ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN



**BÁO CÁO ĐỒ ÁN**

**ĐỀ TÀI : PHÁT HIỆN THIẾT BỊ LƯU TRỮ NGOÀI**

**Ở MẠNG LAN**

**GVHD: Lê Trần Đức**

**NHÓM 9: Nguyễn Đặng Trường Sơn**

**Lê Quang Thông**

**MỤC LỤC**

[DANH SÁCH HÌNH VẼ:…………………………………………………... 2](#_Toc57764331)

[GIỚI THIỆU ĐỀ TÀI 3](#_Toc57764332)

[LỜI NÓI ĐẦU 4](#_Toc57764333)

[ĐẶT VẤN ĐỀ : 5](#_Toc57764334)

[I. NGUYÊN LÝ HỆ ĐIỀU HÀNH : 5](#_Toc57764335)

[1. CƠ SỞ LÝ THUYẾT : 5](#_Toc57764336)

[2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG: 6](#_Toc57764337)

[3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ: 6](#_Toc57764339)

[II. LẬP TRÌNH MẠNG : 6](#_Toc57764344)

[1. CƠ SỞ LÝ THUYẾT : 6](#_Toc57764345)

[1.1 Mạng máy tính : 6](#_Toc57764346)

[1.2 Lập trình mạng : 10](#_Toc57764347)

[1.3 IP Address của máy Client: 14](#_Toc57764352)

[2 PHÂN TÍCH THIẾT KẾ HỆ THỐNG: 14](#_Toc57764353)

[a)Giao thức websocket: 14](#_Toc57764354)

[b)Hệ thống chương trình : 15](#_Toc57764355)

[3 TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ : 16](#_Toc57764356)

[a) Client : 16](#_Toc57764357)

[b) Sever: 17](#_Toc57764358)

[KẾT LUẬN CHUNG : 19](#_Toc57764359)

[ƯU ĐIỂM, NHƯỢC ĐIỂM : 19](#_Toc57764360)

# DANH SÁCH HÌNH VẼ:

Hình 1: Test phần mềm phát hiện thiết bị

Hình 2 : Mô hình client-serve

Hình3 : Ví dụ về mô hình Client-Serve

Hình 4 : mô tả hoạt động websocket

Hình 5 :client-server trong websocket (socket.io)

Hình 6: Thông tin địa chỉ mạng của 1 máy client

Hình 7: Sơ đồ khối hệ thống

Hình 8 Ứng dụng usb-detected.exe

Hình 9: Hiển thị thông báo phát hiện thiết bị lưu trữ ngoài

Hình 10: Hiền thị danh sách từng Client

Hình 11: Các thông số của thiết bị lưu trữ ngoài ở Client

Hình 12: Chức năng dành cho người quản trị

# GIỚI THIỆU ĐỀ TÀI

**Xây dựng ứng dụng phát hiện sử dụng thiết bị lưu trữ gắn ngoài trong mạng LAN**

Mô tả đề tài : Hệ thống mạng LAN thông thuờng duợc sử dụng trong các phòng lab, các hệ thống nội bộ. Ðể kiểm soát việc thất thoát dữ liệu hoặc tang tính an toàn cho hệ thống, tránh các xâm nhập mã dộc từ bên ngoài, trong quá trình triển khai cần có sự giám sát việc sử dụng thiết bị luu trữ gắn ngoài nhu USB flash, ổ đĩa gắn ngoài thông qua cổng giao tiếp USB…Loại trừ sử dụng các công cụ quản trị mạng dể ngan chặn hoạt dộng của cổng USB trên máy tính trong mạng.

