

**Bộ Công Thương**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HCM**

**----🙣🕮🙡----**



**ĐỒ ÁN**

**Môn: Phát Triển Ứng Dụng**

**Đề Tài: Xây dựng hệ thống chat room**



***Giáo Viên Hướng Dẫn:***

**Huỳnh Thái Học**

|  |  |
| --- | --- |
| **Sinh Viên Thực Hiện** | |
| **Nguyễn Công Luật** | **15076361** |
| **Mai Quốc Khánh** | **15074761** |
| **Nguyễn Nghĩa Minh Nam** | **15071451** |

**Link GitHub: https://github.com/luat4568/Topic1\_Ca1\_NguyenCongLuat**

**MỤC LỤC**

**Mục 1:** TỔNG QUAN…………………………………………………………………3

**1.1:** Tên đề tài và nhiệm vụ……………………………………………………………3

**1.2:** Tiến độ thực hiện của tất cả các tuần……………………………………………..3

**Mục 2:** CHI TIẾT XÂY DỰNG HỆ THỐNG CHATROOM………………………...6

**2.1:** Giao diện chức năng của hệ thống Chatroom…………………………………….6

**2.2:** Code của hệ thống Chatroom……………………………………………………..9

**2.2.1:** Client……………………………………………………………………………9

**2.2.2:** ClientThread…………………………………………………………………...15

**2.2.3:** Server…………………………………………………………………………..15

**2.2.4:** ServerThread…………………………………………………………………..18

**Mục 3:** KẾT LUẬN………………………………………………………………….21

**Mục 4:** Tài Liệu Tham Khảo……………………....…………………....…....………22

**Mục 1: TỔNG QUAN**

**1.1 ) Tên đề Tài và nhiệm vụ**

Đề tài: Xây dựng chương trình Chat Room theo mô hình Client-sever

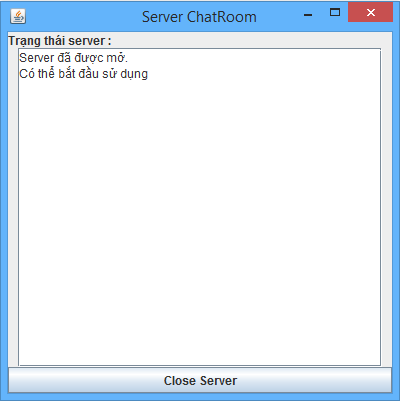
Nhiệm vụ: hoàn thành xây dựng chương trình Chat Room theo mô hình Client-Sever. Đảm bảo cho người dùng có thể dễ dàng sử dụng với giao diện đã thiết đơn giản, gần gũi. Cải tiến thêm các chức năng cho người dùng nếu có thể.

