

BÀI THỰC HÀNH 1: SOCKET VÀ THREAD

1. Tạo lớp để nhận và gửi dữ liệu thông qua Socket và ServerSocket.

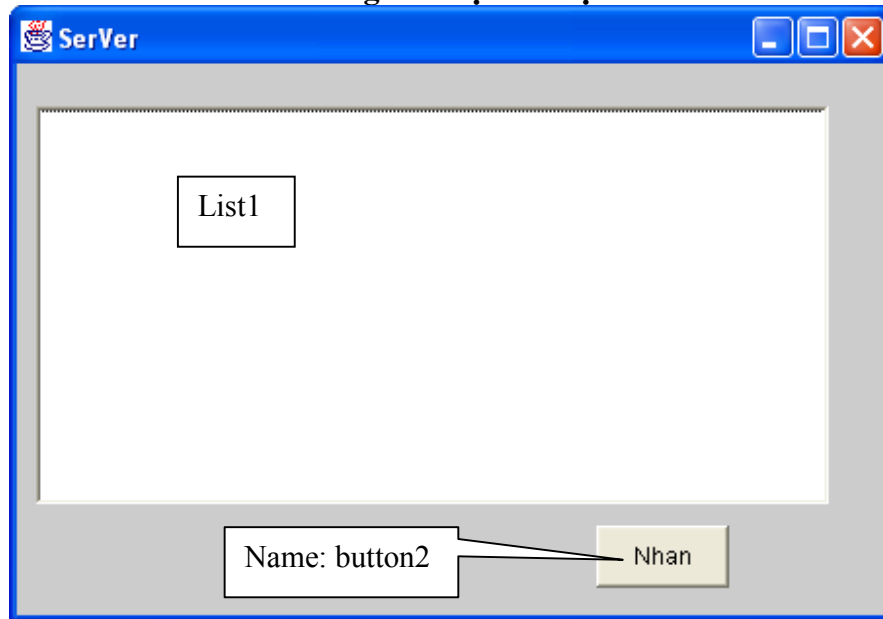
```
import java.net.*;
import java.io.*;
public class GoiNhan {
    ServerSocket Sserver ;
    //Hàm này dùng để mở 1 cổng: số hiệu cổng là cong
    public void MoCong(int cong)
    {
        try{
            Sserver= new ServerSocket(cong); //Mở 1 cổng trên Server
        } catch(Exception tb1)
        {
            System.out.print("Khong mo cong duoc") ;
        }
    }
    //Hàm Goi dùng để:
    // - Tạo Socket để kết nối đến server
    // - Tạo 1 luồng để gửi dữ liệu đi
    public void Goi(String tb,String Ip, int cong) {
        try{
            Socket Sclient= new Socket(Ip,cong); //Tạo ra 1 Socket
            // Tạo ra 1 luồng để xuất dữ liệu từ Socket: Sclient
            DataOutputStream out = new DataOutputStream(Sclient.getOutputStream());
            //Xuất dữ liệu
            // Sclient.getLocalAddress().getHostAddress(): Lấy địa chỉ Ip của máy
            out.writeBytes(Sclient.getLocalAddress().getHostAddress()+":"+ tb);
            out.write(13); //Xuất thêm ký tự Enter (xuống dòng và về đầu dòng)
            out.write(10);
            out.close() ;
            Sclient.close() ;
        } catch(Exception tb2)
        {
            System.out.print("Khong goi duoc") ;
        }
    }
    public String Nhan() { //Hàm này dùng để nhận thông báo gửi về từ client
        try{
            Socket nhan= Sserver.accept(); //Server đồng ý nhận dữ liệu
            //Tạo ra 1 luồng để nhận dữ liệu từ Socket: nhan
            InputStreamReader in = new InputStreamReader(nhan.getInputStream());
            //Tạo ra 1 bộ đệm để đọc dữ liệu từ luồng
            BufferedReader buff = new BufferedReader(in);
            return buff.readLine(); //Nhận về 1 dòng dữ liệu
        } catch(Exception tb3)
        {
            System.out.print("Khong goi duoc") ;
            return null;
        }
    }
}
```

```

    }
}
} //Đóng lớp

