



MÔN: PHÁT TRIỂN ỨNG DỤNG CÓ ĐỒ ÁN



Đồ Án

**ĐỀ TÀI: CONTROLLING MACHINES REMOTELY
WITH JAVA**



GVHD: Huỳnh Thái Học – Đỗ Hà Phương

NHÓM: 3T

Link GitHub: <https://goo.gl/un9sPW>

TP.HCM-Tháng 08 Năm 2017

MỤC LỤC

1. Tổng Quan.	1
1.1 Giới thiệu đề tài. Topic 6.	1
1.2 Tiến độ thực hiện.	2
2. Cơ sở lý thuyết – Cài đặt.....	4
2.1 Cơ sở lý thuyết.	4
2.1.1 Giới thiệu mô hình Client/Server.	4
2.1.1.1 Cách thức hoạt động.....	4
2.1.2 Lập trình với giao thức TCP Socket.	4
2.2 Phương pháp triển khai.....	5
2.3 Thiết kế ứng dụng.	6
2.3.1 Chương trình trên Server (RemoteServer).....	6
2.3.2 Chương trình trên Client (RemoteClient).	9
2.4 Kết quả khi chạy chương trình.....	11
2.4.1 Phía Server.	11
2.4.2 Phía Client.	12
3. Kết Luận.	14
3.1 Tự đánh giá nhận xét.	14
3.2 Chức năng làm thêm.	14
3.3 Hướng phát triển trong tương lai.	14
Tài liệu tham khảo.....	15

DANH MỤC CÁC HÌNH VẼ

Hình 1. Mô hình Client/Server.....	4
Hình 2. Tầng Transport với TCP/IP	5

1. Tổng Quan.

1.1 Giới thiệu đề tài. Topic 6.

Xây dựng chương trình “Controlling Machines Remotely with Java” (Điều khiển từ xa).

Hiện nay, Công Nghệ Thông Tin (CNTT) đang phát triển mạnh mẽ và là ngành công nghiệp mũi nhọn ở nhiều quốc gia trên thế giới, trong đó có Việt Nam. Máy tính trở lên phổ biến, xuất hiện ngày càng nhiều ở các gia đình và trở thành một công cụ không thể thiếu của nhiều người.

Và với nhu cầu sử dụng máy tính hiện nay, con người không chỉ trực tiếp điều khiển trực tiếp trên chính máy tính, mà còn có nhu cầu điều khiển từ xa giống như các thiết bị điện tử khác (tivi, đầu đĩa, ...). Đó là một công cụ hỗ trợ đắc lực cho con người nhất là trong công việc giám sát hỗ trợ từ xa.

Trong đề tài này, chúng em thực hiện xây dựng một chương trình điều khiển từ xa, theo mô hình Client/Server, trong mạng LAN. Trong đó máy điều khiển là Server và máy bị điều khiển là Client. Sau khi kết nối thành công, máy Server sẽ có thể thực hiện các chức năng sau.

➤ **Chức năng yêu cầu.**

- Theo dõi, điều khiển màn hình máy Client.

➤ **Những chức năng mở rộng có thể làm thêm** (nếu hoàn thành xong chức năng yêu cầu).

- Chia sẻ màn hình cho máy Client.
- Chụp màn hình máy Client.
- Thực hiện lệnh Shutdown, Restart máy Client.

Ứng dụng sẽ được xây dựng bằng ngôn ngữ lập trình Java trên Eclipse.

1.2 Tiến độ thực hiện.

Tuần	Công việc	Thành viên	Ghi chú
Tuần 1 (15/8/2017)	<ul style="list-style-type: none"> Hình dung, mô tả ứng dụng sẽ làm. Lập kế hoạch cho 9 tuần tiếp theo. 	<ul style="list-style-type: none"> Nguyễn Văn Chơn Nguyễn Văn Chung Võ Thành Đạt 	- Hoàn thành (100%). Có file word
Tuần 2 (22/8/2017)	<ul style="list-style-type: none"> Tìm hiểu cách thiết lập mạng LAN trên VMWARE Link cài Vmware: https://goo.gl/Zdyr24 Link cài windown 7 trên Vmware: https://goo.gl/kx6kdd Link thiết lập mạng LAN giữa 2 máy: https://goo.gl/5cU7gJ 	<ul style="list-style-type: none"> Nguyễn Văn Chơn Nguyễn Văn Chung Võ Thành Đạt 	- Hoàn thành (100%)
	- Tiến hành xây dựng mạng LAN với 1 máy ảo và 1 máy thật	<ul style="list-style-type: none"> Nguyễn Văn Chung 	- Hoàn thành (100%)
Tuần 3 (29/8/2017)	<ul style="list-style-type: none"> Tìm hiểu về lập trình Socket trên java. (xem code ví dụ). Link 1: https://goo.gl/4WNaSq Link 2: https://goo.gl/oYRE9T Link 3: https://goo.gl/6a628L 	<ul style="list-style-type: none"> Nguyễn Văn Chơn Nguyễn Văn Chung Võ Thành Đạt 	- Hoàn thành (100%)
	- Làm theo ví dụ.	<ul style="list-style-type: none"> Võ Thành Đạt Nguyễn Văn Chơn 	- Hoàn thành (70%)