**1.2 ) Tiến độ thực hiện của tất cả các tuần.**

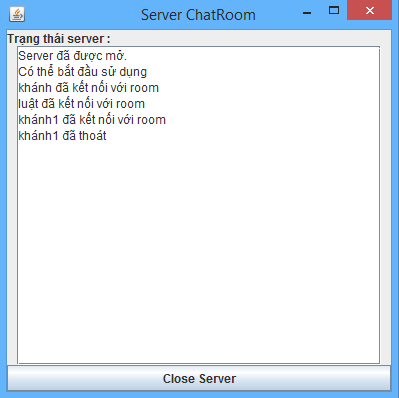
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thời Gian | Người thực hiện | Công việc thực hiện | Khó Khăn | Tiến độ |
| Ngày 1 (15/8/2017) | Thành viên nhóm | Tìm hiểu ý nghĩa các vấn đề sau:  -TCP/IP  -Socket:  +Khái quát về socket  +Cơ chế socket  +Mô hình truyền tin socket  +Một số hàm cơ bản trong  socket  -Thread  -Mô hình Client-Sever | Khái niệm mới nên vẫn chưa hiểu hết được. Nhất là về mô hình client-sever. | Tiến độ :100%  Hoàn thành việc nghiên cứu tài liệu. |
| Ngày 2 (22/8/2017) | Thành viên nhóm | Tìm hiểu hướng đối tượng trong JAVA:  +Học và thực hành các bài tập Java cơ bản trên môi trườn Eclips hoặc Netbean.  +Đọc tài liệu tìm hiểu được cách để viết chương trình chat room bằng ngôn ngữ Java  +Thực hành các dạng bài tập:  -Viết chương trình giải phương trình bậc 2  -Viết chương trình quản lý CD. | Một số thành viên chưa học qua lập trình Java nên tốn khoản thời gian khá nhiều cho việc đọc và học Ngôn ngữ Java. | Tiến độ: 100%  Hoàn thành được hơn một nữa các bài tập thực hành đã đề ra. |
| Ngày 3 (29/8/2017) | -Nguyễn Công Luật(1)  -Mai Quốc Khánh(2)  -Nguyễn Nghĩa Minh Nam(3) | -Tiếp tục thực hành lập trình Java.  -Hoàn tất các bài của tuần trước  -Tham khảo, tìm hiểu các thực hiện chương trình chat client-sever cơ bản.  (3)  -Viết chương trình chat client-sever cơ bản(chưa có giao diện)  (1)(2) | -Khó khăn trong việc tiếp xúc các hàm, thuộc tính để viết các kết nối của chat. | Tiến độ: 100%  Hoàn thành các bài tập Java cơ bản còn lại.  Bắt đầu thiết kế chương trình chat cilent-sever cơ bản. |
| Ngày 4 (5/9/2017) | -Nguyễn Công Luật(1)  -Mai Quốc Khánh(2)  -Nguyễn Nghĩa Minh Nam(3) | -Tiếp tục thiết kế chương trình chat client-sever cơ bản.  (1),(2).  -Học và ôn tập cách thiết kế giao diện(1,2,3):  +Học cách sử dụng các cấu trúc câu, thuộc tính.  +Thực hành các bài giao diện mẫu đơn giản.  -Phân tích cách thiết kế giao diện chat room(1):  +Thiết kế mẫu giao diện chat room bằng hình ảnh | Phần thiết kế giao diện gặp phải khá nhiều thuộc tính mới, nên việc làm quen củng như thực hành gặp nhìu khó khăn, tốn nhiều thời gian | Tiến độ: 100%  -Phần thiết kế giao diện mẫu bằng hình ảnh hoàn thành.  -Phần thiết kế chương trình chat , và học cách thiết kế giao diện hoàn thành. |
| Ngày 5 (12/9/2017) | -Nguyễn Công Luật(1)  -Mai Quốc Khánh(2)  -Nguyễn Nghĩa Minh Nam(3) | -Thiết kế giao diện chat.(1,2)  -Nghiên cứu thêm tài liệu về giao diện hổ trợ cho 2 thành viên còn lại(3). | Thiết kế theo mẫu đã vẽ sẵn nhưng vẫn chưa đc hoàn thiện. | Tiến độ: hoàn thành 100% |
| Ngày 6 (19/9/2017) | -Nguyễn Công Luật(1)  -Mai Quốc Khánh(2)  -Nguyễn Nghĩa Minh Nam(3) | Tiếp tục thiết diện hoàn chỉnh(1,2)  -Tham khảo tài liệu liên kết giao diện, làm liên kết giao diện chat với client-sever(3) | Phần giao diện xác định được bố cục, nhưng khi chạy thì các button bị lệch.  Phần tài liệu liên kết giao diện chat phải tham khảo các nguồn tài liệu nước ngoài nên gặp phải một chút trở ngại. | Tiến độ: 100%  -Phần thiết kế giao diện hoàn thành.  -Phần liên kết hoàn thành. |
| Ngày 7 (26/9/2017) | -Nguyễn Công Luật(1)  -Mai Quốc Khánh(2)  -Nguyễn Nghĩa Minh Nam(3) | Tiếp tục code phần kết nối client-sever(3) | -Khó khăn trong phần viết luồng(thread) để giúp chat được nhiều người. | Tiến độ: 100%  -Hoàn thành phần kết nối client-sever |
| Ngày 8 (3/10/2017) | -Nguyễn Công Luật(1)  -Mai Quốc Khánh(2)  -Nguyễn Nghĩa Minh Nam(3) | Hợp phần code thiết kế với code kết nối client-sever lại, điều chỉnh để có thể chạy được giao diện hoàn chỉnh có thể kết nối các máy với nhau. | Phần hợp code là phần quan trọng và khó khăn nhất, nên các thành viên tốn nhiều thời gian tìm hiểu vấn đề lỗi và fix lỗi. | Tiến độ: 100%  -Hoàn thành việc hợp code thiết kế với code kết nối. |
| Ngày 9 (10/10/2017) | -Nguyễn Công Luật(1)  -Mai Quốc Khánh(2)  -Nguyễn Nghĩa Minh Nam(3) | Kiểm tra chương trình chat room, ghi nhận các lỗi sự cố xảy ra và tìm các khắc phục. | Không có khó khăn. | Tiến độ: 100%  -Hoàn thành việc kiểm tra và chạy chương trình thành công |
| Ngày 10 (17/10/2017) | -Nguyễn Công Luật(1)  -Mai Quốc Khánh(2)  -Nguyễn Nghĩa Minh Nam(3) | Hoàn tất việc xây dựng chương trình chat room. Ngoài ra có thể đề xuất xây dựng thêm:  -Gửi file, chat icon | Chưa xây dựng thêm được chức năng mới (gửi file, chat icon). | Tiến độ: 100%  -Hoàn thành công việc đề ra, hướng tới phát triển thêm các chức năng khác. |