```

2. Tạo ra 1 Frame: Fserver để mở cổng và nhận dữ liệu về:



- Khai báo thêm gói:

```
import java.net.*;
import java.io.*;
```
- Lập trình trên sự kiện của Frame để mở ra 1 cổng:
GoiNhan gn= new GoiNhan();//Tạo ra 1 lớp GoiNhan ở ngoài tất cả các hàm
void this_windowOpened(WindowEvent e) {
 gn.MoCong(1260) ;
 System.out.print("Da mo cong");
}
- Tạo thêm hàm main trong Frame: Fserver để chạy chương trình như sau:
public static void main(String[] ss)
 {
 FServer f= new FServer();
 f.show() ;
 }
- Chạy Frame: Fserver để xem thử đã mở cổng được hay chưa:
 - Start -> run- > Gõ cmd để ra cửa sổ Dos
 - Gõ lệnh: netstat -a:

```

C:\WINDOWS\system32\cmd.exe

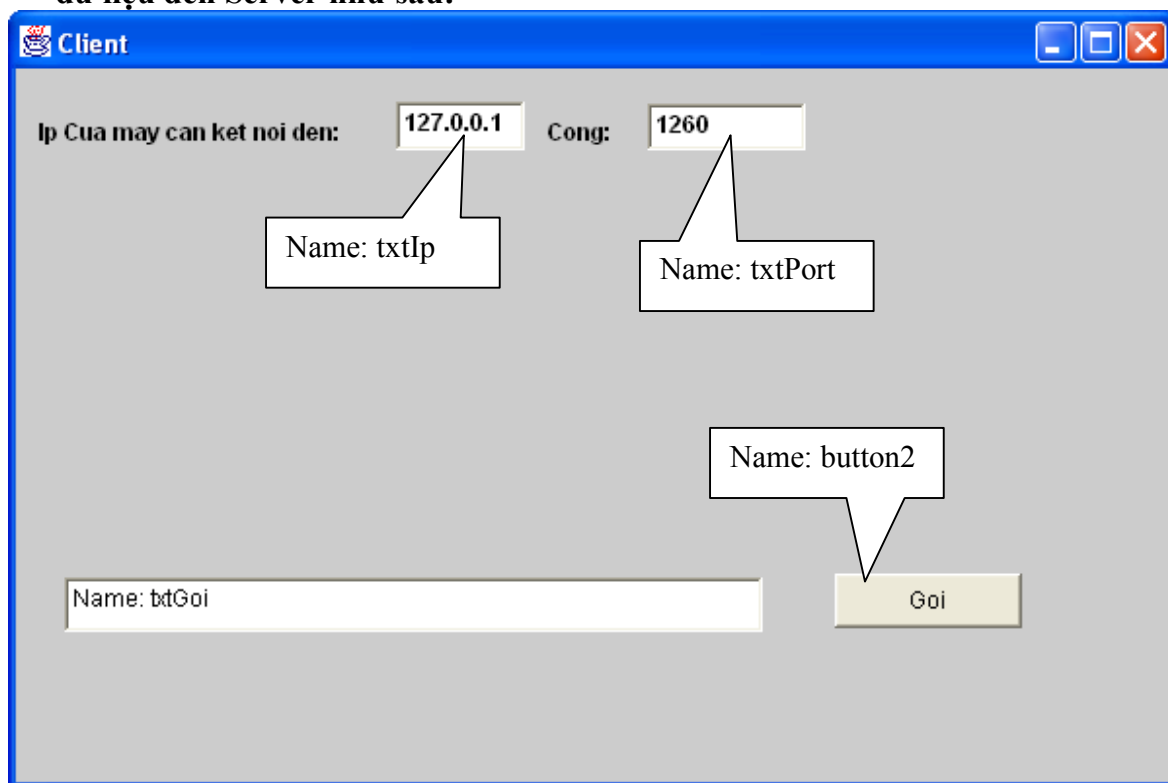
C:\Documents and Settings\Administrator>netstat -a

Active Connections

Proto Local Address           Foreign Address         State
TCP    nhha:smtp                0.0.0.0:0              LISTENING
TCP    nhha:http                0.0.0.0:0              LISTENING
TCP    nhha:epmap               0.0.0.0:0              LISTENING
TCP    nhha:https               0.0.0.0:0              LISTENING
TCP    nhha:microsoft-ds        0.0.0.0:0              LISTENING
TCP    nhha:1026                0.0.0.0:0              LISTENING
TCP    nhha:1260                0.0.0.0:0              LISTENING
TCP    nhha:1031                0.0.0.0:0              LISTENING
TCP    nhha:8888                0.0.0.0:0              LISTENING
TCP    nhha:nethbios-ssn        0.0.0.0:0              LISTENING
UDP    nhha:microsoft-ds        *:*:
UDP    nhha:isakmp              *:*:
UDP    nhha:ms-sql-m            *:*:
UDP    nhha:3456                *:*:
UDP    nhha:4500                *:*:
UDP    nhha:ntp                 *:*:
UDP    nhha:1027                *:*:
UDP    nhha:1900                *:*:
UDP    nhha:ntp                 *:*:
UDP    nhha:nethbios-ns         *:*:
UDP    nhha:nethbios-dgm        *:*:

```

3. Tạo ra 1 Frame: FClient để tạo ra 1 Socket kết nối đến Server qua cổng 1260 và gửi dữ liệu đến Server như sau:



- a. Lập trình trên sự kiện **actionPerformed** của nút gửi để gửi đến Server nội dung gõ trên txtGoi:

```

GoiNhan gn = new GoiNhan();//Tạo lớp GoiNhan ở ngoài tất cả các hàm
void button2_actionPerformed(ActionEvent e) {
    gn.Goi(txtGoi.getText(),txtIp.getText(),Integer.parseInt(txtPort.getText()));
}

```

- b. Mở Frame: Fserver và lập trình trên sự kiện **actionPerformed** của nút Nhận để nhận dữ liệu về và hiển thị dữ liệu:

```

void button2_actionPerformed(ActionEvent e) {

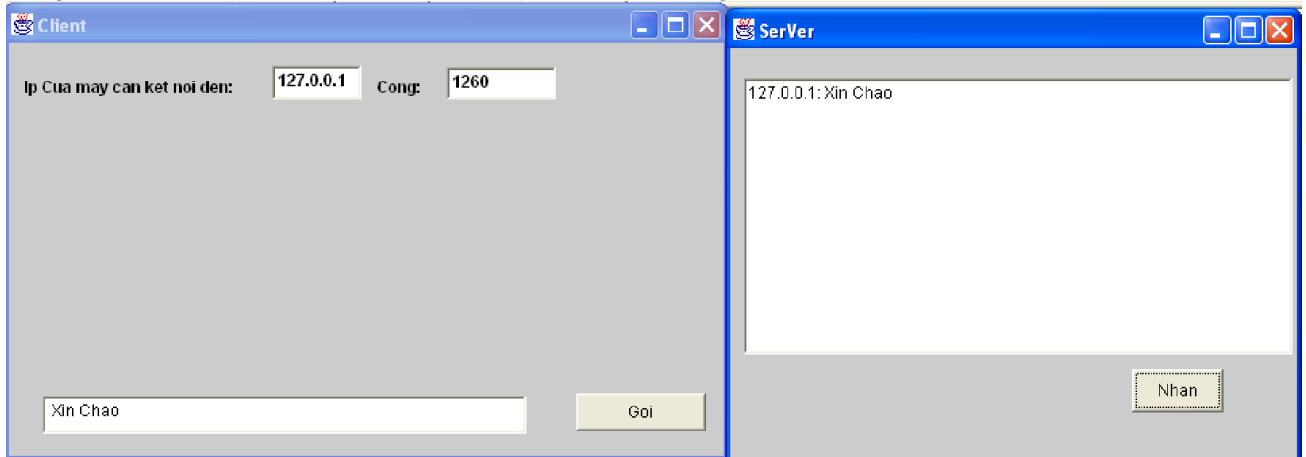
```

```
list1.add(gn.Nhan());
}
```

