Mục lục

[NHẬN XÉT CỦA GIẢNG VIÊN 2](#_Toc495959072)

[1. Giới thiệu 3](#_Toc495959073)

[1.1. Tên đề tài và nhiệm vụ: 3](#_Toc495959074)

[1.2. Kết quả đạt được: 3](#_Toc495959075)

[2. Phân tích lý thuyết 4](#_Toc495959076)

[2.1. Một vài lí thuyết cơ bản về Client và Server 4](#_Toc495959077)

[2.2. Socket, Port (Java) 4](#_Toc495959078)

[2.3. Chia sẻ màn hình 6](#_Toc495959079)

[3. Triển khai 6](#_Toc495959080)

[3.1. Mô hình 6](#_Toc495959081)

[3.2. Thực hiện 7](#_Toc495959082)

[4. Kết luận 8](#_Toc495959083)

[4.1. Sản phẩm và hướng phát triển 8](#_Toc495959084)

[4.2. Tài liệu tham khảo 8](#_Toc495959085)

[4.3. Bảng phân công công việc 9](#_Toc495959086)

# NHẬN XÉT CỦA GIẢNG VIÊN

# 1. Giới thiệu

Cuộc cách mạng công nghệ hiện đại đang không ngừng được cải tiến tốt hơn từng ngày, từng giờ. Sự phát triển của kỹ thuật điện tử đã tạo ra hàng loại những thiết bị dùng để tối ưu cuộc sống của chúng ta, với nhiều đặc điểm nổi bậc như sự chính xác cao, tốc độ nhanh, gọn nhẹ. Song song đó, nhóm ngành công nghệ thông tin cũng luôn cải thiện, cập nhật những điều mới mẻ. Cuộc sống hiện đại ngày càng gắn bó với những thiết bị điều khiển từ xa, nó khiến cho con người làm việc “dễ dàng” hơn, và cải thiện tốc độ công việc, đặc biệt là việc “Điều khiển máy tính từ xa”.

Trước kia, khi không có phần mềm điều khiển máy tính từ xa, các nhân viên kĩ thuật của công ty phần mềm phải rất vất vả khi đội mưa, đội nắng tới hỗ trợ khách hàng. Điều này vừa tốn nhiều công sức, vừa tốn nhiều thời gian cho cả khách hàng và nhân viên, cho nên phần mềm điều khiển máy tính từ xa ra đời với mục đích hỗ trợ khách hàng ngay lập tức, giúp công việc của bạn trở nên đơn giản, nhanh chóng, thuận tiện hơn.

## 1.1. Tên đề tài và nhiệm vụ:

Tên đề tài: CONTROLLING MACHINES REMOTELY WITH JAVA.

Nhiệm vụ: Chia sẻ được màn hình hai máy thông qua mạng LAN.

## 1.2. Kết quả đạt được:

Thông qua Lan để kết nối Client tới Server, yêu cầu truy cập vào máy Server.

Client có thể view màn hình của Server.

Có các sự kiện của chuột cũng như bàn phím Server thông qua LAN. Hiện tại, Client đã có thể *điều khiển* được máy Server.

# 2. Phân tích lý thuyết

## 2.1. Một vài lí thuyết cơ bản về Client và Server

Về cơ bản, để thực hiện việc điều khiển từ xa chúng ta cần 2 yếu tố quan trọng là “máy chủ” (Server) - thiết bị bị điều khiển, “máy khách” (Client) - máy dùng để điều khiển máy chủ. Phần mềm hỗ trợ điều khiển từ xa được cài đặt trên cả Client và Server.

Mô hình Client – Server được áp dụng rộng rãi, ý tưởng của nó là Client sẽ gửi những yêu cầu (Request) để Server (đóng vài trò người cung cấp dịch vụ) sẽ xử lý và trả về kết quả cho Client. Ở đây, Client sẽ gửi yêu cầu việc chia sẻ màn hình đến Server, Server chấp nhận và chia sẻ màn hình cho Client thông qua cùng một mạng LAN.

Đối với một số hệ điều hành Windows, khi bạn sử dụng Điều khiển máy tính để kết nối đến máy chủ từ xa, cùng một lúc chỉ có 1 người có thể sử dụng máy chủ đó (1 Server, 1 Client). Trong trường hợp này, Server là chương trình được cài đặt trên máy chủ đó và Server này chỉ chấp nhận 1 Client trong một thời gian nhất định. Client là chương trình Điều khiển máy tính trên máy của bạn.