**Mục 2: CHI TIẾT XÂY DỰNG HỆ THỐNG CHATROOM**

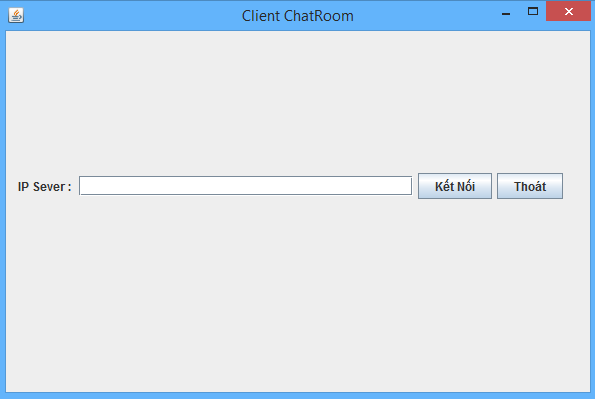
**2.1: Giao diện chức năng của hệ thống Chatroom**

****

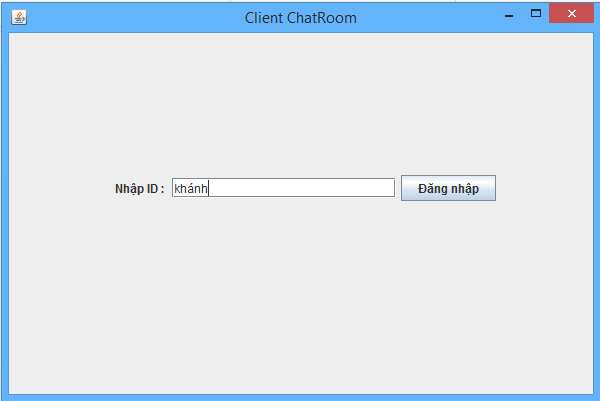
*Hình 1: Giao diện thông báo chạy server (Bước 1)*

**

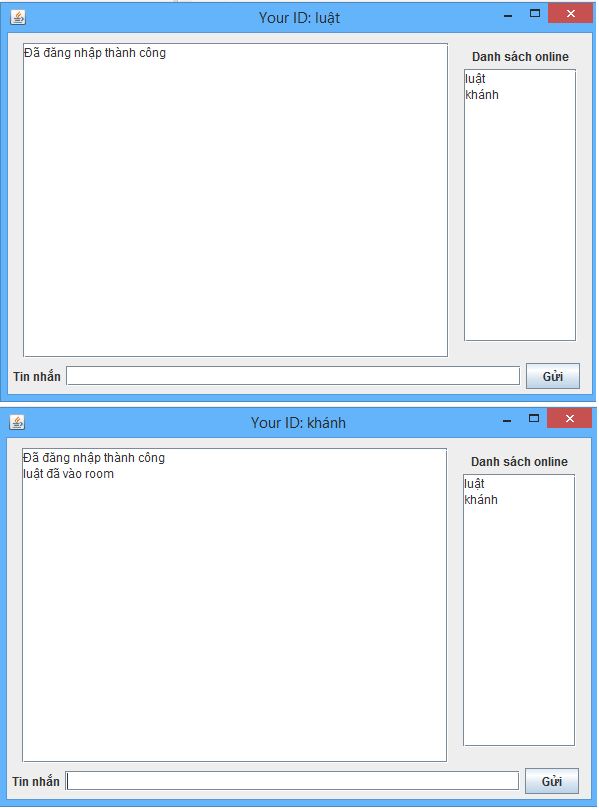
*Hình 2: Giao diện thông báo của server*