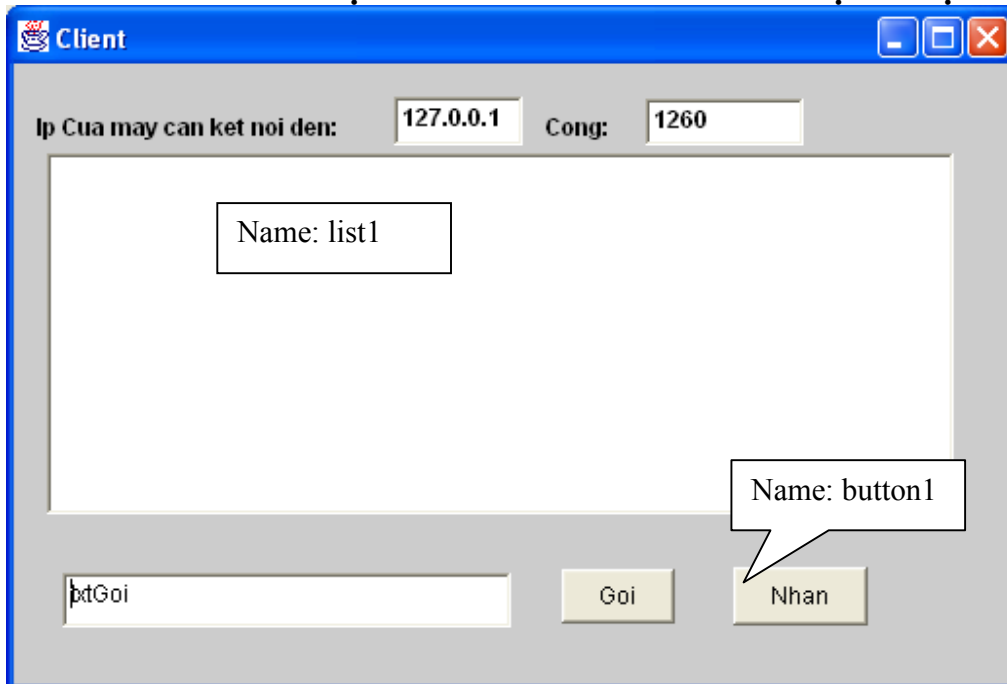
c. Tạo thêm hàm main trong Frame: Fclient để chạy chương trình như sau:

```
public static void main(String[] ss)
{
    FClient f= new FClient();
    f.show() ;
}
```

d. Chạy Frame: Fserver để mở cổng, sau đó chạy Frame: Fclient để nhập dữ liệu và gửi dữ liệu, sang Frame: Fserver chọn nút Nhan để nhận dữ liệu gửi về:



4. Mở Frame: Fclient tạo thêm 1 list1 và nút Nhan để nhận dữ liệu từ Server:



a. Lập trình trên sự kiện windowOpened của Frame: Fclient để mở cổng 1261:

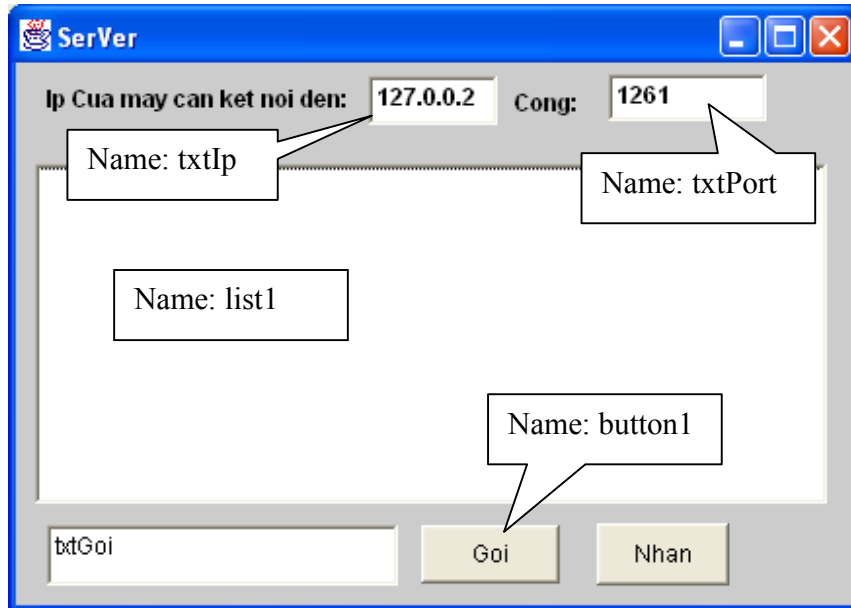
```
this_windowOpened(WindowEvent e) {
    gn.MoCong(1261);
}
```

b. Lập trình trên sự kiện actionPerformed của nút Nhan để nhận dữ liệu từ Server như sau:

```
void button1_actionPerformed(ActionEvent e) {
```

```
list1.add(gn.Nhan());
}
```

5. Mở Frame: FServer tạo thêm 3 Text Box và nút Goi:



+ Lập trình trên sự kiện của nút Goi để gửi dữ liệu đến Client:

```
void button1_actionPerformed(ActionEvent e) {
    gn.Goi(txtGoi.getText(),txtIp.getText(),Integer.parseInt(txtPort.getText()));
}
```

6. Chạy 2 Frame: Fclient và FServer để gửi nhận dữ liệu từ Server và Client.

7. Tạo 1 tuyệt đoạn trên Frame: FServer để nhận dữ liệu tự động từ Client gửi lên Server:

```
public class FServer extends JFrame {
    public class Mythread extends Thread
    { public void run()
      { while(1==1)
        { System.out.print("Dang cho :");
          list1.add(gn.Nhan());
          System.out.print("Da nhan:");
        }
      }
    }
}
```

...