## 2.2. Socket, Port (Java)

Trước tiên cần nói đến TCP, TCP (Transmission Control Protocol) là một giao thức hướng kết nối, nó cung cấp một đường truyền dữ liệu tin cậy giữa hai máy tính. Tính tin cậy thể hiện ở việc nó đảm bảo dữ liệu được gửi sẽ đến được đích và theo đúng thứ tự như khi nó được gửi. TCP đặc trưng được sử dụng qua Internet Protocol, được xem như là TCP/IP.

***Socket:***

Có nhiều định nghĩa khác nhau về Socket tùy theo các nhìn của người sử dụng: Một cách tổng quát nhất có thể định nghĩa Một Socket là một điểm cuối trong một kết nối giữa hai chương trình đang chạy trên mạng.

Các Socket cung cấp kỹ thuật giao tiếp giữa hai máy tính sử dụng TCP. Một chương trình Client tạo một Socket trên đầu cuối của giao tiếp và cố gắng để kết nối Socket đó tới một Server.

Khi kết nối được tạo, Server tạo một đối tượng Socket trên đầu cuối của giao tiếp. Client và Server bây giờ có thể giao tiếp bằng việc đọc và ghi từ Socket.

Sau khi các kết nối được thành lập, giao tiếp có thể xảy ra bởi sử dụng I/O stream. Mỗi Socket có cả một OutputStream và InputStream. OutputStream của Client được kết nối với InputStream của Server, và InputStream của Client được kết nối với OutputStream của Server.

|  |  |  |
| --- | --- | --- |
| Giao thức | Client | Server |
| Hướng kết nối (TCP/IP) | Socket | ServerSocket |

***Port:***

Ta có thể hiểu Port là một cổng kết nối được mở ra bởi máy chủ và luôn ở trạng thái “sẵn sàng kết nối”. Hình dung như một chỗ ngồi trong nhà hàng luôn có ở đó để tiếp khách vậy. Khi một Client muốn kết nối, sử dụng dịch vụ mà Server cung cấp, Client đó sẽ cần địa chỉ cụ thể của Server bao gồm IP và Port tương ứng với dịch vụ cần sử dụng.

Ví dụ: Server quy định Port 11111 cho dịch vụ gửi và nhận mail, Port 22222 cho dịch vụ tải tập tin lên Server. Khi Client muốn sử dụng dịch vụ gửi, nhận mail, nó sẽ đi vào Port 11111, nếu muốn sử dụng dịch vụ truyền tải thông tin ở cổng Port 22222, Client cần “mở” cổng 22222 đó và tiến hành sử dụng dịch vụ do Server cung cấp.

Port có thông số chạy từ 1-65535 (nhưng nên chọn từ 1024 trở lên để tránh những Port đã được quy định sẵn; ở ngôn ngữ lập trình Java, khi đặt cho 1 Port thông số dưới 1024, ta cần một số lệnh đặc biệt khác).

