**Chapters**

**1.Project Definition**

**LAN chat** – Chat with Friends without the need of internet connected within the LAN

**2**.**Description**

* LAN chat application is one of easiest way to chat with a your friends through LAN.
* No internet connection is needed. The only thing which requires is server IP address and you will be able to connect to others members through LAN .
* It can help you to talk to your friends even you both do not have internet connection. As it is based on LAN

**3**. **Software and Hardware Requirement**

**Software requirements**:

Operating System: XP/Vista/windows 8/windows 10/linux.

Java IDE

**Hardware requirements**:

Processor: 1 Ghz.

Hard Disk Space: 45 MB Available HDD Space.

Memory: 512 MB RAM.

Video Card: 3D graphics accelerator equivalent to GF6200 or higher.

Sound Card: 16-bit Sound Card.

**4**. **Major Functionalities**

**LOGIN**

* In computer security, a login or logon is the process by which individual access to a computer system is controlled by identification of the user using credentials provided by the user.
* A user can log in to a system to obtain access and can then log out or log off (perform a logout / logoff) when the access is no longer needed.
* To log out is to close off one's access to a computer system after having previously logged in.
* The registered user needs to provide his username and password to authenticate his account.

**JOIN CHAT ROOM**

In this you can connect to particular room which is currently running or have at least 1 user in it .This type of room are made to share particular information. You can join any chat room if room is available and can talk to other people who are logged in right now .

**CHAT**

After logged in user is allowed to chat with each other in a group or in private. This help user to get any information or solution of their problem and also to share his knowledge with others. You can also connect with your friends only if you want to talk to him privately.

**Code**:

Adduser.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Java Chat-Admin Console-Add User</title>

</head>

<body>

<table width=700>

<tr>

<td width=300><!--%@ include file="menu.jsp"%--></td>

<td>

<div align=center>

<table>

<tr>

<td align=center height=30 width=200 bgcolor=red><a

href=adduser.jsp>Add Users</a></td>

<td align=center height=30 width=200 bgcolor=CCCCFF><a

href=viewuser.jsp>View Users</a></td>

<td align=center height=30 width=200 bgcolor=CCFFFF><a

href=AdminChatServlet>Configure Rooms</a></td>

<td align=center height=30 width=200 bgcolor=9966FF><a

href=logout.jsp>Login</a></td>

</tr>

</table>

<br>

<br>

<font size=5 color=000033>Enter New User Information</font></div>

</tr>

<tr>

<td></td>

<td>

<form action=adduserhandler.jsp method=post>

<table align=center cellpadding=5>

<tr>

<td>Name</td>

<td><input type=text name=name></td>

</tr>

<tr>

<td>Email</td>

<td><input type=text name=email></td>

</tr>

<tr>

<td>Loginid</td>

<td><input type=text name=loginid></td>

</tr>

<tr>

<td>Password</td>

<td><input type=password name=password></td>

</tr>

<tr>

<td>Type</td>

<td><select name=type>

<option value=admin>admin</option>

<option value=user>user</option>

</select></td>

<tr>

<td></td>

<td align=center><input type=submit value=Submit></td>

</tr>

</table>

</form>

</td>

</tr>

</table>

</body>

</html>

Login.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1" import="java.util.\*,java.text.\*"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Login page</title>

</head>

<body>

<table width=700>

<tr>

<td width=200><!--%@ include file="menu.jsp"%--></td>

<td>&nbsp;&nbsp;&nbsp;

<div><font color=000033 size=20>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<%

if("admin".equals(request.getParameter("type")))

{

out.print("Admin Console</font>");

}

else

{

out.print("User Console</font>");

}

DateFormat df=new SimpleDateFormat("EEEE, dd MMMM, yyyy ");

String date=df.format(new Date());

out.print("<br>");

for(int i=0;i<37;i++)

out.print("&nbsp;");

out.print("<font size=3 color=3300FF>"+date+"</font>");

%> </font></div>

</td>

</tr>

<tr>

<td></td>

<td>

<form action=loginhandler.jsp method=post>

<table align=center cellpadding=5>

<tr>

<td>UserName</td>

<td><input type=text name=userName></td>

</tr>

<tr>

<td>Password</td>

<td><input type=password name=password></td>

</tr>

<tr>

<td>Type</td>

<td>

<select name=type>

<option>admin</option>

<option>user</option>

</select>

</td>

</tr>

<tr>

<td></td>

<td align=center><input type=submit value=Submit></td>

</tr>

</table>

</form>

</td>

</tr>

</table>

</body>

</html>

ChatClient.java

package chat;

import java.awt.\*;

import java.awt.event.\*;

import java.net.\*;

import java.io.\*;

public class ChatClient extends Frame implements ActionListener,Runnable {

private Button button1, button2;

private TextArea textarea1, textarea2;

private TextField textfield1;

private Label label1, label2, label3;