+ Tại sự kiện *windowOpened* của Frame: Fserver cho tuyến đoạn hoạt động:

```
void this_windowOpened(WindowEvent e) {
    gn.MoCong(1260);
    System.out.print("Da mo cong");
    Thread tdnhan = new Thread(new Mythread());
    tdnhan.start();
}
```

8. Tạo 1 tuyệt đoạn trên Frame: Fclient để nhận dữ liệu tự động từ Server gửi lên Client:

```
public class FServer extends JFrame {
```

```

public class Mythread extends Thread
{
    public void run()
    {
        while(l==1)
        {
            System.out.print("Dang cho :");
            list1.add(gn.Nhan());
            System.out.print("Da nhan:");
        }
    }
}
...
}
+ Tại sự kiện windowOpened của Frame: Fclient cho tuyến đoạn hoạt động:
void this_ windowOpened(WindowEvent e) {
    gn.MoCong(1261) ;
    Thread tdnhan = new Thread(new Mythread());
    tdnhan.start() ;
}

```

9. Chạy 2 Frame: Fclient và FServer để gửi nhận dữ liệu từ Server và Client.

BÀI THỰC HÀNH 2: DatagramSocket, DatagramPacket

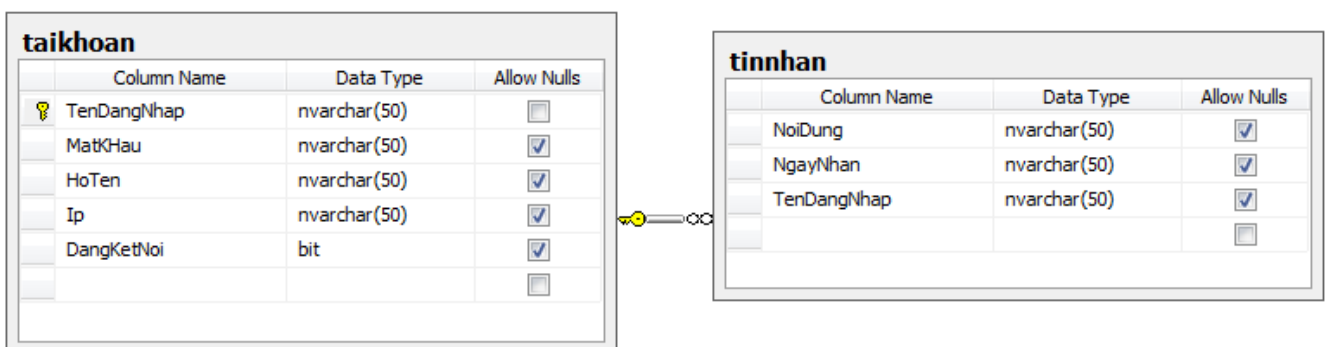
Yêu cầu:

Sử dụng 2 lớp **DatagramSocket**, **DatagramPacket** để viết chương trình chat giữa 2 máy bất kỳ, chương trình bao gồm các chức năng sau:

- Người dùng phải đăng nhập trước khi chat, khi đăng nhập thông tin của máy đăng nhập sẽ được lưu vào cơ sở dữ liệu
- Người dùng có thể đăng ký trước khi đăng nhập
- 2 máy bất kỳ có thể truyền gói tin (chat) cho nhau
- 2 máy bất kỳ có thể truyền file cho nhau
- Lưu lại các thông tin nhận được vào cơ sở dữ liệu
- Hiển thị các tin đã lưu
- Người dùng có thể đổi mật khẩu

Chuẩn bị:

1. Tạo CSDL ChatDb trong My SQL bao gồm các bảng có quan hệ như sau:



Trong đó:

- o Ip lưu lại Ip của máy đăng nhập vào
- o DangKetNoi=true nếu tên đăng nhập này đang sử dụng chương trình, ngược lại =0

2. Tạo Frame: FrmClient bao gồm menu và các điều khiển sau:

Tài khoản Thông tin cá nhân

Đăng ký Lưu tin

Thoát Hiển thị tin

Đổi mật khẩu

Chọn 1 tài khoản để nhắn tin Choice1 Nạp lại tên đăng nhập

list1

txtGoi Gửi tin Gửi file

3. Lần lượt tạo ra 4 Frame sau: FrmDangNhap, FrmDangKy, FrmHienThiTin, FrmDoiMatKhau lần lượt gắn các Frame: FrmDangKy, FrmHienThiTin, FrmDoiMatKhau vào các mục tương ứng trên menu.
4. Add thư viện: sqljdbc4.jar vào Project

Lập trình trên các lớp

1. **Tạo lớp CoSo để kết nối đến cơ sở dữ liệu ChatDb và lấy dữ liệu về từ bảng như sau:**

```
import java.sql.*;
import javax.swing.table.*;
public class CoSo {
    public static Connection cn;//Biến cn để kết nối đến CSDL
    public static String TenDn=""; //Lưu lại tên người dùng đăng nhập vào hệ thống
    public static String IpGoiDen=""; //Lưu lại Ip của máy cần gửi đến
    public static int DoDaiGoiTinNhan=0;//Lưu lại độ dài của gói tin vừa nhận
    public static void MoKetNoi() { //Mở kết nối đến CSDL
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            cn = DriverManager.getConnection("jdbc:sqlserver://tên máy:1433; databaseName=chatDb;
                user=...; password=...");
            System.out.println("Da ket noi CSDL chatDB");
        }
        catch (Exception t1) {
            System.out.print(t1.getMessage());
        }
    }
    //Truyền vào 1 bảng có tên là tb và trả về 1 ResultSet chứa toàn bộ dữ liệu của bảng này
    public static ResultSet LayBang(String tb) {
        try {
            Statement cmd = cn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
                ResultSet.CONCUR_READ_ONLY);
            return cmd.executeQuery("select * from " + tb);
        }
        catch (Exception t2) {
            System.out.print(t2.getMessage());
            return null;
        }
    }
}
```