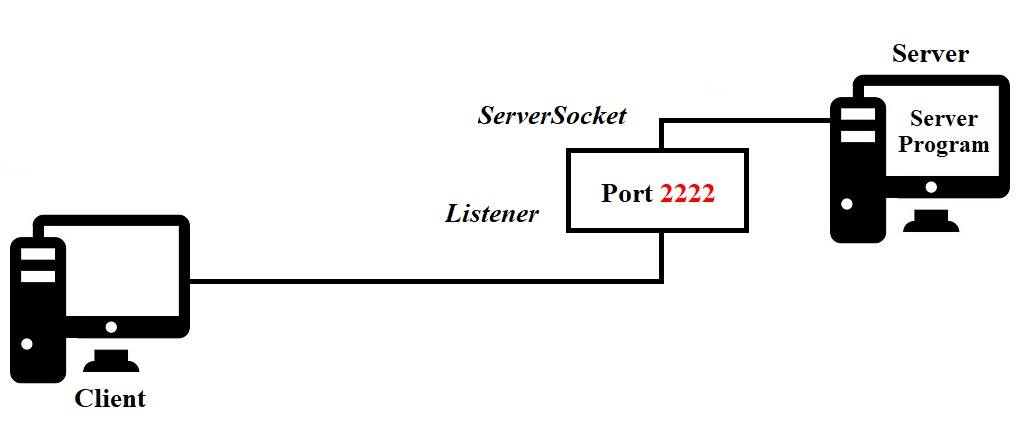
## 2.3. Chia sẻ màn hình

Client gửi yêu cầu đến Server, mong muốn nhận về ảnh chụp màn hình. Sau khi đã được kết nối với nhau thông qua LAN, chúng cần những phương thức nhất định để có thể tiến hành gửi cũng như đáp ứng những yêu cầu đó. Trong ngôn ngữ lập trình Java, Java có sẵn một con Robot sẵn sàng đáp ứng những nhu cầu này của người dùng. Nó sẽ thực hiện hầu như toàn bộ công việc ta cần: chụp màn hình, định vị vị trí chuột khi di chuyển, vị trí click chuột, xác định phím được nhấn. Tương ứng với các công việc đó là các Port được quy định rõ ràng, nhờ vào địa chỉ IP và các Port này, Robot có thể thực hiện các công việc trên để Server có thể tiến hành chia sẻ màn hình cũng như việc điều khiển Client của Server.

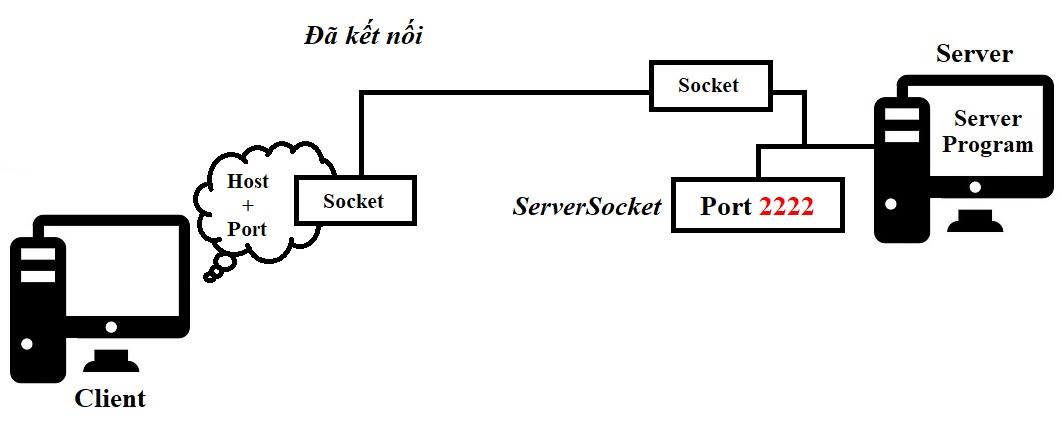
# 3. Triển khai

## 3.1. Mô hình

Trình điều khiển từ xa có thể được minh họa bằng mô hình như sau:



Về phía Client, người dùng tiến hành gửi một yêu cầu kết nối tới máy chủ thông qua địa chỉ IP và cổng kết nối (port) được cài đặt trong chương trình. Sau khi các thông số được nhập chính xác, máy chủ cho phép kết nối và Client có thể tiến hành điều khiển máy chủ bằng chương trình thông qua kết nối trong Lan.



## 3.2. Thực hiện

Đầu tiên, ở Server, cần khởi tạo những Port có thông số cụ thể. Đồng thời thiết lập để Server cho phép kết nối khi có yêu cầu của Client đến Port này.

Tiếp theo, thực hiện các phương thức gửi và nhận ở cả Client và Server. Thông tin vào của cái này là đầu ra của cái kia, và ngược lại. Bằng cách này, Client và Server có thể giao tiếp qua lại với nhau.

Sau khi đã có đường truyền dữ liệu, ta thực hiện phần Robot Action. Cụ thể là các công việc chụp màn hình, đọc vị trí chuột, vị trí click chuột và các phím được nhấn. Sau đó nhờ các phương thức gửi và nhận đã có, ta có thể truyền những thông số mà máy có thể hiểu là vị trí hay tọa độ yêu cầu.

Cuối cùng, hoàn thành Form cho Server để hiển thị địa chỉ IP, qua đó Client có thể sử dụng địa chỉ IP này để kết nối tới Server. Quá trình được thực hiện qua 1 Form đơn giản, dễ sử dụng với người dùng.