**LỜI NÓI ĐẦU**

Với kiến thức mà đã học được từ bộ môn Hệ điều hành và Mạng máy tính. Nhóm đã trang bị những kiến thức đầy đủ để có thể thực hiện đồ án lần này. Nhóm chúng em xin chân thành cảm ơn thầy Lê Trần Đức và Công ty TNHH tư vấn công nghệ Lê Gia vì đã hướng dẫn nhóm thực hiện hoàn thiện đồ án lần này

Vì đồ án lần này khá mới với nhóm và kiến thức tự tìm hiểu được cũng có hạn nên đồ án có những sai sót và thiếu hiệu quả mong thầy cô và doanh nghiệp có thể đưa ra những lời khuyên cho nhóm. Mọi chỉ bảo của thầy cô, doanh nghiệp đểu là kiến thức mới để nhóm có thể hoàn thiện đồ án hơn. Và có thể phát triển đồ án đi vào thực tế

Nhóm em xin chân thành cảm ơn!

**ĐẶT VẤN ĐỀ :**

- Trong một hệ thống mạng LAN như phòng LAB, hệ thống nội bộ, việc rò rỉ thông tin thông qua việc sử dụng thiết bị lưu trữ ngoài là rất nguy hiểm. Do đó cần có biện pháp cho người quản lý có thể biết được thông báo khi có một máy nào đó sử dụng thiết bị lưu trữ ngoài này

- Do đó, việc xây dựng phần mềm phát hiện thiết bị lưu trữ ngoài và gửi thông báo tự động ấy cho người quản lý là biện pháp tối ưu.

- Nhóm xây dựng phần mềm phát hiện USB DETECTED và gửi thông báo từ các máy CLIENT đến SEVER

1. **NGUYÊN LÝ HỆ ĐIỀU HÀNH :**
2. **CƠ SỞ LÝ THUYẾT :**
3. **NVM :**

- NMP là viết tắt của Node package manager là một công cụ tạo và quản lý các thư viện lập trình Javascript cho [Node.js](https://nodejs.org/). Trong cộng đồng Javascript, các lập trình viên chia sẻ hàng trăm nghìn các thư viện với các đoạn code đã thực hiện sẵn một chức năng nào đó. Nó giúp cho các dự án mới tránh phải viết lại các thành phần cơ bản, các thư viện lập trình hay thậm chí cả các [framework](https://topdev.vn/blog/framework-la-gi/).

- Mỗi đoạn code này có thể phụ thuộc vào rất nhiều các mã nguồn mở khác, sẽ dễ dàng khi có các công cụ quản lý thư viện ra đời, nếu không sẽ mất rất nhiều công sức trong việc quản lý các thư viện này.

1. **Package.json:**

- Để quản lý các gói cài đặt cục bộ bằng npm thì cách tốt nhất là thông qua file package.json, chính là file nằm trong thư mục gốc của project. File [JSON](https://topdev.vn/blog/json-la-gi/) này chứa các nội dung:

* Các gói thư viện lập trình mà project sử dụng.
* Cho phép xác định phiên bản chính xác của các gói thư viện lập trình được sử dụng.
* Các gói bạn xây dựng có thể chia sẻ dễ dàng với các lập trình viên khác trên toàn cầu thông qua npm.

1. **PHÂN TÍCH THIẾT KẾ HỆ THỐNG:**

Chương trình usb-detection:

* Usb-detection : lắng nghe từng sự kiện thêm/ rút USB của bạn vào máy tính

## *usbDetect.startMonitoring() :* bắt đầu lắng nghe về các sự kiện thêm/rút. Nó sẽ khiến quy trình Node.js mở cho đến khi gọi đến usbDetect.stopMonitoring()

## TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ:

## a)Xây dựng trình phát hiện thiết bị lưu trữ ngoài

## API (****Application Programming Interface)**** :

* **Usb-detection :**
* Thư viện cho phép lắng nge các sự kiện cắm, rút thiết bị USB trên hệ thống máy tính .
* Cách sử dụng : var usbDetect **=** require('usb-detection');
* **API :**
* usbDetect.startMonitoring() : bắt đầu lắng nge các sự kiện cắm, rút, thay đổi trên USB. Điều nãy sẽ khiến quá trình Node.js luôn hoạt động đến khi gọi usbDetect.stopMonitoring() .
* usbDetect.stopMonitoring(): ngừng lắng nge các sự kiện cắm, rút,thay đổi trên USB.
* usbDetect.find(): tìm kiếm số thiết bị ngoại vi trên máy.
* usbDetect.on(eventName,callback):
  + - eventName:
* add: sự kiện cắm usb.
* remove: sự kiện rút usb.
* Change sự kiện thay đổi khi rút,cắm.
  1. **Cải tiến phần mềm :**
* **Node disk info:**
* Node module lấy thông tin đĩa trong Windown,Linux hoặc Mac. Thư viện sẽ gọi lệnh hệ thống để lấy thông tin ổ đĩa, phân tích cú pháp kết quả và tải thông tin trong mảng.
* Cách sử dụng:const nodeDiskInfo=require(‘./dist/index’);
* **API:**
* nodeDiskinfo.getDiskInfoSync(): trả về hệ thông tin ổ đĩa trên máy tính.

function countDisk(){

    try {

       const disks = nodeDiskInfo.getDiskInfoSync();

     console.log(disks.length);

    return disks.length;

    } catch (e) {

        console.error(e);

    }

}

var tamp=countDisk();//số ổ đĩa ban đầu của client

 function dulieu(){

    try {

    const disks = nodeDiskInfo.getDiskInfoSync();

     console.log(disks.length);

     if(tamp!=disks.length){//sau khi cắm usb thì số disk tăng lên

            return disks[disks.length1].mounted;//tên ổ : viddu :" G:/

     }

     else return "Không tìm thấy dữ liệu trong USB ";

    } catch (e) {

        console.error(e);

    }

**}**

* **Cho phép tắt nguồn máy client :**
* Sử dụng phương thức exitter() ở phía client để cho phép tắt nguồn.
* Ở phía sever lắng nghe sự kiện khi người dùng click chuột vào Button “Tắt máy ”

socket.on("clienttatmay",function(*data*){

    console.log('dang tat may ');

    console.log(*data*);

    (async () => {

        exiter();

      })();

});// hàm ở phía client

*socket*.on("tatmay",function(*data*){

    console.log('server lang nge dang tat may');

    var socid=*data*.idclient;

        io.to(socid).emit("clienttatmay",*data*);