2. **Tạo ra lớp TaiKhoan để thao tác trên bảng TaiKhoan như sau:**

```
import java.sql.*;
public class TaiKhoan {
    public int DangKetNoi(String tt,String un) {
        try{
            //Thiết lập câu lệnh sql để cập nhật lại trạng thái của tài khoản
            String sql="update TaiKhoan set DangKetNoi=? where TenDangNhap=?";
            PreparedStatement cmd= CoSo.cn.prepareStatement(sql) ;
            cmd.setString(1, tt); //Gán tham số cho câu lệnh sql
            cmd.setString(2, un);
            return cmd.executeUpdate();//thực hiện câu lệnh sql
        }catch(Exception loi) {
            System.out.print("Loi o ham loi DangKetNoi"+loi) ;
            return 0;
        }
    }
}
```



```

}
//Cập nhật lại trường Ip=ip cho tài khoản có tên đăng nhập là: un
public int LuuIp(String ip, String un) {
    try{
        //Thiết lập câu lệnh sql để cập nhật lại ip của tài khoản
        String sql="update TaiKhoan set Ip=? where TenDangNhap=?";
        PreparedStatement cmd= CoSo.cn.prepareStatement(sql) ;
        cmd.setString(1, ip); //Gán tham số cho câu lệnh sql
        cmd.setString(2, un);
        return cmd.executeUpdate();
    }catch(Exception loi) {
        System.out.print("Loi o LuuIp: "+loi) ;
        return 0;
    }
}
//Hàm trả về là true nếu tìm ra người đăng nhập có tên đăng nhập là un và mật khẩu là: pwd
public boolean ktDangNhap(String un, String pwd) {
    try{
        //Thiết lập câu lệnh sql để kiểm tra đăng nhập
        String sql="select * from TaiKhoan where tendangnhap=? and matkhau=?";
        PreparedStatement cmd= CoSo.cn.prepareStatement(sql) ;
        cmd.setString(1, un); //Gán tham số cho câu lệnh sql
        cmd.setString(2, pwd);
        ResultSet rs=cmd.executeQuery();
        return rs.next() ;
    }catch(Exception loi){
        System.out.print("Loi o ktDangNhap: "+loi) ;
        return false;
    }
}
//Hàm trả về địa chỉ Ip của người đăng nhập là un
public String LayIp(String un) {
    try{
        //Thiết lập câu lệnh sql để kiểm tra đăng nhập
        String sql="select * from TaiKhoan where tendangnhap=?";
        PreparedStatement cmd= CoSo.cn.prepareStatement(sql) ;
        cmd.setString(1, un); //Gán tham số cho câu lệnh sql
        ResultSet rs=cmd.executeQuery();
        if( rs.next() )//Nếu tìm được tài khoản: un
            return rs.getString("Ip");//Trả về Ip
        else
            return null;
    }catch(Exception loi){
        System.out.print("Loi o LayIp: "+loi) ;
        return null;
    }
}
//Hàm trả về thông tin của tất cả các tài khoản đang đăng nhập vào hệ thống: DangKetNoi=true
public ResultSet LayTaiKhoanDangKetNoi() {

```

```

try{
    Statement cmd= CoSo.cn.createStatement() ;
    String sql="select * from TaiKhoan where DangKetNoi=1";
    ResultSet rs=cmd.executeQuery(sql);
    return rs;

} catch(Exception loi)
{
    System.out.print("Loi o LayTaiKhoanDangKetNoi: "+loi) ;
    return null;
}
}
//Lấy về trạng thái đang kết nối của tài khoản có tên đăng nhập là: TenDangNhap
public String LayTrangThaiKetNoi(String TenDangNhap) {
try{
    String sql="select * from TaiKhoan where tendangnhap=?";
    PreparedStatement cmd= CoSo.cn.prepareStatement(sql) ;
    cmd.setString(1, TenDangNhap); //Gán tham số cho câu lệnh sql
    ResultSet rs=cmd.executeQuery();
    rs.next();
    return rs.getString("DangKetNoi");
} catch(Exception loi)
{
    System.out.print("Loi o LayTrangThaiKetNoi: "+loi) ;
    return null;
}
}
//Nhập vào bảng TaiKhoan 1 dòng dữ liệu
public int TaoTaiKhoan(String tenDn, String matKhau, String hoten) {
try{
    String sql="insert into TaiKhoan(TenDangNhap,MatKHau,HoTen) values(?,?,?)";
    PreparedStatement cmd= CoSo.cn.prepareStatement(sql) ;
    cmd.setString(1, tenDn); //Gán tham số cho câu lệnh sql
    cmd.setString(2, matKhau);
    cmd.setString(3, hoten);
    int kt= cmd.executeUpdate();
    return kt;
} catch(Exception loi) {
    loi.printStackTrace();
    System.out.print("Loi o TaoTaiKHOan: "+ loi.getMessage() );
    return 0;
}
}
}
//Kết thúc lớp TaiKhoan

```

3. Mở Frame FrmDangNhap thiết kế giao diện như sau:

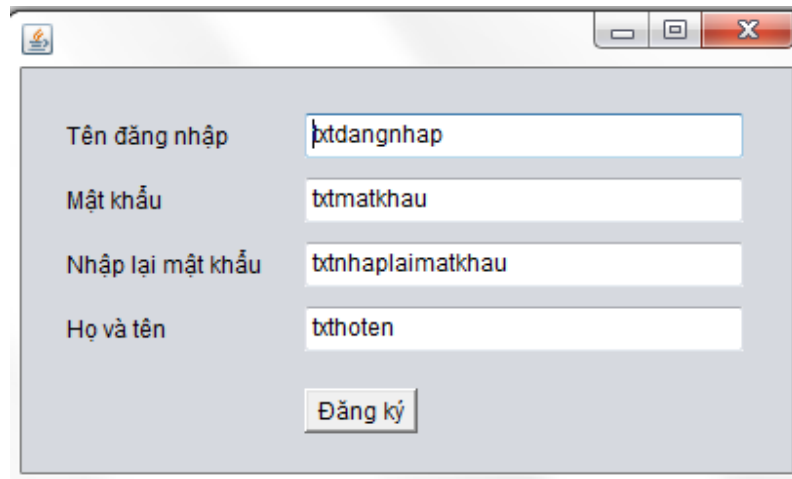
The image shows a simple login window. It has two text input fields: the first is labeled 'Tên đăng nhập' (Username) and contains the text 'txtdangnhap'; the second is labeled 'Mật khẩu' (Password) and contains the text 'txtmatkhau'. Below these fields are two buttons: 'Đăng nhập' (Login) on the left and 'Đăng ký' (Register) on the right.