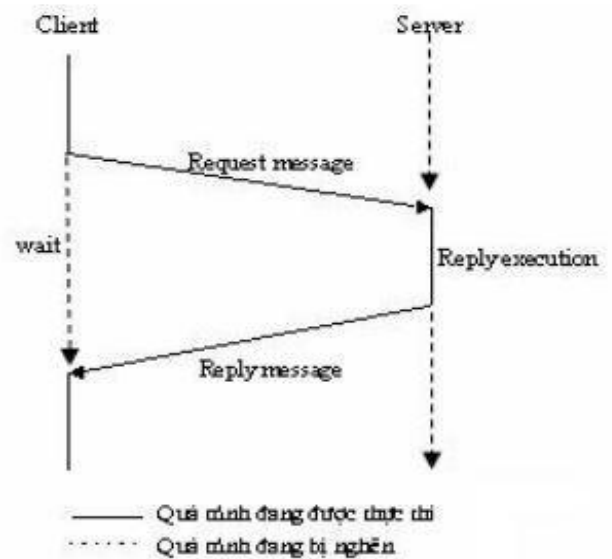
Tuần 4 (05/9/2017)	- Tiến hành kết nối giữa 2 máy Client/Server (Trên mạng LAN đã thiết lập ở trên). Bằng câu lệnh trong java.	- Nguyễn Văn Chơn - Nguyễn Văn Chung - Võ Thành Đạt	- Hoàn thành (100%). Nhưng bị lán sang tuần 5.
Tuần 5 (12/9/2017)	- Xem & hiểu cách theo dõi màn hình máy Client Link 1: https://goo.gl/oMNrnL Link 2: https://goo.gl/c8EaJL	- Nguyễn Văn Chơn - Nguyễn Văn Chung - Võ Thành Đạt	- Hoàn thành (100%)
Tuần 6 & 7 (19/9/2017 & 26/9/2017)	- Hiện thực theo dõi màn hình, điều khiển máy Client	- Nguyễn Văn Chơn - Nguyễn Văn Chung - Võ Thành Đạt	- Hoàn thành (100%)
Tuần 8 (03/10/2017)	- Thêm chức năng shutdown máy Client. - Truyền gửi thông điệp giữa Client và Server.	- Nguyễn Văn Chơn - Võ Thành Đạt	- Hoàn thành (100%)
Tuần 9 & 10 (10&17/10/2017)	- Test lại các chức năng - Hoàn thiện ứng dụng	- Nguyễn Văn Chung - Võ Thành Đạt	- Hoàn thành (100%)

2. Cơ sở lý thuyết – Cài đặt

2.1 Cơ sở lý thuyết.

2.1.1 Giới thiệu mô hình Client/Server.

Mô hình Client-Server được sử dụng trong các hệ phân tán và bao gồm hai thành phần riêng biệt là: Server đóng vai trò là người phục vụ, cung cấp chức năng và Client đóng vai trò là người tiêu thụ sử dụng chức năng đó. Client là bên chủ động tạo kết nối và gửi yêu cầu đến Server, trong khi Server thụ động lắng nghe và thực hiện yêu cầu đó.



Hình 1. Mô hình Client/Server

2.1.1.1 Cách thức hoạt động.

Hoạt động theo kiểu giao thức bất đối xứng.

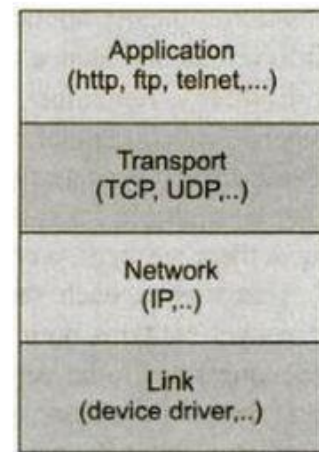
Thể hiện quan hệ một chiều giữa Client và một Server.

Các tiến trình Client và Server có thể chạy cùng một host hoặc các host khác nhau và là các đối tượng logic tách biệt, liên lạc với nhau qua mạng để cùng thực hiện một công việc. Client bắt đầu phiên hội thoại bằng cách yêu cầu dịch vụ, Server sẵn sàng chờ các yêu cầu từ Client gửi tới.