    });// ở phía sever

## b)Kết quả :

## Test phần mềm phát hiện thiết bị ở chính máy tính của mình :

## 

## *Hình1: Test phần mềm phát hiện thiết bị*

## Mỗi hành động Thêm/Rút đều được phần mềm phát hiện và ngoài ra còn hiển thị thông số của thiết bị

1. **LẬP TRÌNH MẠNG :**
2. **CƠ SỞ LÝ THUYẾT :**
   1. **Mạng máy tính :**

* **Mạng máy tính** là [mạng viễn thông](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_vi%E1%BB%85n_th%C3%B4ng) [kỹ thuật số](https://vi.wikipedia.org/w/index.php?title=T%C3%ADn_hi%E1%BB%87u_k%C4%A9_thu%E1%BA%ADt_s%E1%BB%91&action=edit&redlink=1) cho phép [các nút mạng](https://vi.wikipedia.org/wiki/N%C3%BAt_m%E1%BA%A1ng) chia sẻ tài nguyên. Trong các mạng máy tính, [các thiết bị máy tính](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh) [trao đổi dữ liệu](https://vi.wikipedia.org/wiki/Truy%E1%BB%81n_d%E1%BB%AF_li%E1%BB%87u) với nhau bằng các kết nối ([liên kết dữ liệu](https://vi.wikipedia.org/w/index.php?title=Li%C3%AAn_k%E1%BA%BFt_d%E1%BB%AF_li%E1%BB%87u&action=edit&redlink=1)) giữa các nút. Các liên kết dữ liệu này được thiết lập qua [cáp mạng](https://vi.wikipedia.org/w/index.php?title=C%C3%A1p_m%E1%BA%A1ng&action=edit&redlink=1) như dây hoặc [cáp quang](https://vi.wikipedia.org/wiki/C%C3%A1p_quang) hoặc [phương tiện không dây](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_kh%C3%B4ng_d%C3%A2y) như [Wi-Fi](https://vi.wikipedia.org/wiki/Wi-Fi).
* Các thiết bị máy tính mạng làm nhiệm vụ khởi động, [định tuyến](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%8Bnh_tuy%E1%BA%BFn) và chấm dứt dữ liệu được gọi là các nút mạng.[[1]](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_m%C3%A1y_t%C3%ADnh#cite_note-1) Các nút thường được xác định bởi [địa chỉ mạng](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%8Ba_ch%E1%BB%89_m%E1%BA%A1ng) và có thể bao gồm [máy chủ mạng](https://vi.wikipedia.org/w/index.php?title=M%C3%A1y_ch%E1%BB%A7_(m%E1%BA%A1ng)&action=edit&redlink=1) như [máy tính cá nhân](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh_c%C3%A1_nh%C3%A2n), [điện thoại](https://vi.wikipedia.org/wiki/%C4%90i%E1%BB%87n_tho%E1%BA%A1i) và [máy chủ](https://vi.wikipedia.org/wiki/M%C3%A1y_ch%E1%BB%A7), cũng như [phần cứng mạng](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_c%E1%BB%A9ng_m%E1%BA%A1ng_m%C3%A1y_t%C3%ADnh) như bộ định tuyến và chuyển mạch. Hai thiết bị như vậy có thể được cho là được kết nối với nhau khi một thiết bị có thể trao đổi thông tin với thiết bị kia, cho dù chúng có kết nối trực tiếp với nhau hay không. Trong hầu hết các trường hợp, các [giao thức truyền thông](https://vi.wikipedia.org/wiki/Giao_th%E1%BB%A9c_truy%E1%BB%81n_th%C3%B4ng) dành riêng cho ứng dụng được [xếp lớp](https://vi.wikipedia.org/wiki/Giao_th%E1%BB%A9c_truy%E1%BB%81n_th%C3%B4ng) (nghĩa là mang theo [trọng tải](https://vi.wikipedia.org/wiki/Payload_(m%C3%A1y_t%C3%ADnh))) so với các giao thức truyền thông chung khác. Bộ sưu tập [công nghệ thông tin](https://vi.wikipedia.org/wiki/C%C3%B4ng_ngh%E1%BB%87_th%C3%B4ng_tin) ghê gớm này đòi hỏi phải có những người [quản lý mạng](https://vi.wikipedia.org/w/index.php?title=Qu%E1%BA%A3n_l%C3%BD_m%E1%BA%A1ng&action=edit&redlink=1) lành nghề để giữ cho tất cả hệ thống mạng hoạt động tốt.
  + 1. **Mô hình mạng LAN**
* [LAN](https://vi.wikipedia.org/wiki/LAN) (từ [Anh ngữ](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): *local area network*), hay còn gọi là "mạng cục bộ", là mạng tư nhân trong một toà nhà, một khu vực (trường học hay cơ quan chẳng hạn) có cỡ chừng vài km.[[5]](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_m%C3%A1y_t%C3%ADnh#cite_note-5) Chúng nối các [máy chủ](https://vi.wikipedia.org/wiki/M%C3%A1y_ch%E1%BB%A7) và các [máy trạm](https://vi.wikipedia.org/wiki/Workstation) trong các văn phòng và [nhà máy](https://vi.wikipedia.org/wiki/Nh%C3%A0_m%C3%A1y) để chia sẻ tài nguyên và trao đổi thông tin. [LAN](https://vi.wikipedia.org/wiki/LAN) có 3 đặc điểm:
* Giới hạn về tầm cỡ phạm vi hoạt động từ vài mét cho đến 1 km.
* Thường dùng kỹ thuật đơn giản chỉ có một đường dây cáp (*cable*) nối tất cả máy. Vận tốc truyền dữ liệu thông thường là 10 [Mbps](https://vi.wikipedia.org/wiki/%C4%90%C6%A1n_v%E1%BB%8B_t%E1%BB%91c_%C4%91%E1%BB%99_truy%E1%BB%81n_d%E1%BB%AF_li%E1%BB%87u#Megabit_tr%C3%AAn_gi%C3%A2y), 100 Mbps, 1 Gbps, và gần đây là 100 [Gbps](https://vi.wikipedia.org/w/index.php?title=Gbps&action=edit&redlink=1).[[6]](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_m%C3%A1y_t%C3%ADnh#cite_note-6)
* Ba kiến trúc mạng kiểu LAN thông dụng bao gồm:

a) Mạng bus hay mạng tuyến tính. Các máy nối nhau một cách liên tục thành một hàng từ máy này sang máy kia. Ví dụ của nó là [Ethernet](https://vi.wikipedia.org/wiki/Ethernet) (chuẩn [IEEE](https://vi.wikipedia.org/wiki/IEEE) 802.3).

b) Mạng vòng. Các máy nối nhau như trên và máy cuối lại được nối ngược trở lại với máy đầu tiên tạo thành vòng kín. Thí dụ mạng vòng thẻ bài [IBM](https://vi.wikipedia.org/wiki/IBM) (*IBM token ring*).

c) Mạng sao

**1.1.2 Mô hình ứng dụng mạng :**

Một máy tính trên mạng có thể thuộc một trong ba loại như sau:

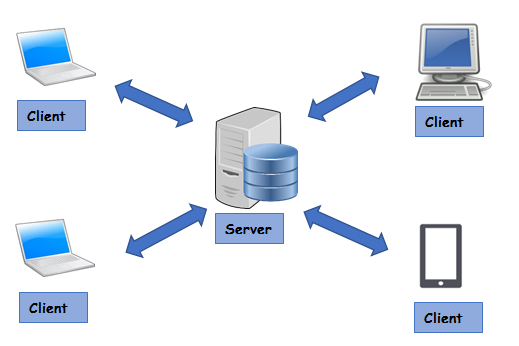
* **Máy trạm (Client)*:*** Không cung cấp tài nguyên mà chỉ sử dụng tài nguyên từ mạng.
* **Máy chủ (Server):** Cung cấp tài nguyên và các dịch vụ cho các máy trên mạng.
* **Peer*:*** Sử dụng tài nguyên và đồng thời cũng cung cấp tài nguyên cho mạng.

Dựa vào cách mà các máy tính được nối vào mạng cũng như cách mà chúng tương tác với mạng và với nhau, mạng máy tính được chia làm ba mô hình cơ bản nhưng với đồ án này, chúng ta chỉ đi xét mô hình client-server. Nên sẽ đi sâu hơn về mô hình này

* 1. **Mô hình Client-Server:**

Mỗi phần mềm xây dựng theo mô hình Client/Server sẽ được chia làm hai phần: phần hoạt động trên máy phục vụ gọi là phần phía Server và phần hoạt động trên trạm làm việc gọi là phần phía Client. Với mô hình này các trạm làm việc cũng được gọi là các Client (hay máy Client) còn các máy phục vụ gọi là các Server. Nhiệm vụ của mỗi phần được quy định như sau:

* Phần phía Server quản lý các giao tiếp môi trường bên ngoài tại Server và với các Client, tiếp nhận các yêu cầu dưới dạng các xâu ký tự (query string), phân tích các query string, xử lý dữ liệu và gửi kết quả trả lời về phía các Client.
* Phần phía Client tổ chức giao tiếp với người dùng, với môi trường bên ngoài tại trạm làm việc và với phía Server, tiếp nhận yêu cầu của người dùng, thành lập các query string gửi về phía Server, tiếp nhận kết quả và tổ chức trình diễn chúng.