****

*Hình 3: Giao diện nhập địa chỉ IP server (Bước 2)*

****

*Hình 4: Giao diện nhập Tên (ID) tham gia chat (Bước 3)*

****

*Hình 5: Giao diện nhắn tin (Bước 4)*

**2.2: Code của hệ thống Chatroom**

**2.2.1: Client**

public class Client extends JFrame implements ActionListener, KeyListener {

private JButton send, exit, login, Connect, Thoat;

private JPanel p\_dangnhap, p\_chat, p\_ketnoi;

private JTextField nick2, nick1, message, IP;

private JTextArea msg, online;

private Socket client;

private ClientThread dataStream;

private DataOutputStream dos;

private DataInputStream dis;

private JLabel lblIpsever;

public Client() {

super("Client ChatRoom");

setDefaultCloseOperation(JFrame.*DO\_NOTHING\_ON\_CLOSE*);

this.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

exit();

}

});

setSize(600, 400);

addItem();

setVisible(true);

}

// -----[ Tạo giao diện ]--------//

private void addItem() {

setLayout(new BorderLayout());

exit = new JButton("Thoát");

exit.addActionListener(this);

send = new JButton("Gửi");

send.addActionListener(this);

login = new JButton("Đăng nhập");

login.addActionListener(this);

//Jpanel chát

p\_chat = new JPanel();

p\_chat.setLayout(new BorderLayout());

JPanel p1 = new JPanel();

p1.setLayout(new FlowLayout(FlowLayout.*LEFT*));

nick2 = new JTextField();

JPanel p2 = new JPanel();

p2.setLayout(new BorderLayout());

JPanel p22 = new JPanel();

p22.setLayout(new FlowLayout(FlowLayout.*CENTER*));

p22.add(new JLabel("Danh sách online"));

p2.add(p22, BorderLayout.*NORTH*);

online = new JTextArea(10, 10);

online.setEditable(false);

p2.add(new JScrollPane(online), BorderLayout.*CENTER*);

p2.add(new JLabel(" "), BorderLayout.*SOUTH*);

p2.add(new JLabel(" "), BorderLayout.*EAST*);

p2.add(new JLabel(" "), BorderLayout.*WEST*);

msg = new JTextArea(1, 1);

msg.setEditable(false);

JPanel p3 = new JPanel();

p3.setLayout(new FlowLayout(FlowLayout.*LEFT*));

p3.add(new JLabel("Tin nhắn"));

message = new JTextField(41);

p3.add(message);

p3.add(send);

p\_chat.add(new JScrollPane(msg), BorderLayout.*CENTER*);

p\_chat.add(p1,BorderLayout.*NORTH*);

p\_chat.add(p2, BorderLayout.*EAST*);

p\_chat.add(p3, BorderLayout.*SOUTH*);

p\_chat.add(new JLabel(" "), BorderLayout.*WEST*);

p\_chat.setVisible(false);

add(p\_chat, BorderLayout.*CENTER*);

// JPanel đăng nhập nick

p\_dangnhap = new JPanel();

javax.swing.Box b =javax.swing.Box.*createVerticalBox*();

p\_dangnhap.add(b);

b.add(javax.swing.Box.*createVerticalStrut*(300));

p\_dangnhap.setLayout(new FlowLayout(FlowLayout.*CENTER*));

p\_dangnhap.add(new JLabel("Nhập ID : "));

nick1 = new JTextField(20);

p\_dangnhap.add(nick1);

p\_dangnhap.add(login);

p\_dangnhap.setVisible(false);

add(p\_dangnhap, BorderLayout.*NORTH*);