Socket socket;

Thread thread;

InputStream in;

OutputStream out;

// PrintWriter out;

public static void main(String[] args)

{

new ChatClient();

}

public ChatClient()

{

setLayout(null);

label1 = new Label("Server IP address:");

label1.setBounds(35, 80, 105, 20);

add(label1);

textfield1 = new TextField("127.0.0.1");

textfield1.setBounds(145, 80, 100, 20);

add(textfield1);

button1 = new Button("Connect");

button1.setBounds(255, 80, 80, 20);

add(button1);

button1.addActionListener(this);

button2 = new Button("Send");

button2.setBounds(160, 390, 60, 20);

add(button2);

button2.addActionListener(this);

textarea1 = new TextArea("", 7, 45, TextArea.SCROLLBARS\_VERTICAL\_ONLY);

textarea1.setBounds(20, 110, 340, 120);

add(textarea1);

label2 = new Label();

label2.setBounds(20, 240, 100, 20);

label2.setText("Type here:");

add(label2);

textarea2 = new TextArea("", 7, 45,

TextArea.SCROLLBARS\_VERTICAL\_ONLY);

textarea2.setBounds(20, 260, 340, 120);

textarea2.setForeground(Color.RED);

add(textarea2);

label3 = new Label("Chat Client");

label3.setFont(new Font("Times New Roman", Font.BOLD, 36));

label3.setBounds(100, 35, 200, 30);

add(label3);

setSize(400, 430);

setTitle("Chat Client");

setVisible(true);

//textarea2.requestFocus();

this.addWindowListener(new WindowAdapter(){

public void windowClosing(

WindowEvent e){

System.exit(0);

}

}

);

}

public void actionPerformed(ActionEvent event)

{

if(event.getSource()==button1)

{

String hostname=textfield1.getText();

try

{

socket = new Socket(hostname, 13);

in=socket.getInputStream();

out=socket.getOutputStream();

textarea1.setText("you are now connected\n");

thread=new Thread(this);

thread.start();

}catch (UnknownHostException e)

{

textarea1.setText(e.getMessage());

}catch (IOException e)

{

textarea1.setText(e.getMessage());

}

}

if(event.getSource()==button2)

{

try

{

String str = textarea2.getText() + "\n";

byte buffer[] = str.getBytes();

out.write(buffer);

textarea1.setForeground(Color.RED);

textarea1.append(str+"\n");

textarea2.setText("");

textarea2.requestFocus();

}catch(Exception e){}

}

}

public void run()

{

String instring;

try {

BufferedReader in = new BufferedReader (new InputStreamReader(socket.getInputStream()));

textarea1.setForeground(Color.BLUE);

while((instring = in.readLine()) != null){

textarea1.append(instring + "\n");

}

}catch (Exception e)

{

textarea1.setText(e.getMessage());

}

}

}

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

private void initComponents() {

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

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGap(0, 400, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

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

.addGap(0, 300, Short.MAX\_VALUE)

);

pack();

}

ChatServer.java

package chat;

import java.awt.\*;

import java.awt.event.\*;

import java.net.\*;

import java.util.Date;

import java.io.\*;

public class ChatServer2 extends Frame implements ActionListener,Runnable {

private Label label1,label2;

private Button button1;

private TextArea textarea1,textarea2;

ServerSocket socket;

Socket insocket;

int port=13;

Thread thread;

ServerSocket server;

Socket connection=null;

InputStream in;

OutputStream out;

public static void main(String[] arg)

{

new ChatServer2();

}

public ChatServer2()

{

setLayout(null);

label2=new Label("Chat Server");

label2.setBounds(100,35,200,30);

label2.setFont(new Font("Times New Roman",Font.BOLD,36));

add(label2);

setSize(400,400);

setVisible(true);

setTitle("Chat Server");

button1 = new Button("Send");

button1.setBounds(160, 360, 60, 20);

add(button1);

button1.addActionListener(this);

textarea1 = new TextArea("", 7, 45, TextArea.SCROLLBARS\_VERTICAL\_ONLY);

textarea1.setBounds(20, 80, 340, 100);

//textarea1.setText("hi");

add(textarea1);

label1=new Label();

label1.setBounds(20,210,100,20);

label1.setText("Type Here");

add(label1);

textarea2 = new TextArea("", 7, 45, TextArea.SCROLLBARS\_VERTICAL\_ONLY);

textarea2.setBounds(20, 230, 340, 120);

textarea2.setForeground(Color.RED);

add(textarea2);

this.addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

try

{

socket.close();

}catch(Exception ex){}

}

});

try

{

socket = new ServerSocket(port);

insocket = null;

try {

insocket = socket.accept( );