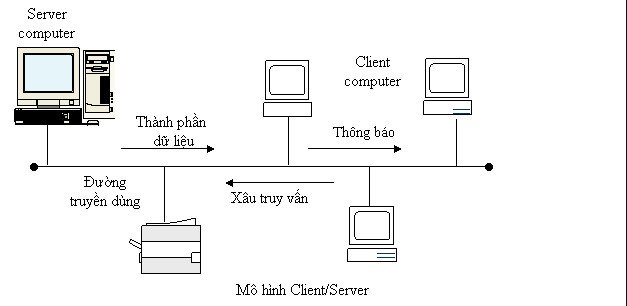
*Hình2 : Mô hình client-serve*

* 1. **Tính năng của Client/Server Network:**

- Client/server network cho phép mạng tập trung các chức năng và các ứng dụng tại một hay nhiều máy dịch vụ file chuyên dụng

- Các máy dịch vụ file trở thành trung của hệ thống

- Hệ điều hành Client/Server network cung cấp cơ chế tích hợp tất cả các bộ phận của mạng và cho phép nhiều người dùng đồng thời chia sẻ cùng một tài nguyên, bất kể vị trí địa lý



*Hình3 : Ví dụ về mô hình Client-Serve*

### 1.2 Lập trình mạng :

**1.2.1 Web thời gian thực ( real time ) :**

Hiện nay ứng dụng web đã phát triển khác xa so với ngày đầu nó xuất hiện, kèm theo đó là vô số các kỹ thuật mới được áp dụng để phục vụ cho quá trình này nhằm đem lại trải nghiệm mới mẻ, đầy hứng thú và cũng không kém phần tiện dụng cho người dùng. Công nghệ web thời gian thực(real time) ngày càng trở nên phổ biết. Vì vậy với yêu cầu của đồ án, nhóm đã lên ý tưởng xây dựng web real time để có thể lắng nge hay phát sự kiện của việc phát hiện usb được nhanh chóng . Có nhiều công nghệ, phương pháp giúp xây dựng ứng dụng thời gian thực

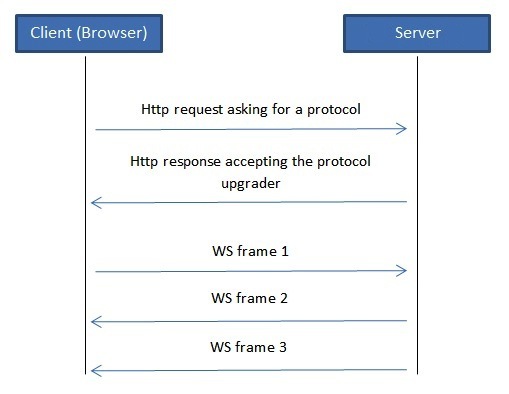
* *AJAX LONG-POLLING:*
* *SERVER SENT EVENTS (SSE)*
* *COMET*
* *WEBSOCKET*
  + 1. **Xây dựng web real-time với websocket :**

WebSocket là một giao thức giúp truyền dữ liệu hai chiều giữa server-client qua một kết nối TCP duy nhất. Hơn nữa, webSocket là một giao thức được thiết kế để truyền dữ liệu bằng cách sử dụng cổng 80 và cổng 443. Vì vậy, webSockets có thể hoạt động trên các cổng web tiêu chuẩn, nên không có rắc rối về việc mở cổng cho các ứng dụng, lo lắng về việc bị chặn bởi các tường lửa hay proxy server

Không giống với giao thức HTTP là cần client chủ động gửi yêu cầu cho server, client sẽ chờ đợi để nhận được dữ liệu từ máy chủ. Hay nói cách khác với giao thức Websocket thì server có thể chủ động gửi thông tin đến client mà không cần phải có yêu cầu từ client.

Tất cả dữ liệu giao tiếp giữa client-server sẽ được gửi trực tiếp qua một kết nối cố định làm cho thông tin được gửi đi nhanh chóng và liên tục khi cần thiết. WebSocket làm giảm độ trễ bởi vì một khi kết nối WebSocket được thành lập, server không cần phải chờ đợi cho một yêu cầu từ client.