//JPanel kết nối IP

p\_ketnoi = new JPanel();

javax.swing.Box b1 =javax.swing.Box.*createVerticalBox*();

p\_ketnoi.add(b1);

b1.add(javax.swing.Box.*createVerticalStrut*(300));

p\_ketnoi.add(lblIpsever = new JLabel("IP Sever : "));

p\_ketnoi.add(IP = new JTextField(30));

p\_ketnoi.add(Connect = new JButton("Kết Nối"));

p\_ketnoi.add(Thoat = new JButton("Thoát"));

Thoat.addActionListener(this);

Connect.addActionListener(this);

add(p\_ketnoi, BorderLayout.*LINE\_START*);

//Gán sự kiện phím Enter

IP.addKeyListener(this);

nick1.addKeyListener(this);

message.addKeyListener(this);

}

// ---------[ Socket ]-----------//

private void go(String IP) {

try {

client = new Socket(IP, 2207);

dos = new DataOutputStream(client.getOutputStream());

dis = new DataInputStream(client.getInputStream());

} catch (Exception e) {

JOptionPane.*showMessageDialog*(this, "Lỗi kết nối, vui lòng kiểm tra lại địa chỉ IP hoặc Server.", "Message Dialog",

JOptionPane.*WARNING\_MESSAGE*);

System.*exit*(0);

}

}

public static void main(String[] args) {

new Client();

}

private void sendMSG(String data) {

try {

dos.writeUTF(data);

dos.flush();

} catch (IOException e) {

e.printStackTrace();

}

}

private String getMSG() {

String data = null;

try {

data = dis.readUTF();

} catch (IOException e) {

e.printStackTrace();

}

return data;

}

public void getMSG(String msg1, String msg2) {

int stt = Integer.*parseInt*(msg1);

switch (stt) {

// tin nhắn của những người khác

case 3:

this.msg.append(msg2);

break;

// update danh sách online

case 4:

this.online.setText(msg2);

break;

// server đóng cửa

case 5:

dataStream.stopThread();

exit();

break;

// bổ sung sau

default:

break;

}

}

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

private void checkSend(String msg) {

if (msg.compareTo("\n") != 0) {

this.msg.append("Tôi : " + msg);

sendMSG("1");

sendMSG(msg);

}

}

private boolean checkLogin(String nick2) {

if (nick2.compareTo("") == 0)

return false;

else if (nick2.compareTo("0") == 0) {

return false;

} else {

sendMSG(nick2);

int sst = Integer.*parseInt*(getMSG());

if (sst == 0)

return false;

else

return true;

}

}

private void exit() {

try {

sendMSG("0");

dos.close();

dis.close();

client.close();

} catch (IOException e1) {

e1.printStackTrace();

}

System.*exit*(0);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == exit) {

dataStream.stopThread();

exit();

} else if (e.getSource() == send) {

checkSend(message.getText() + "\n");

message.setText("");

} else if (e.getSource() == login) {

if (checkLogin(nick1.getText())) {

p\_chat.setVisible(true);

p\_dangnhap.setVisible(false);

nick2.setText(nick1.getText());

nick2.setEditable(false);

this.setTitle("Your ID: "+nick1.getText());

msg.append("Bạn đã đăng nhập thành công\n");

dataStream = new ClientThread(this, this.dis);

} else {

JOptionPane.*showMessageDialog*(this, "ID này đang được sử dụng, vui lòng nhập nick2 khác .",

"Message Dialog", JOptionPane.*WARNING\_MESSAGE*);

}

} else if (e.getSource() == Connect) {

go(IP.getText());

p\_dangnhap.setVisible(true);

p\_ketnoi.setVisible(false);

} else if (e.getSource() == Thoat) {

System.*exit*(0);