//OutputStreamWriter out = new OutputStreamWriter(connection.getOutputStream( ));

in=insocket.getInputStream();

out=insocket.getOutputStream();

thread=new Thread(this);

thread.start();

}catch (IOException e) {}

}catch (IOException e)

{

System.err.println(e);

}

}

public void actionPerformed(ActionEvent event)

{

if(event.getSource()==button1)

{

try

{

String str = textarea2.getText() ;

byte buffer[] = str.getBytes();

out.write(buffer);

textarea1.setForeground(Color.RED);

textarea1.append(str+"\n");

textarea2.setText("");

textarea2.requestFocus();

}catch(Exception e){}

}

}

public void run()

{

String instring;

try {

BufferedReader in = new BufferedReader (new InputStreamReader(insocket.getInputStream()));

textarea1.setForeground(Color.BLUE);

while((instring = in.readLine()) != null){

textarea1.append(instring + "\n");

}

}catch (Exception e)

{

textarea1.setText(e.getMessage());

}

}

}

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

private void initComponents() {

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

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGap(0, 400, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

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

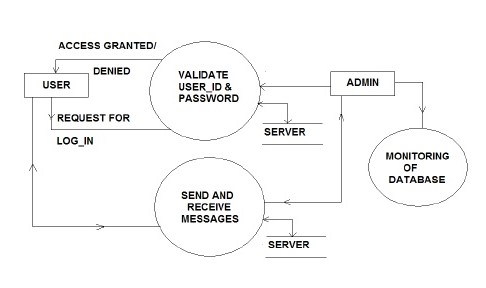
.addGap(0, 300, Short.MAX\_VALUE)

);

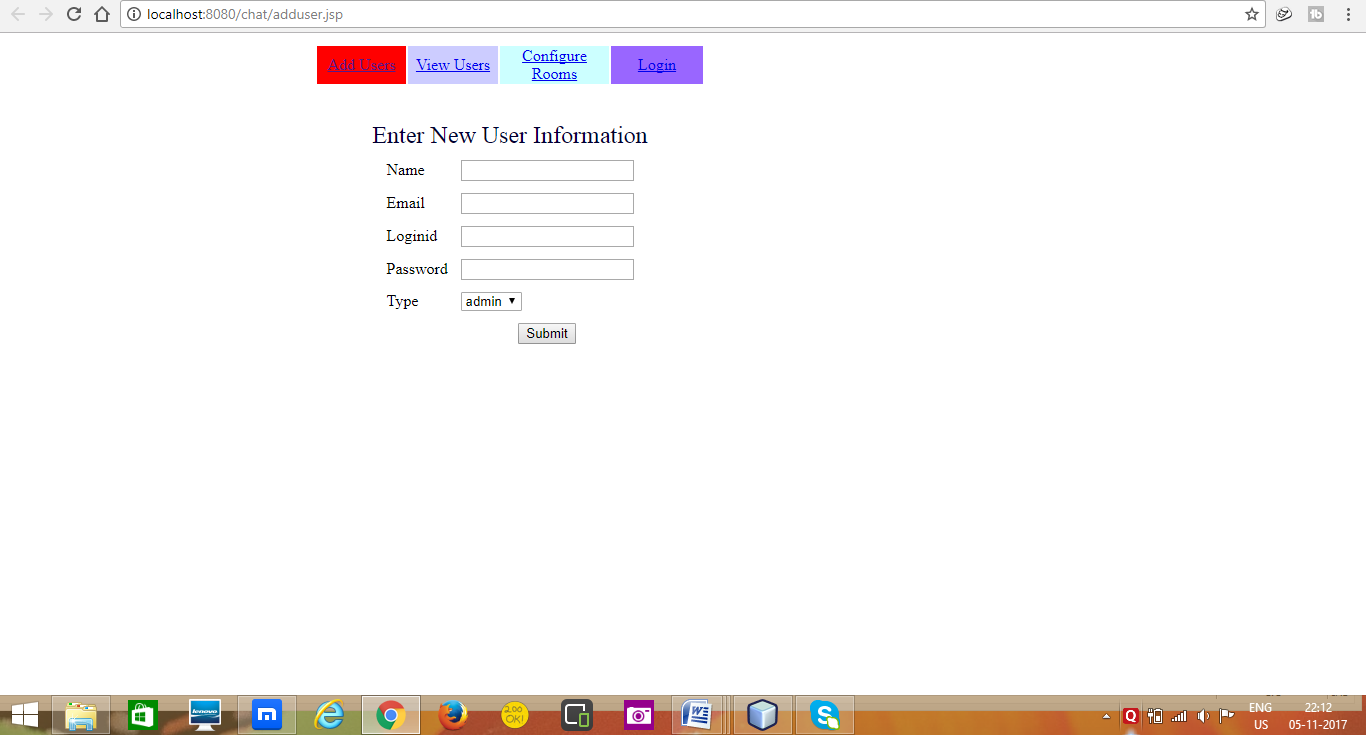
pack();