Tương tự như vậy, client có thể gửi tin nhắn đến server bất cứ lúc nào. Yêu cầu duy nhất này giúp làm giảm đáng kể độ trễ, mà sẽ gửi một yêu cầu trong khoảng thời gian, cho dù thông điệp có sẵn.



*Hình 4 : mô tả hoạt động websocket*

Giao thức có hai phần: Bắt tay và truyền dữ liệu Ban đầu client sẽ gửi yêu cầu khởi tạo kết nối websocket đến server, server kiểm tra và gửi trả kết quả chấp nhận kết nối, sau đó kết nối được tạo và quá trình gửi dữ liệu có thể được thực hiện, dữ liệu chính là các Ws frame

* + 1. **Socket.io**

**a)Khái niệm :**

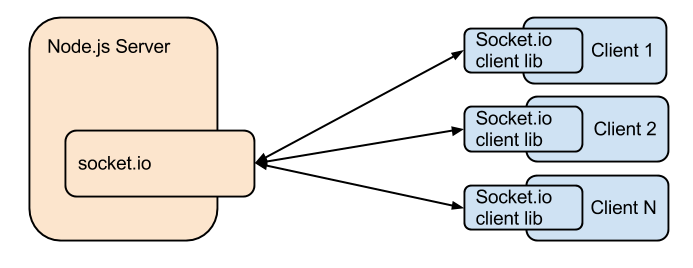
[Socket.IO](http://socket.io/) là một bộ thư viện dành cho các ứng dụng web, mobile đê phát triển các ứng dụng realtime. Với đặc trưng mạnh mẽ và dễ sử dụng, [Socket.IO](http://socket.io/) đang ngày càng được sử dụng rộng rãi từ những trang mạng xã hội cần sự tương tác cao, đến các blog hay các trang web thương mại điện tử. Với bộ thư viện này, làm việc với WebSockets trở nên đơn giản hơn rất nhiều. Thư viện gồm 2 phần

* Phía client: gồm bộ thư viện viết cho web(JavaScript), iOS, Android
* Phía server: viết bằng JavaScript và dùng cho các máy chủ node.JS

[Socket.IO](http://socket.io/) hỗ trợ sử dụng rất nhiều các công nghệ realtime đặc biệt là WEBSOCKET .

Xử dụng socket.io  giống nhau ở cả client lẫn server nó bao gồm 3 phần chính:

* Khởi tạo kết nối
* Lắng nghe event
* Gửi event
* Vì vậy nhóm đã sử dụng socket.io để xây dựng ứng dụng web realtime



*Hình 5 :client-server trong websocket (socket.io)*

**b) Nổi bật của socket.io:**

- Tính bảo mật : Khi Socket.io xuất hiện, nó sẽ tự động tạo những kết nối bảo mật như:

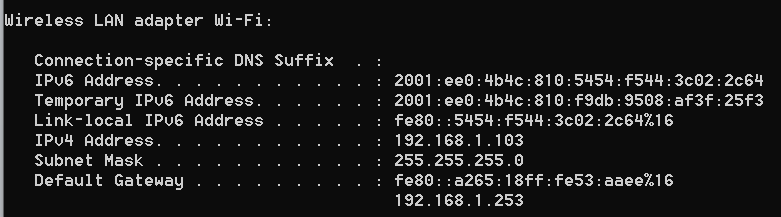
* Proxy và cân bằng tải.
* Tường lửa cá nhân và phần mềm chống vi-rút.

-Tự động kết nối lại, phát hiện ngắt kết nối: Mặc định, trong quá trình chạy, khi client nếu bị ngắt kết nối, nó sẽ tự động gắng kết nối lại mãi mãi cho tới khi server có phản hổi lại, tính năng này có thể được tùy chỉnh nếu muốn. Ngoài ra, **Socket.io** còn cung cấp cho chúng ta các events để phát hiện ngắt kết nối giữa client và server, hoặc ngược lại.