2.1.2 Lập trình với giao thức TCP Socket.

Socket là một phương pháp để thiết lập kết nối truyền thông giữa một chương trình yêu cầu dịch vụ (Client) và một chương trình cung cấp dịch vụ (Server) trên mạng LAN, WAN hay Internet và đôi lúc là giữa những quá trình ngay bên trong máy tính. Mỗi socket có thể được xem như là một điểm cuối trong một kết nối. Khi một socket được thiết lập phù hợp, hai máy tính có thể trao đổi dịch vụ dữ liệu.

Trong mô hình Client-Server, thông thường các chương trình chạy trên máy Client sẽ gửi yêu cầu tới một chương trình (thường được gọi là chương trình Server) đang chạy trên máy Server. Các gói yêu cầu này bao gồm các dịch vụ mạng được cung cấp bởi tầng giao vận (Transport layer) trong mô hình mạng, thường được gọi là TCP/IP (Transport Control Protocol/Internet Protocol). Tầng này bao gồm hai giao thức là TCP và UDP (User Datagram Protocol).



Hình 2. Tầng Transport với TCP/IP

Giao thức TCP là một trong các giao thức cốt lõi của giao thức TCP/IP. Sử dụng TCP, các máy được nối mạng với nhau có thể tạo các “kết nối” với nhau, mà qua đó chúng có thể trao đổi dữ liệu hoặc các gói tin. Giao thức này đảm bảo truyền dữ liệu tới nơi nhận một cách đáng tin cậy và đúng thứ tự.

Giao thức UDP cũng là một trong các giao thức cốt lõi của giao thức TCP/IP. Sử dụng UDP, chương trình trên mạng máy tính có thể gửi những dữ liệu ngắn được gọi là datagram tới máy khác. UDP không cung cấp sự tin cậy và thứ tự truyền nhận. Các gói dữ liệu có thể đến không đúng thứ tự hoặc bị mất mà không có thông báo nào. Tuy nhiên UDP nhanh và hiệu quả hơn đối với các mục tiêu như kích thước nhỏ và yêu cầu khắt khe về thời gian.

2.2 Phương pháp triển khai.

Dùng phương pháp lập trình mạng TCP Socket threaded Server để tạo kết nối giữa máy Client với máy Server và tạo các luồng Input/Output.

Dùng giao thức UDP để truyền dữ liệu giữa Client và Server.

Client sẽ kết nối tới Server thông qua địa chỉ IP của Server. Lúc này Server chấp nhận kết nối từ Client và thực hiện việc điều khiển.

Để cho phép nhiều Client có thể kết nối đến Server thì Server phải là chương trình đa nhiệm, đa luồng cho phép nhiều tiến trình, tiểu trình có thể chạy song song cùng một thời điểm. Mỗi tuyến (thread) đảm nhận việc liên lạc với Client. Nghĩa là khi có một Client kết nối đến, chương trình Server sinh ra một tuyến (thread) để điều khiển việc truyền thông với Client.

2.3 Thiết kế ứng dụng.

2.3.1 Chương trình trên Server (RemoteServer)

➤ Lớp ClientHandler.

Mỗi client kết nối tới, lớp này sẽ tạo ra một đối tượng (thread). Nó cho thấy khung màn hình của mỗi client.

Code: Tạo khung màn hình máy Client với kích thước tối đa.

```
public void drawGUI(){
    interFrame.setLayout(new BorderLayout());
    interFrame.getContentPane().add(cPanel,BorderLayout.CENTER);
    interFrame.setSize(100,100);
    desktop.add(interFrame);
    try {
        interFrame.setMaximum(true);
    } catch (PropertyVetoException ex) {
        ex.printStackTrace();
    }
}
```

➤ Lớp ClientCommandsSender.

Chờ lệnh từ máy chủ và gửi chúng đến máy Client. Gồm các lệnh di chuyển chuột, nhấn bàn phím,

Code: Nhận lệnh và gửi lệnh tới Client.

```
public void mouseMoved(MouseEvent e)
{
    double xScale = clientScreenDim.getWidth()/cPanel.getWidth();
    System.out.println("xScale: " + xScale);
    double yScale = clientScreenDim.getHeight()/cPanel.getHeight();
    System.out.println("yScale: " + yScale);
    System.out.println("Mouse Moved");
    writer.println(EnumCommands.MOVE_MOUSE.getAbbrev());
}
```

```
        writer.println((int)(e.getX() * xScale));
        writer.println((int)(e.getY() * yScale));
        writer.flush();
    }
    public void mouseClicked(MouseEvent e) {
    }
    public void mouseReleased(MouseEvent e) {
        System.out.println("Mouse Released");
        writer.println(EnumCommands.RELEASE_MOUSE.getAbbrev());
        int button = e.getButton();
        int xButton = 16;
        if (button == 3) {
            xButton = 4;
        }
        writer.println(xButton);
        writer.flush();
    }

    public void keyPressed(KeyEvent e) {
        System.out.println("Key Pressed");
        writer.println(EnumCommands.PRESS_KEY.getAbbrev());
        writer.println(e.getKeyCode());
        writer.flush();
    }
}
```

➤ Lớp ClientScreenReciever.