}

}

@Override

public void keyPressed(KeyEvent e) {

// TODO Auto-generated method stub

if (e.getKeyCode() == KeyEvent.*VK\_ENTER* && e.getSource().equals(IP)) {

go(IP.getText());

p\_dangnhap.setVisible(true);

p\_ketnoi.setVisible(false);

}else if(e.getKeyCode() == KeyEvent.*VK\_ENTER* && e.getSource().equals(nick1)){

System.*out*.println("enter");

if (checkLogin(nick1.getText())) {

p\_chat.setVisible(true);

p\_dangnhap.setVisible(false);

nick2.setText(nick1.getText());

nick2.setEditable(false);

this.setTitle("Your ID: "+nick1.getText());

msg.append("Đã đăng nhập thành công\n");

dataStream = new ClientThread(this, this.dis);

} else {

JOptionPane.*showMessageDialog*(this, "Đã tồn tại níck này trong room, bạn vui lòng nhập lại.",

"Message Dialog", JOptionPane.*WARNING\_MESSAGE*);

}

}else if(e.getKeyCode() == KeyEvent.*VK\_ENTER* && e.getSource().equals(message)){

checkSend(message.getText() + "\n");

message.setText("");

}

}

@Override

public void keyReleased(KeyEvent e) {

// TODO Auto-generated method stub

}

@Override

public void keyTyped(KeyEvent arg0) {

// TODO Auto-generated method stub

}

}

**2.2.2: ClientThread**

public class ClientThread extends Thread {

private boolean run;

private DataInputStream dis;

private Client client;

public ClientThread(Client client,DataInputStream dis){

run=true;

this.client=client;

this.dis=dis;

this.start();

}

public void run(){

String msg1,msg2;

while(run){

try {

msg1=dis.readUTF();

msg2=dis.readUTF();

client.getMSG(msg1,msg2);

} catch (IOException e) {

e.printStackTrace();

}

}

try {

dis.close();

} catch (IOException e) {

e.printStackTrace();

}

}

public void stopThread(){

this.run=false;

}

}

**2.2.3: Server**

public class Server extends JFrame implements ActionListener{

private JButton close;

public JTextArea user;

private ServerSocket server;

public Hashtable<String, ServerThread> listUser;

public Server(){

super("Server ChatRoom");

setDefaultCloseOperation(JFrame.*DO\_NOTHING\_ON\_CLOSE*);

this.addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

try {

//gởi tin nhắn tới tất cả client

server.close();

System.*exit*(0);

} catch (IOException e1) {

e1.printStackTrace();

}

}

});

setSize(400, 400);

addItem();

setVisible(true);

}

private void addItem() {

setLayout(new BorderLayout());

add(new JLabel("Trạng thái server : \n"),BorderLayout.*NORTH*);

add(new JPanel(),BorderLayout.*EAST*);

add(new JPanel(),BorderLayout.*WEST*);

user = new JTextArea(10,20);

user.setEditable(false);

add(new JScrollPane(user),BorderLayout.*CENTER*);

close = new JButton("Close Server");

close.addActionListener(this);

add(close,BorderLayout.*SOUTH*);

user.append("Server đã được mở.\n");

}

private void go(){

try {

listUser = new Hashtable<String, ServerThread>();

server = new ServerSocket(2207);

user.append("Có thể bắt đầu sử dụng\n");

while(true){

Socket client = server.accept();

new ServerThread(this,client);

}

} catch (IOException e) {

user.append("Không thể khởi động Server\n");

}

}

public static void main(String[] args) {

new Server().go();

}

public void actionPerformed(ActionEvent e) {

try {

server.close();

} catch (IOException e1) {

user.append("Không thể dừng Server\n");

}

System.*exit*(0);

}

public void sendAll(String from, String msg){

Enumeration e = listUser.keys();

String name=null;

while(e. hasMoreElements()){

name=(String) e.nextElement();

if(name.compareTo(from)!=0) listUser.get(name).sendMSG("3",msg);

}

}

public void sendAllUpdate(String from){

Enumeration e = listUser.keys();

String name=null;

while(e. hasMoreElements()){

name=(String) e.nextElement();

if(name.compareTo(from)!=0) listUser.get(name).sendMSG("4",getAllName());

}

}

public String getAllName(){

Enumeration e = listUser.keys();