- Lập trình trên sự kiện actionPerformed của butDangNhap để kiểm tra xem người dùng đăng nhập đúng hay không, nếu đúng thì:
 - o Lưu lại Ip của máy mà người dùng vừa đăng nhập vào CSDL
 - o Lưu lại thông tin của người đăng nhập
 - o Đổi trạng thái DangKetNoi=true cho tài khoản này
 - o Mở Frame: FrmClient để người dùng thao tác

```
private void butDangNhapActionPerformed(java.awt.event.ActionEvent evt) {
try{
    String un=txtdangnhap.getText() ;//Lấy tên tài khoản
    String pwd=txtmatkhau.getText() ;//Lấy mật khẩu
    CoSo.MoKetNoi();//mở kết nối đến csdl: DbChat
    TaiKhoan tk=new TaiKhoan();
    boolean kt=tk.ktDangNhap(un,pwd);
    if(kt==true)//Nếu đăng nhập đúng
    { CoSo.TenDn =un;//Lưu lại TenDangNhap
      String ip= InetAddress.getLocalHost().getHostAddress();//Lay Ip của Localhost
      tk.LuuIp(ip,un);
      tk.DangKetNoi("true",un);
      this.dispose();//hủy Frame: FrmDangNhap
      FrmClient f=new FrmClient();
      f.setVisible(true);//Mở form: frmClient
    }
    else
        JOptionPane.showMessageDialog(this,"Dang nhap sai") ;
} catch(Exception loi) {
    JOptionPane.showMessageDialog(this,loi.getMessage()) ;
}
}
```

- Sinh viên tự lập trình trên sự kiện actionPerformed của butDangKy để mở Frame FrmDangKy

4. Mở Frame FrmDangKy thiết kế giao diện như sau:



- Lập trình trên sự kiện actionPerformed của butDangKy để người dùng đăng ký 1 tài khoản mới

```
void butDangKy_actionPerformed(ActionEvent e) {
    try{
        String tenDn=txtdangnhap.getText();
        String matKhau=txtmatkhau.getText();
        String hoTen=txthoten.getText() ;
        if(txtnhaplaimatkhau.getText().equals(matKhau)==false)
        {
            JOptionPane.showMessageDialog(this,"mat khau va nhap lai mat khau k0 dung");
            return;
        }
        TaiKhoan tk= new TaiKhoan();
        if (tk.TaoTaiKhoan(tenDn,matKhau,hoTen)==1)
            JOptionPane.showMessageDialog(this,"Da tao tai khoan" );
        else
            JOptionPane.showMessageDialog(this,"Tai khoan nay da co" );
    } catch(Exception loi)
    {
        JOptionPane.showMessageDialog(this,loi.getMessage() );
    }
}
```

Sinh viên viết thêm mã để kiểm tra tên tài khoản này đã có hay chưa, nếu chưa có thì thêm vào 1 tài khoản.

5. Tạo lớp GoiNhan để người dùng gửi đi và nhận về 1 gói tin

```
import java.net.*;
import java.io.*;
public class GoiNhan {
    DatagramSocket Sserver ;
    public void MoCong(int cong)//Mở ra 1 cổng
    {
        try{
            Sserver=new DatagramSocket(cong );//Tạo ra 1 DatagramSocket và mở cổng
        } catch(Exception tb1)
        {
            System.out.print("Khong mo cong duoc") ;
        }
    }
}
```

```

    }
}
//Gửi đến máy có địa chỉ Ip 1 chuỗi tb, gửi vào cổng: cong
public void Goi(String tb,String Ip, int cong) {
try{
    DatagramSocket Sclient=new DatagramSocket();//Tạo ra 1 DatagramSocket để nhận gói tin
    InetAddress ip =InetAddress.getByName(Ip);//Lấy về thông tin của máy có địa chỉ Ip là: ip
    //Tạo ra 1 DatagramPacket để chứa gói tin và địa chỉ đích
    DatagramPacket dr=new DatagramPacket(tb.getBytes(),tb.length(),ip,cong);
    Sclient.send(dr);//Gửi gói tin đi
    Sclient.close() ;
    System.out.print("da goi");

} catch(Exception tb2)
{
    System.out.print("Khong goi duoc") ;
}
}
//Nhận về 1 gói tin theo dạng chuỗi
public String Nhan() {
try{
    byte[] buff= new byte[512];//Tạo ra 1 bộ đệm
    //Tạo ra 1 DatagramPacket để chứa gói tin nhận về
    DatagramPacket dr=new DatagramPacket(buff,buff.length);
    Sserver.receive(dr);//Nhận về 1 gói tin
    //Lấy dữ liệu về và đổi ra chuỗi
    String data=new String(dr.getData(),0,dr.getLength());
    System.out.print("da nhan");
    return data ;//Trả về chuỗi nhận được
} catch(Exception tb3)
{
    System.out.print("Khong nhan duoc") ;
    return null;
}
}
//Tương tự như phương thức Goi nhưng gửi đi 1 dãy các byte, phương thức này để gửi file
//Gửi đến máy có địa chỉ Ip là Ip 1 dãy các byte có độ dài là: dodai qua cổng: cong
public void GoiByte(byte tb[], int dodai,String Ip, int cong) {
try{
    DatagramSocket Sclient=new DatagramSocket();
    InetAddress ip =InetAddress.getByName(Ip);
    DatagramPacket dr=new DatagramPacket(tb,dodai,ip,cong);
    Sclient.send(dr);
    Sclient.close() ;
    System.out.print("da goi");
} catch(Exception tb2)
{
    System.out.print("Khong goi duoc") ;
}
}

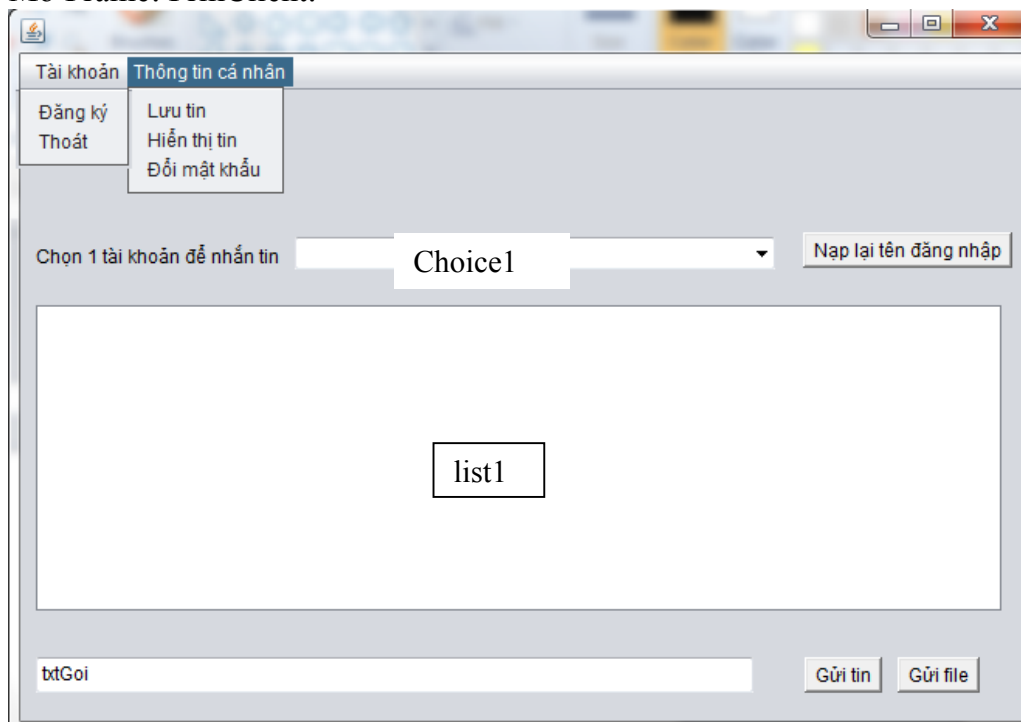
```

```

    }
}
//Nhập về 1 dãy các byte, phương thức này dùng để nhận file về
public byte[] NhanByte() {
    try{
        byte[] buff= new byte[512];
        DatagramPacket dr=new DatagramPacket(buff,buff.length);
        Sserver.receive(dr);
        byte[] DuLieuNhan=new byte[dr.getLength()];
        DuLieuNhan=dr.getData();
        CoSo.DoDaiGoiTinNhan =dr.getLength();//Lưu lại độ dài của gói tin vừa nhận
        return DuLieuNhan;//Trả về dãy các byte vừa nhận
    }catch(Exception tb3)
    {
        System.out.print("Khong nhan duoc") ;
        return null;
    }
}
}
}