Nhận màn hình thu được được từ Client gửi đến, sau đó hiển thị chúng.

Code:

```
while(continueLoop){
    System.out.println("trying to New image recieve");
    ImageIcon imageIcon = (ImageIcon) cObjectInputStream.readObject();
    System.out.println("New image recieved");
    Image image = imageIcon.getImage();
    image = image.getScaledInstance(cPanel.getWidth(),cPanel.getHeight(),
                                   Image.SCALE_FAST);
    Graphics graphics = cPanel.getGraphics();
    graphics.drawImage(image, 0, 0,
        cPanel.getWidth(),cPanel.getHeight(),cPanel);
}
```

➤ Lớp EnumCommands.

Xác định các hằng số để biểu diễn lệnh từ máy Server.

➤ **Lớp ProcessServer.**

Lớp này dùng để kiểm soát mọi chức năng trên giao diện (Kết thúc, Phản hồi, Giới thiệu,...), cũng như đưa ra các nhắc nhở khi quản trị viên cần thực hiện chức năng.

➤ **Lớp Remote.**

Lớp này dùng để thêm các Client vào danh sách mỗi khi một Client kết nối đến.

Code:

```
while(true){
    Socket SOCK=SERVER.accept();
    ConnectionArray.add(SOCK);

    System.out.println("Client kết nối
từ:"+SOCK.getLocalAddress().getHostName());
```

➤ **Lớp ResponClient.**

Lớp này dùng để nhận và phản hồi thông điệp yêu cầu từ phía Client gửi đến.

Code:

```
INPUT=new Scanner(SOCK.getInputStream());
OUT=new PrintWriter(SOCK.getOutputStream());
while(true)
{
    CheckConnection();
    if(!INPUT.hasNext()){
        return;
    }
    MESSAGE=INPUT.nextLine();
    for(int i=1;i<=Remote.ConnectionArray.size();i++)
    {
        Socket TEMP_SOCK=(Socket) Remote.ConnectionArray.get(i-1);
        PrintWriter TEMP_OUT=new
PrintWriter(TEMP_SOCK.getOutputStream());
        TEMP_OUT.println(MESSAGE);
        TEMP_OUT.flush();
        System.out.println("Sent to
:"+TEMP_SOCK.getLocalAddress().getHostName());
    }
}
```

➤ **Lớp Server_GUI.**

Lớp này dùng để tạo giao diện ứng dụng.

➤ **Lớp ServerInitiator.**

Chờ kết nối từ Client. Và tạo 1 phần giao diện.

Code:

```
Socket client = new Socket(o,12345);
    System.out.println("Connection Established.");
    new ClientHandler(client,desktop);
}
public void drawGUI(){
    frame.add(desktop, BorderLayout.CENTER);
    frame.addWindowListener(new WindowAdapter() {
```

2.3.2 Chương trình trên Client (RemoteClient).

➤ **Lớp ClientInitiator.**

Lớp này là lớp chính của chương trình trên Client. Dùng để kết nối tới Server và thực hiện gọi các phương thức từ các lớp khác.

➤ **Lớp EnumCommands.**

Lớp này dùng để xác định các hằng số được sử dụng để biểu diễn các lệnh của máy Server.

➤ **Lớp ProcessClient.**

Lớp này dùng để kiểm soát các chức năng trên giao diện (Ngắt kết nối, kết nối tới Server, Gửi yêu cầu, ...), cũng như nhắc nhở người dùng khi thực hiện chức năng (ví dụ Thoát ứng dụng).

➤ **Lớp RemoteClient.**

Lớp này dùng để tạo giao diện phía Client, thực thi các chức năng Shutdown và điều khiển màn hình từ phía Server kích hoạt.

Code:

```
String timef="shutdown -s -f -t ";
    String exrun=timef+"60";

    Runtime r=Runtime.getRuntime();
```

➤ **Lớp ScreenSpyer.**

Lớp này dùng để chụp màn hình và gửi chúng xuyên suốt tới Server.

