Use the given files to create a complete chat system with java. simple is better.

the chat code/java files will be bellow are the 'ChatFrame' 'ChatHandler' 'ChatServer' 'ListFrame' 'MyCardFrame'.

Use card layout to obtain name from user.

Prepend name to messages sent.

Announce entrances and exits from the chat.

Working user list of all user within chat.

Private messages to user within the chat.

Please base the code off of the code already given.

//ChatFrame.java

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import java.net.\*;

public class ChatFrame extends Frame{

public ChatFrame(){

setSize(500,500);

setTitle("Echo Client");

addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent we){

System.exit(0);

}

});

ChatPanel cp = new ChatPanel();

add(cp, BorderLayout.CENTER);

setVisible(true);

}

public static void main(String[] args){

ChatFrame cf = new ChatFrame();

}

}

class ChatPanel extends Panel implements ActionListener, Runnable{

TextField tf;

TextArea ta;

Socket s;

PrintWriter send;

BufferedReader rec;

Thread t;

Button connect, disconnect;

public ChatPanel(){

setLayout(new BorderLayout());

tf = new TextField();

tf.addActionListener(this);

add(tf, BorderLayout.NORTH);

ta = new TextArea();

add(ta, BorderLayout.CENTER);

//new stuff below

connect = new Button("Connect");

connect.addActionListener(this);

disconnect = new Button("Disconnect");

disconnect.addActionListener(this);

disconnect.setEnabled(false);

Panel p = new Panel();

p.add(connect);

p.add(disconnect);

add(p, BorderLayout.SOUTH);

}

public void actionPerformed(ActionEvent ae){

if(ae.getSource()==tf){

String temp = tf.getText();

send.println(temp);

tf.setText("");

}else if(ae.getSource() == connect){

try{

s = new Socket("localhost", 3000);

send = new PrintWriter(s.getOutputStream(), true);

rec = new BufferedReader(new InputStreamReader(

s.getInputStream()));

}catch(UnknownHostException uhe){

System.out.println(uhe.getMessage());

}catch(IOException ioe){

System.out.println(ioe.getMessage());

}

t = new Thread(this, "whatever");

t.start();

connect.setEnabled(false);

disconnect.setEnabled(true);

}else if(ae.getSource() == disconnect){

//how to kill connection?

send.println("BYE");

send = null;

rec = null;

s = null;

t = null;

disconnect.setEnabled(false);

connect.setEnabled(true);

}

}

public void run(){

String temp = "";

try{

while(((temp = rec.readLine()) != null) && (t != null)){

ta.append(temp + "\n");

}

}catch(IOException ioe){

System.out.println(ioe.getMessage());

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//ChatHandler

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class ChatHandler extends Thread{

Socket s;

BufferedReader br;

PrintWriter pw;

String temp;

ArrayList handlers;

public ChatHandler(Socket s, ArrayList handlers){

this.s = s;

this.handlers = handlers;

}

public void run(){

try{

handlers.add(this);

br = new BufferedReader(new InputStreamReader(s.getInputStream()));

pw = new PrintWriter(s.getOutputStream(), true);

temp = "";

while((temp = br.readLine()) != null){

for (ChatHandler ch : handlers){

ch.pw.println(temp);

}

System.out.println(temp);

}

}catch(IOException ioe){

System.out.println(ioe.getMessage());

}finally{

handlers.remove(this);

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//ChatServer

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class ChatServer{

Socket s;

ArrayList handlers;

public ChatServer(){

try{

ServerSocket ss = new ServerSocket(3000);

handlers = new ArrayList();

for(;;){

s = ss.accept();

new ChatHandler(s, handlers).start();

}

}catch(IOException ioe){

System.out.println(ioe.getMessage());

}

}

public static void main(String[] args){

ChatServer tes = new ChatServer();

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//ListFrame

import java.awt.\*;

import java.awt.event.\*;

public class ListFrame extends Frame{

ListPanel lp;

public ListFrame(){

setSize(200,500);

setTitle("List Client");

addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent we){

System.exit(0);

}

});

lp = new ListPanel();

add(lp, BorderLayout.CENTER);

setVisible(true);

}

public static void main(String[] args){

ListFrame lf = new ListFrame();

}

}

class ListPanel extends Panel implements ActionListener{

List lst;

TextField addtf, removetf;

public ListPanel(){

setLayout(new BorderLayout());

lst = new List(4, false);

lst.add("Mercury");

lst.add("Venus");

lst.add("Earth");

add(lst, BorderLayout.CENTER);

addtf = new TextField();

addtf.addActionListener(this);

add(addtf, BorderLayout.SOUTH);

removetf = new TextField();

removetf.addActionListener(this);

add(removetf, BorderLayout.NORTH);

}

public void actionPerformed(ActionEvent ae){

if(ae.getSource() == addtf){

String temp = addtf.getText();

lst.add(temp);

addtf.setText("");

}else if(ae.getSource() == removetf){

String temp = removetf.getText();

lst.remove(temp);

removetf.setText("");

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//MyCardFrame

import java.awt.\*;

import java.awt.event.\*;

public class MyCardFrame extends Frame{

LoginPanel lp;

ChatPanel cp;

public MyCardFrame(){

setLayout(new CardLayout());

setTitle("Chat");

setSize(500,500);

lp = new LoginPanel(this);

cp = new ChatPanel();

add(lp, "login");

add(cp, "chat");

addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent we){

System.exit(0);

}

});

setVisible(true);

}

public static void main(String[] args){

MyCardFrame mcf = new MyCardFrame();

}

}

class LoginPanel extends Panel implements ActionListener{

TextField tf;

MyCardFrame mcf;

public LoginPanel(MyCardFrame mcf){

this.mcf = mcf;

tf = new TextField(20);

tf.addActionListener(this);

add(tf);

}

public void actionPerformed(ActionEvent ae){

mcf.cp.setUserName(tf.getText()); //call setUserName of chatpanel which is a member of mycardframe

CardLayout cl = (CardLayout)(mcf.getLayout());

cl.next(mcf);

tf.setText("");

}

}

class ChatPanel extends Panel{

Label label1,label2;

String userName;

public ChatPanel(){

label1 = new Label("Chat Panel: ");

label2 = new Label("Name will go here");

add(label1);

add(label2);

}

public void setUserName(String userName){

this.userName = userName;

label2.setText(getUserName());

}

public String getUserName(){

return userName;

}

}