managing servers , managing database security , sharing and storage .

**7.2 FEATURES :**

1. At a time 50 clients can be connected to the server and access the cloud services.
2. Encryption and decryption is performed at each stage of message passing.
3. Packets sent are filtered by the Packet Filter.
4. Each client activity is addressed and logs are generated.
5. Screenshots are taken when certain conditions are fulfilled.

**7.3 ADVANTAGES :**

1. The software provides high security and efficiency over network.
2. Cloud can be managed by the firm.
3. Cost Efficiency.
4. Software provides encryption and decryption

Chapter 8

**Future Scope :**

* Increase the amount of data for the client at the time of account creation for free .
* Provide Wire-less Connections .
* Provide E-mail feature from client and alarm notifications .

Chapter 9

**BIBLIOGRAPHY :**

**Books Referred –**

Java Complete Reference(J2 EE)

**Sites Referred –**

[www.google.com](http://www.google.com)

[www.stackoverflow.com](http://www.stackoverflow.com)

[www.coding4fun.com](http://www.coding4fun.com)