Code:

```
while(continueLoop){
    BufferedImage image = robot.createScreenCapture(rectangle);
    ImageIcon imageIcon = new ImageIcon(image);
    try {
        System.out.println("before sending image");
        oos.writeObject(imageIcon);
        oos.flush();
        oos.reset();
        System.out.println("New screenshot sent");
    }
```

➤ **Lớp ServerDelegate.**

Lớp này dùng để nhận lệnh từ Server và thực hiện chúng ngay trên chính Client.

Code:

```
Scanner scanner = null;
try {

    System.out.println("Preparing InputStream");
    scanner = new Scanner(socket.getInputStream());

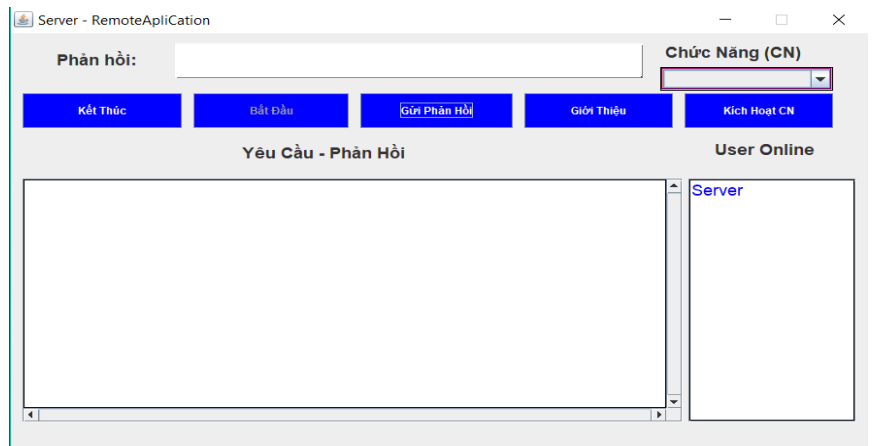
    while(continueLoop){

        System.out.println("Waiting for command");
        int command = scanner.nextInt();
        System.out.println("New command: " + command);
        switch(command){
            case -1:
                robot.mousePress(scanner.nextInt());
                break;
            case -2:
                robot.mouseRelease(scanner.nextInt());
                break;
            case -3:
                robot.keyPress(scanner.nextInt());
                break;
            case -4:
                robot.keyRelease(scanner.nextInt());
                break;
        }
```

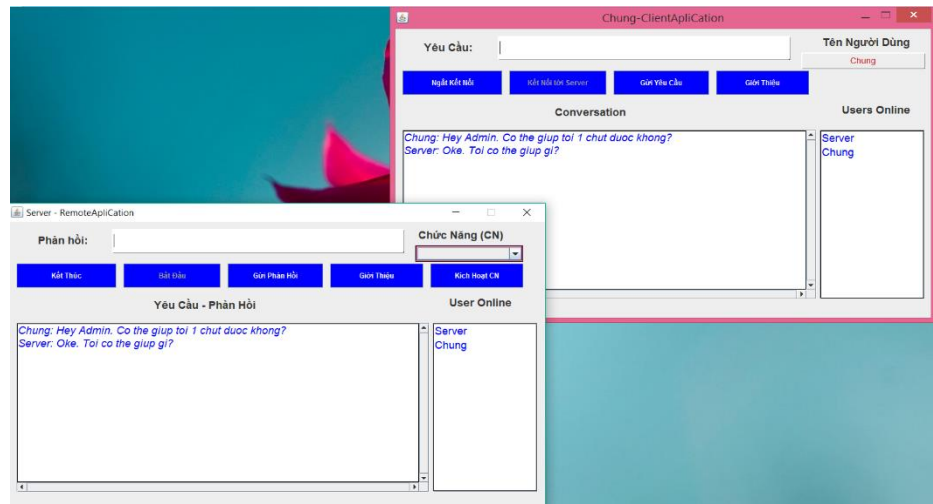
2.4 Kết quả khi chạy chương trình.