}

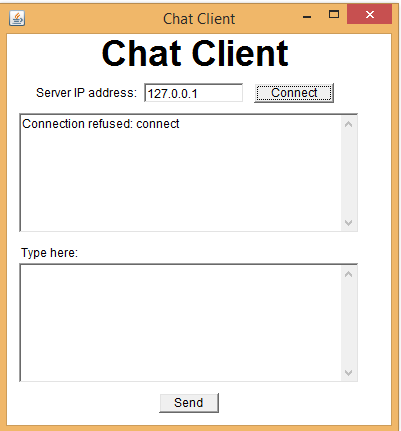
**5**. **Flow chart (Geographical Representation)**

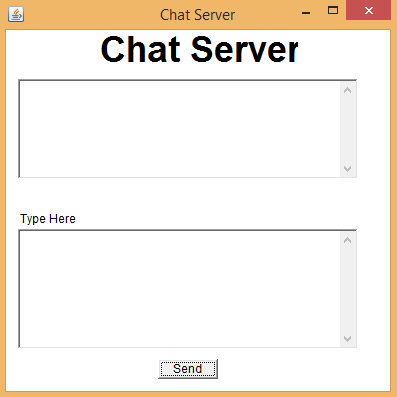


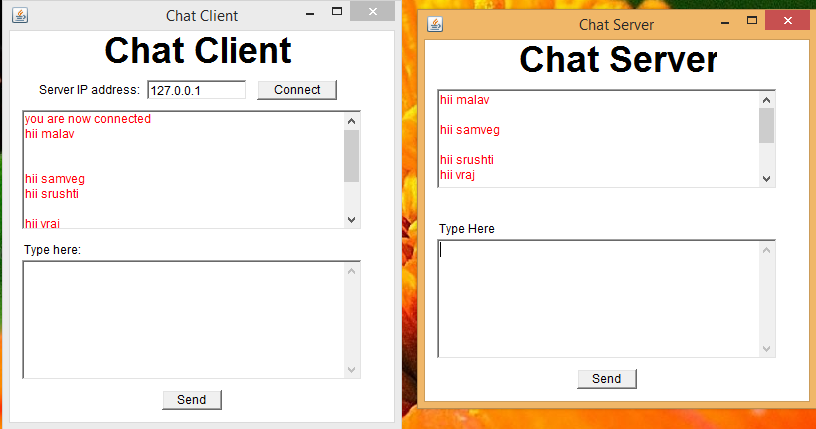
**6**. **Screenshots**

****

****

****

****

****

**7. Limitations in project**

* If server fails, whole system fails.
* You can chat only when you are connected to the same LAN.
* If you are on a different network, you cannot chat.

**8. Outcome**

One can chat for free without internet connection.

Following is the example

LAN chat is one the way can be used to talk to your friend in your circle .All you have to do is to provide his IP address and then you both will get connect to each other and then you can talk.It is helpful because if you have any problem while solving something you do not have to go to his room and to meet him and you can talk to him on chat if you both are connected to each other and can get solution of your problem .Chat application is one of the most useful software which is used by every business to communicate with his employers if he is out of station.

**9. Future Enhancement**

1. Jabber framework can be used for instant messaging.
2. Session initiation protocol [SIP] can help us to provide this TEXT
3. chat special features such as voice chat and video chat.
4. We can also add the feature of file sharing.

**10. References**

* [**http://projectsgeek.com/2011/06/lan-chat-and-file-sharing-java-project.html**](http://projectsgeek.com/2011/06/lan-chat-and-file-sharing-java-project.html)
* [**https://www.scribd.com/doc/168098480/Lan-Chat-and-File-Sharing-Java-Project**](https://www.scribd.com/doc/168098480/Lan-Chat-and-File-Sharing-Java-Project)
* [**https://www.youtube.com/watch?v=kqBmsLvWU14**](https://www.youtube.com/watch?v=kqBmsLvWU14)
* [**https://www.google.co.in/search?q=flow+chart+of+lan+chat+java&dcr=0&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjd\_\_21-KfXAhVEuI8KHRELBRoQ\_AUICigB&biw=1366&bih=662#imgrc=u8NVW7HluXXnCM**](https://www.google.co.in/search?q=flow+chart+of+lan+chat+java&dcr=0&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjd__21-KfXAhVEuI8KHRELBRoQ_AUICigB&biw=1366&bih=662#imgrc=u8NVW7HluXXnCM)**:**
* **https://stackoverflow.com/questions/11079068/creating-a-simple-lan-messenger**