String name="";

while(e. hasMoreElements()){

name+=(String) e.nextElement()+"\n";

}

return name;

}}

**2.2.4: ServerThread**

public class ServerThread extends Thread {

public Socket client;

public Server server;

private String nickName;

private DataOutputStream dos;

private DataInputStream dis;

private boolean run;

public ServerThread(Server server, Socket client){

try {

this.server=server;

this.client=client;

dos= new DataOutputStream(client.getOutputStream());

dis= new DataInputStream(client.getInputStream());

run=true;

this.start();

} catch (IOException e) {

e.printStackTrace();

}

}

public void run(){

// xữ lý đăng nhập

String msg=null;

while(run){

nickName=getMSG();

if(nickName.compareTo("0")==0){

logout();

}

else {

if(checkNick(nickName)){

sendMSG("0");

}

else{

server.user.append(nickName+" đã kết nối với room\n");

server.sendAll(nickName,nickName+" đã vào room\n");

server.listUser.put(nickName, this);

server.sendAllUpdate(nickName);

sendMSG("1");

diplayAllUser();

while(run){

int stt = Integer.*parseInt*(getMSG());

switch(stt){

case 0:

run=false;

server.listUser.remove(this.nickName);

exit();

break;

case 1:

msg = getMSG();

server.sendAll(nickName,nickName+" : "+msg);

break;

}

}

}

}

}

}

private void logout() {

try {

dos.close();

dis.close();

client.close();

} catch (IOException e) {

e.printStackTrace();

}

}

private void exit(){

try {

server.sendAllUpdate(nickName);

dos.close();

dis.close();

client.close();

server.user.append(nickName+" đã thoát\n");

server.sendAll(nickName,nickName+" đã thoát\n");

} catch (IOException e) {

e.printStackTrace();

}

}

private boolean checkNick(String nick){

return server.listUser.containsKey(nick);

}

private void sendMSG(String data){

try {

dos.writeUTF(data);

dos.flush();

} catch (IOException e) {

e.printStackTrace();

}

}

public void sendMSG(String msg1,String msg2){

sendMSG(msg1);

sendMSG(msg2);

}

private String getMSG(){

String data=null;

try {

data=dis.readUTF();

} catch (IOException e) {

e.printStackTrace();

}

return data;

}

private void diplayAllUser(){

String name = server.getAllName();

sendMSG("4");

sendMSG(name);

}

}

**Mục 3: KẾT LUẬN**

● Công việc làm được:

* + - Xây dựng được mô hình Chatroom cơ bản như yêu cầu đề ra.
    - Mô phỏng được mô hình Client Server trong mạng Lan.
    - Biết được cách dùng giao thức TCP/IP trong truyền dữ liệu.

● Những vấn đề còn tồn tại:

* + - Chương trình chat còn đơn giản và sơ sài.
    - Chưa thể dung để gửi được file, cũng như thêm tiện ích khác cho giao diện chat.

● Định hướng sau này:

* + - Cải tiến chương trình có thể gửi file qua lại giữa Client - Server.
    - Phát triển tính năng password đối với mỗi tài khoản chat.

**Mục 4: Tài Liệu Tham Khảo**

**https://sites.google.com/site/huynhhoc/distributed-object**

[**http://www.toancau.vn/vietnam/4/5/32/giao-thuc-tcpip-la-gi**](http://www.toancau.vn/vietnam/4/5/32/giao-thuc-tcpip-la-gi)

[**http://hus.edu.vn/thcs/bai10.html**](http://hus.edu.vn/thcs/bai10.html)

[**https://voer.edu.vn/c/mo-hinh-client-server/761b0302/eedbe7c9**](https://voer.edu.vn/c/mo-hinh-client-server/761b0302/eedbe7c9)

[**http://www.cit.ctu.edu.vn/~dtnghi/netprog/4a-Socket.pdf**](http://www.cit.ctu.edu.vn/~dtnghi/netprog/4a-Socket.pdf)

[**http://www.cit.ctu.edu.vn/~dtnghi/netprog/4a-Socket.pdf**](http://www.cit.ctu.edu.vn/~dtnghi/netprog/4a-Socket.pdf)