2.4.1 Phía Server.

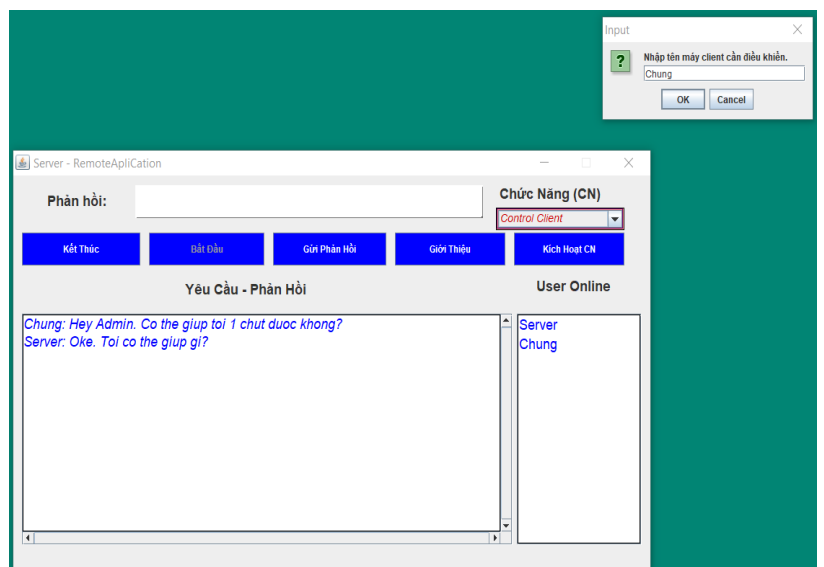
- Khi chương trình được khởi động. Cần click chuột vào “Bắt đầu” để kích hoạt chương trình và chờ các kết nối từ Client đến.



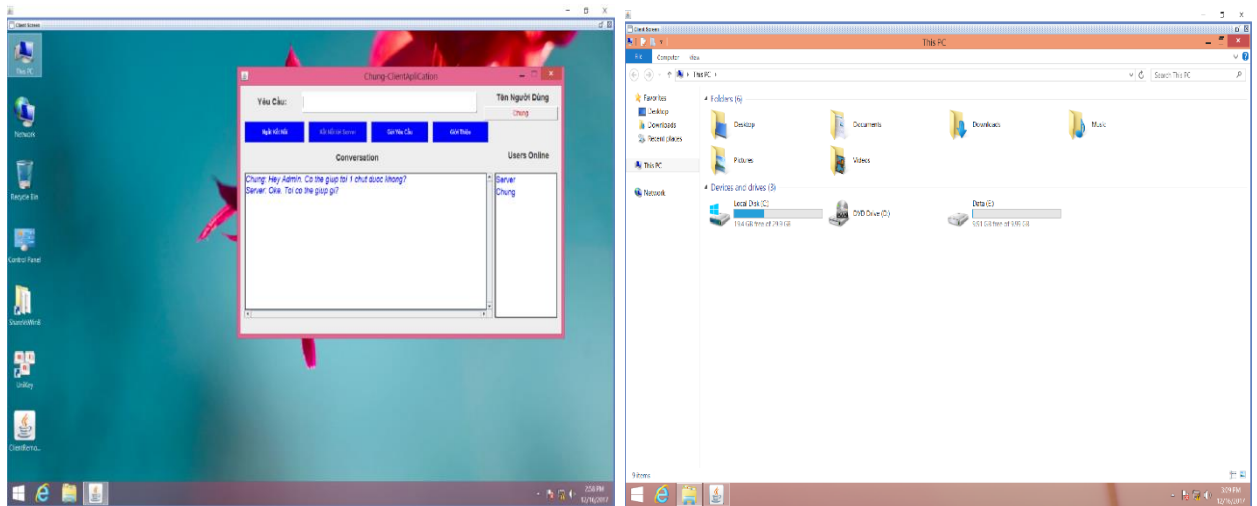
- Nhận được thông điệp từ Client và phản hồi lại.



- Để có thể điều khiển cũng như Shutdown, người dùng cần click chuột vào mũi tên “Chức năng” rồi chọn chức năng cần thực hiện, sau đó nhập tên máy Client để thực hiện chức năng. Ví dụ chức năng điều khiển. Chức năng Shutdown tương tự.

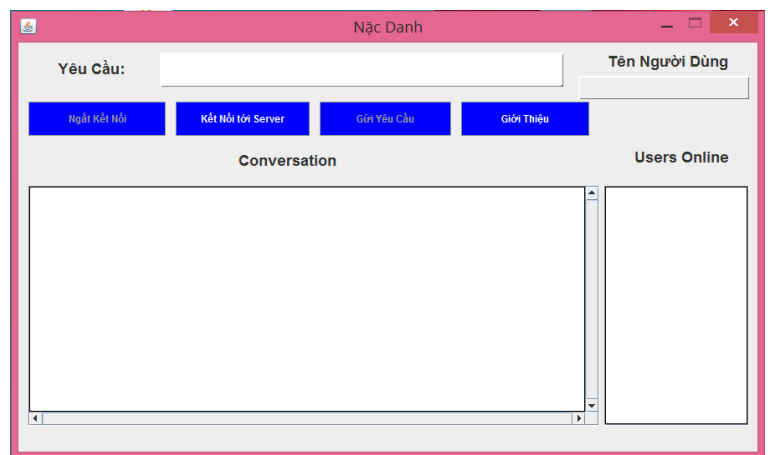


- Kết quả là thấy được màn hình máy Client cần theo dõi và điều khiển.