# 4. Kết luận

## 4.1. Sản phẩm và hướng phát triển

Qua những kiến thức được trang bị tại trường cũng như quá trình tự học tại nhà, Nhóm đã xây dựng được trình điều khiển từ xa theo mô hình Client - Server. Chương trình có điểm tối ưu là có sự kiện chuột và bàn phím, giao diện rất đơn giản và dễ sử dụng.

Hiện tại, chương trình có thể phát triển thêm việc kết nối cho một mạng có nhiều Client thay vì 1 Client - 1 Server như hiện nay. Ngoài ra, sẽ có thêm hệ thống tự động lấy địa chỉ IP từ máy chủ, sau đó tiến hành điều khiển hoặc yêu cầu các tính năng khác.

## 4.2. Tài liệu tham khảo

<http://o7planning.org/vi/10393/huong-dan-lap-trinh-java-socket>

<https://vi.wikipedia.org/wiki/Client-server>

<http://tech.agu.edu.vn/clbtinhoc/index.php?threads/java-huong-dn-vit-chuong-trinh-chat-client-server-don-gian.512/>

<http://congdongjava.com/forum/threads/hỏi-về-lỗi-kết-nối-socket-trong-mạng-lan.2743/>

<http://phocode.com/java/javaswing/java-swing-quan-ly-layout/>

<https://tranhoangminh.wordpress.com/2012/10/27/cach-xac-dinh-dia-chi-ip-cua-may-bang-java/>

<https://www.daniweb.com/programming/software-development/threads/477492/remote-desktop-software-using-java>

<http://www.java2s.com/Code/JavaAPI/java.awt/ToolkitgetScreenSize.htm>

### 4.4. Bảng phân công công việc

|  |  |  |
| --- | --- | --- |
| Tuần | Công việc | Thành viên tham gia |
| 1 | Tìm hiểu chủ đề: Socket, Port  Tìm hiểu chung về ngôn ngữ Java | Vũ Đức Anh Tuấn,  Huỳnh Quốc Thắng |
| 2 | Tìm kiếm thêm thông tin về chủ đề  Tiếp tục tìm hiểu thêm về ngôn ngữ Java | Vũ Đức Anh Tuấn,  Huỳnh Quốc Thắng |
| Phác thảo giao diện sơ cấp | Huỳnh Quốc Thắng |
| 3 | Nghiên cứu cụ thể về Socket và Port trong Java (các phương thức có trong Java) | Vũ Đức Anh Tuấn,  Huỳnh Quốc Thắng |
| 4 | Dựng được giao diện cơ bản cho chương trình  Nghiên cứu về cách kết nối 2 máy qua Lan | Vũ Đức Anh Tuấn,  Huỳnh Quốc Thắng |
| 5 | Thực hiện code cho Class Server và Client phần kết nối qua Socket và Port, tiến hành gửi tin nhắn từ qua lại giữa Client và Server để kiểm tra kết nối | Vũ Đức Anh Tuấn,  Huỳnh Quốc Thắng |
| Kiểm tra lại sau khi hoàn thành  Đăng tải bài làm lên Github | Huỳnh Quốc Thắng |
| 6 | Thực hiện code phần truy cập màn hình cho class Server và Client | Vũ Đức Anh Tuấn,  Huỳnh Quốc Thắng |
| Kiểm tra lại sau khi hoàn thành | Vũ Đức Anh Tuấn |
| Đăng tải bài làm lên Github | Huỳnh Quốc Thắng |
| 7 | Hoàn thiện việc truy cập màn hình (phát hiện được lỗi)  Làm thêm 1 class tạo giao diện cơ bản cho Server, hiển thị được IP của Server | Vũ Đức Anh Tuấn,  Huỳnh Quốc Thắng |
| Đăng tải bài báo cáo lên Github | Huỳnh Quốc Thắng |
| 8 | Hoàn thành các phương thức nhận và gửi cho Client và Server  Hoàn thành sự kiện chuột và bàn phím | Vũ Đức Anh Tuấn,  Huỳnh Quốc Thắng |
| 9 | Tiến hành kiểm tra cuối, hoàn thành chương trình  Hoàn thành bài báo cáo  Đăng tải lên Github | Vũ Đức Anh Tuấn,  Huỳnh Quốc Thắng |