```

6. Mở Frame: FrmClient:



- Khai báo các biến và thư viện dùng chung cho toàn Frame

```

import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.io.*;
import javax.swing.*;
import java.sql.*;
public class FrmClient extends JFrame {
    FileOutputStream f;//Khai báo file để lưu dữ liệu

```

```

int ktNhanfile=0;//Kiểm tra xem người dùng gửi thông báo hay file
GoiNhan gn = new GoiNhan();//Tạo ra lớp GoiNhan để gửi nhận thông báo hoặc file
Thread tdnhanfile; //Khai báo tuyến đoạn nhận file
...
}
- Tạo 2 lớp con: ThreadGoiTinNhan và ThreadNhanFile của lớp FrmClient để gửi nhận
  thông báo và nhận file.
public class FrmClient extends JFrame {
    FileOutputStream f;
    int ktNhanfile=0;
    GoiNhan gn = new GoiNhan();
    Thread tdnhanfile;
//Tạo ra 1 lớp tuyến đoạn để nhận thông báo và tên file
    public class ThreadGoiTinNhan extends Thread {
        public void run() {
            while (1 == 1) {//Lặp vô hạn để nhận thông báo
                try{
                    if(ktNhanfile==0)//Nếu gửi thông báo
                    {
                        System.out.print("Dang cho:");
                        String dulieu = gn.Nhan();//Chờ và nhận dữ liệu gửi về
                        //Qui ước: nếu ký tự đầu là khoảng trắng là gửi thông báo
                        //Nếu 4 ký tự đầu là: "/tf:" là gửi tên file cần chuyển
                        if (dulieu.substring(0, 4).equals("/tf:")) {//Nếu người dùng gửi tên file
                            list1.add("Dang nhan file: " + dulieu);//Hiển thị tên file cần gửi
                            String TenFileNhan = dulieu.substring(4, dulieu.length());//Lấy tên file
                            f = new FileOutputStream("d:\\\" + TenFileNhan);//Tạo ra 1 file có tên vừa nhận lưu vào ổ d
                            ktNhanfile = 1;//Bắt đầu nhận file
                            tdnhanfile=new Thread(new ThreadNhanFile());//Tạo ra tuyến đoạn để nhận file
                            tdnhanfile.start();//Bắt đầu tuyến đoạn -> bắt đầu nhận file
                        }
                    }
                    else//Nếu gửi thông báo
                    {
                        list1.add(dulieu); //Hiển thị dữ liệu ra list1
                        System.out.print("Da nhan:");
                    }
                }
                catch(Exception loi)
                {
                    System.out.print(loi);
                }
            }
        }
    }
}
//Kết thúc lớp ThreadGoiTinNhan
//Tạo ra 1 lớp tuyến đoạn để nhận file, mỗi lần nhận tối đa 512 byte
public class ThreadNhanFile extends Thread
{
    public void run()
    {
        try {
            while (true) {//Nếu còn dữ liệu để nhận