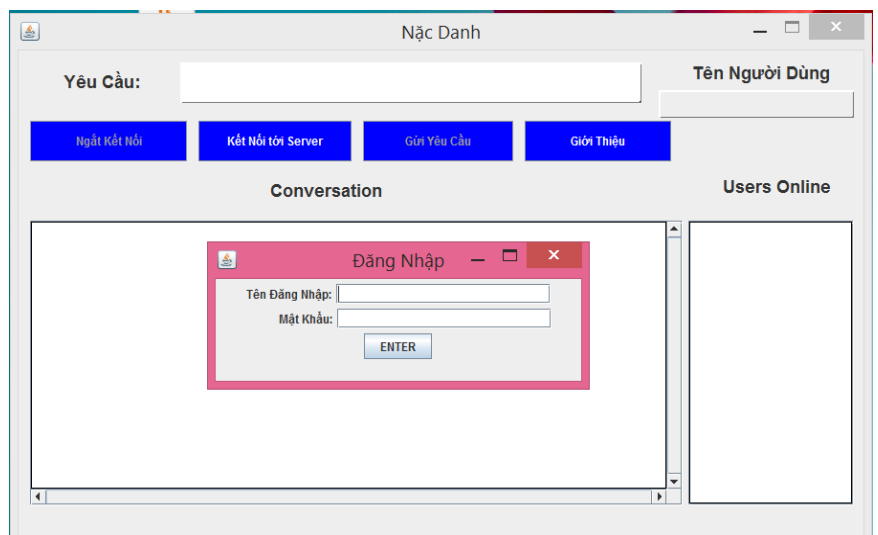


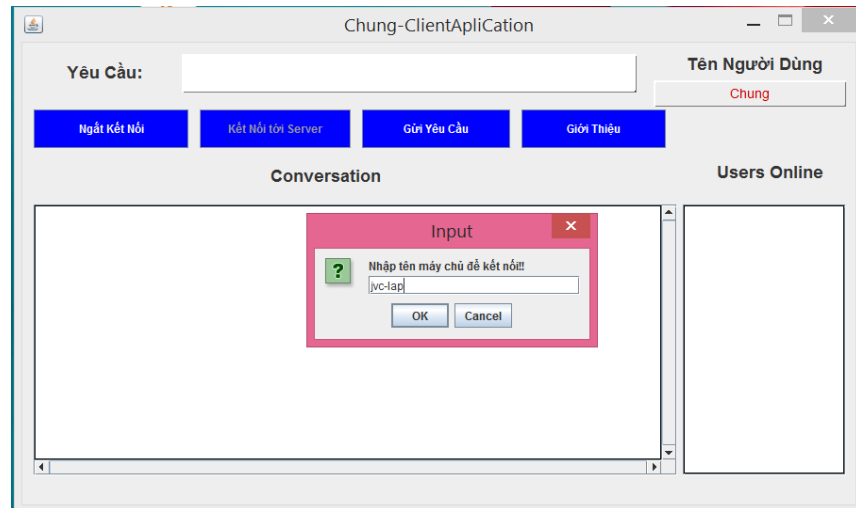
2.4.2 Phía Client.

- Chương trình sau khi được khởi động lên.

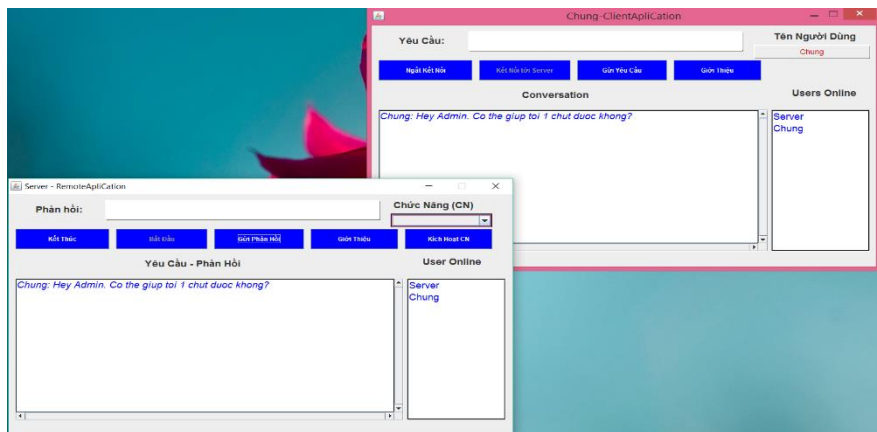


- Để bắt đầu người dùng cần click chuột vào “Kết nối tới Server” và sau đó đăng nhập bằng tên và mật khẩu là “client” và nhập tên máy Server cần kết nối tới.

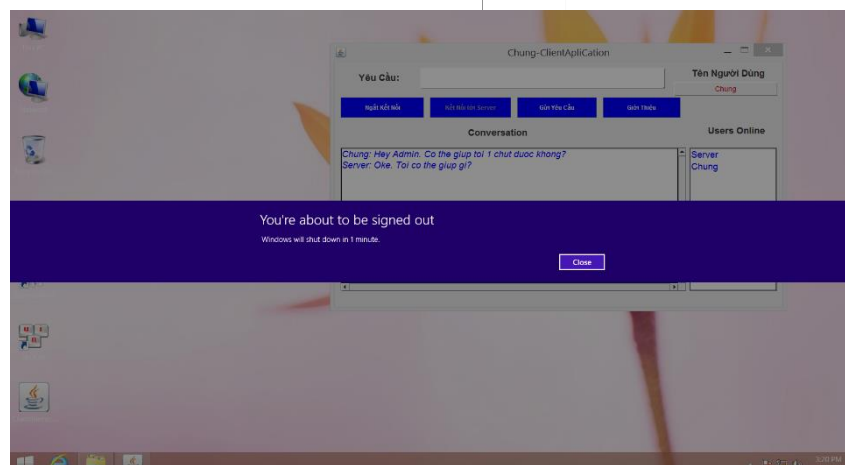
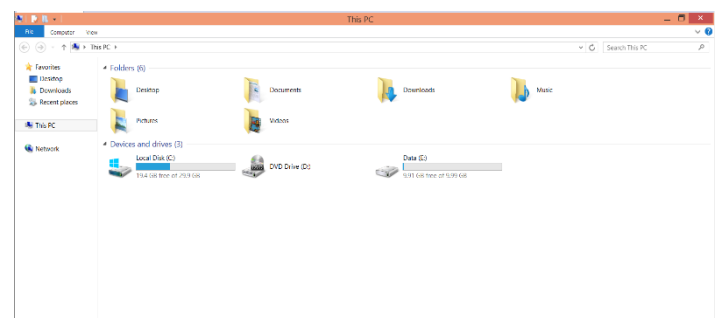




- Tên máy Client đã được thêm vào “User Online” và người dùng có thể gửi yêu cầu tới Server.



- Máy Client bị điều khiển và chuẩn bị tắt máy sau khi phía Server thực hiện chức năng “Shutdown”.



3. Kết Luận.

3.1 Tự đánh giá nhận xét.

Yêu cầu Giảng viên (GV) đưa ra là “Nhận được màn hình và điều khiển được máy Client”. Nhóm đã hoàn thành được yêu cầu của GV. Nhưng trong quá trình làm việc, nhóm ít trao đổi với GV, nên mức độ hoàn thiện ứng dụng có thể chưa được tốt. Ứng dụng mới đang ở mức cơ bản.

3.2 Chức năng làm thêm.

Ngoài yêu cầu chính thì nhóm em có làm thêm được chức năng đó là:

- Gửi và phản hồi thông điệp giữa Client và Server.
- Shutdown máy Client.

3.3 Hướng phát triển trong tương lai.

Trong tương lai ứng dụng sẽ được.

- Chính chu hơn về phần giao diện,
- Thêm nhiều chức năng hơn ví dụ: Chat, Truyền nhận file, Share màn hình máy Server cho Client, ...
- Có kết nối với cơ sở dữ liệu để phân quyền người dùng.
- Tốc độ truyền nhận (gửi và phản hồi) được cải thiện.
- Hỗ trợ tiếng Việt.
- Kết nối thông qua mạng có dây và không dây.
- Tương tác 2 chiều, vừa có thể là Server vừa là Client khi cần.

Tài liệu tham khảo.

1. Sách: Object-oriented programming with Java: Essentials and applications

Link: <https://goo.gl/UW3S4m>

2. Tìm hiểu về lập trình Socket trên java..

Link 1: <https://goo.gl/4WNaSq>

Link 2: <https://goo.gl/oYRE9T>

Link 3: <https://goo.gl/6a628L>

3. Các chương trình mẫu tham khảo.

Link 1: <https://goo.gl/oMnrnL>

Link 2: <https://goo.gl/c8EaJL>

Link 3: <https://goo.gl/X2ALRS>

4. Link cài Vmware: <https://goo.gl/Zdyr24>

5. Link cài window 7 trên Vmware: <https://goo.gl/kx6kdd>

6. Link thiết lập mạng LAN giữa 2 máy: <https://goo.gl/5cU7gJ>