```

```

        if (ktNhanfile == 1) //Nếu đang nhận file
        {
            txtGoi.enable(false); //Không cho người dùng gõ thông báo
            byte[] buff = new byte[512]; //Tạo bộ đệm để nhận
            buff = gn.NhanByte(); //Nhận về 1 dãy các byte lưu vào buff
            //Chuyển 1 dãy các byte sang chuỗi
            String dl = new String(buff, 0, CoSo.DoDaiGoiTinNhan);
            if (dl.trim().equals("quit")) { //Nếu chuỗi nhận về là quit: kết thúc việc chuyển file
                f.close(); //Đóng và lưu file
                ktNhanfile = 0; //Chuyển sang chế độ nhận thông báo
                txtGoi.enable(true); //Cho phép gõ thông báo
                list1.add("Da nhan file luu o o d:"); //Hiển thị thông báo đã nhận file
                tdnhanfile.stop();
            }
            else
                f.write(buff, 0, CoSo.DoDaiGoiTinNhan); //Ghi dãy các byte từ buff vào file
        }
    }
}
catch (Exception tt) {
    System.out.print(tt.getMessage());
}
}
} //Kết thúc lớp ThreadNhanfile
- Lập trình trên sự kiện windowOpened của FrmClient để mở cổng 1261 kích hoạt tuyến
đoạn nhận thông báo và nạp các tài khoản đang sử dụng (DangKetNoi=true) vào choice
public void NapDuLieuVaoChoice() //Nạp các tài khoản đang sử dụng vào choice
{
    try {
        choice1.removeAll(); //Xóa tất cả dữ liệu trên choice
        TaiKhoan tk = new TaiKhoan();
        ResultSet rs = tk.LayTaiKhoanDangKetNoi(); //Lấy về các tài khoản đang sử dụng
        while(rs.next()) //Nạp vào choice
            choice1.add(rs.getString("TenDangNhap"));
        rs.close();
    } catch (Exception loi)
    {
        JOptionPane.showMessageDialog(this, loi.getMessage());
    }
}
private void formWindowOpened(java.awt.event.WindowEvent evt) {
    gn.MoCong(1261); //Mở cổng 1261
    Thread tdnhan = new Thread(new ThreadGoiTinNhan());
    tdnhan.start(); //Bắt đầu tuyến đoạn nhận thông báo
    NapDuLieuVaoChoice(); //Nạp dữ liệu vào choice
    this.setTitle("Xin chào: "+CoSo.TenDn); //Hiển thị tiêu đề của Frame
}
- Lập trình trên sự kiện actionPerformed của nút: Nạp Lại Ten Dang Nhap để nạp lại các tên
đăng nhập đang sử dụng: (Trong quá trình đang sử dụng chương trình có 1 số người dùng
đăng nhập thêm vào hệ thống, nút này có tác dụng làm cập nhật lại)

```



```

private void button1ActionPerformed(java.awt.event.ActionEvent evt) {
    NapDuLieuVaoChoice();
}

```

- Lập trình trên sự kiện actionPerformed của nút Goi để gửi 1 thông báo:
 - o Lấy về tên tài khoản trên choice1 => Ip của máy cần gửi
 - o Thông báo gửi lấy từ TxtGoi

```

void butGoi_actionPerformed(ActionEvent e) {
    TaiKhoan tk=new TaiKhoan();
    //Nếu tài khoản này đang dùng thì gửi thông báo
    if(tk.LayTrangThaiKetNoi(choice1.getSelectedItemAt() ).equals("1"))
    { //Dựa vào tên đăng nhập để lấy Ip của máy cần gửi đến
        String IpMayGoiDen=tk.LayIp(choice1.getSelectedItemAt() );
        //Ký tự đầu tiên của thông báo là khoảng trắng, tiếp theo là tên tài khoản và thông báo
        String dulieugoi=" "+CoSo.TenDn+"("+tk.LayIp(CoSo.TenDn ) +")" +txtGoi.getText();
        gn.Goi(dulieugoi,IpMayGoiDen,1261); //Gửi dữ liệu
    }
    else
        JOptionPane.showMessageDialog(this,choice1.getSelectedItemAt()+"Da log out" );
}

```

Chú ý:

- Để thử chạy trên 1 máy thì mở cổng 1261, và gửi dữ liệu qua 1260. Chạy frmClient để đăng nhập vào tài khoản 1. Ngược lại: mở cổng 1260 và gửi dữ liệu trên 1261. Chạy frmClient để đăng nhập vào tài khoản 2. Chat trên 2 tài khoản này
- Nếu chạy trên 2 máy thì mở cổng 1261 và gửi dữ liệu trên cổng 1261

Lập trình trên sự kiện actionPerformed của nút Goi File để gửi đi 1 file:

- o Mở cửa sổ để chọn file cần gửi
- o Lần lượt đọc file (mỗi lần đọc tối đa 512 byte), và gửi đi dữ liệu vừa đọc

```

void butGoiFile_actionPerformed(ActionEvent e) {
    try{
        //Mở cửa sổ để chọn file
        JFileChooser chooser = new JFileChooser();
        chooser.showOpenDialog(this); //Hiện thị cửa sổ

        // chooser.getSelectedFile().getPath(): lấy đường dẫn và tên file vừa chọn
        // chooser.getSelectedFile().getName(); //Lấy tên file vừa chọn
        TaiKhoan tk=new TaiKhoan();
        //Lấy Ip của máy cần gửi
        String IpMayGoiDen=tk.LayIp(choice1.getSelectedItemAt() );
        //Mở file vừa chọn để đọc
        FileInputStream f=new FileInputStream(chooser.getSelectedFile().getPath());
        byte[] buff=new byte[512]; //Tạo bộ đệm để chứa dữ liệu
        String tenfile="/tf:"+ chooser.getSelectedFile().getName();
        //Gửi tên file đi
        gn.Goi(tenfile,IpMayGoiDen,1261);
        Thread.sleep(2); //Dừng chương trình lại 2/1000 giây
        int sb=f.read(buff); //Đọc file tối đa 512 byte, sb; chứa số byte đọc được
        while(sb>0) //Duyệt file
        { gn.GoiByte(buff,sb,IpMayGoiDen,1261); //Gửi dữ liệu
          Thread.sleep(2);
        }
    }
}

```

```

        sb=f.read(buff,0,buff.length );//Đọc tiếp dữ liệu
    }while(sb>0);
    String st=new String("quit");//Kết thúc việc gửi
    gn.Goi("quit",IpMayGoiDen,1261);//Gửi thông báo đã chuyển xong file
    f.close() ;
    System.out.print("Da goi");

    }catch(Exception tt)  { System.out.print(tt) ; }
}

```

7. Yêu cầu bổ sung

- Đối với bài trên khi người dùng gửi file thì không gửi được thông báo, sửa lại thuật toán để vừa gửi file vừa gửi thông báo
- Lập trình trên sự kiện **actionPerformed** của menu Lưu File để lưu toàn bộ nội dung trên list1 vào bảng TinNhan của CSDL: ChatDb với ngày trường NgayNhan là ngày hiện tại
- Lập trình FrmHienThiTin để hiển thị và tìm kiếm theo ngày các mẫu tin trên bảng TinNhan
- Lập trình trên FrmDoiMatKhau để người dùng đổi lại mật khẩu