- Hỗ trợ tạo kênh và phòng : **Socket.io**cũng cho phép tạo ra các kênh riêng biệt, từ đó tạo ra mối quan hệ riêng giữa các phần như mỗi module riêng lẻ hoặc dựa trên các quyền khác nhau. Ngoài ra, còn cho phép bạn tạo ra các phòng khác nhau, những clients có thể được tham gia vào phòng khác nhau. Đây là một tính năng hữu ích để gửi thông báo cho một nhóm người dùng hoặc cho một người dùng nhất định được kết nối trên một số thiết bị chẳng hạn

* 1. **IP Address của máy Client:**
* Mỗi máy khi sử dụng mạng đều có địa chỉ IP riêng. Để thiết lập phần mềm ta cần lấy địa chỉ IP của từng máy

Sử dụng lên ipconfig ở trang command



*Hình 6: Thông tin địa chỉ mạng của 1 máy client*

**2.PHÂN TÍCH THIẾT KẾ HỆ THỐNG:**

**a)Giao thức websocket:**

WebSocket là một giao thức giúp truyền dữ liệu hai chiều giữa server-client qua một kết nối TCP duy nhất. Hơn nữa, webSocket là một giao thức được thiết kế để truyền dữ liệu bằng cách sử dụng cổng 80 và cổng 443 và nó là một phần của HTML5. Vì vậy, webSockets có thể hoạt động trên các cổng web tiêu chuẩn, nên không có rắc rối về việc mở cổng cho các ứng dụng, lo lắng về việc bị chặn bởi các tường lửa hay proxy server

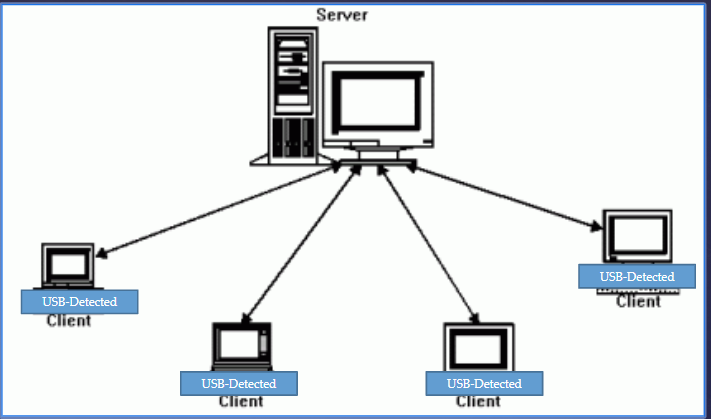
Không giống với giao thức HTTP là cần client chủ động gửi yêu cầu cho server, client sẽ chời đợi để nhận được dữ liệu từ máy chủ. Hay nói cách khác với giao thức Websocket thì server có thể chủ động gửi thông tin đến client mà không cần phải có yêu cầu từ client.

Tất cả dữ liệu giao tiếp giữa client-server sẽ được gửi trực tiếp qua một kết nối cố định làm cho thông tin được gửi đi nhanh chóng và liên tục khi cần thiết. WebSocket làm giảm độ trễ bởi vì một khi kết nối WebSocket được thành lập, server không cần phải chờ đợi cho một yêu cầu từ client.

Tương tự như vậy, client có thể gửi tin nhắn đến server bất cứ lúc nào. Yêu cầu duy nhất này giúp làm giảm đáng kể độ trễ, mà sẽ gửi một yêu cầu trong khoảng thời gian, cho dù thông điệp có sẵn.

Để có thể sử dụng được Websocket thì không phải chỉ cần trình duyệt hỗ trợ mà còn phải có server Websocket, server Websocket có thể được tạo ra bằng bất kỳ ngôn ngữ server-side nào, nhưng Node.js được sử dụng rộng rãi hơn cả vì nó viết bằng Javascript nên mang nhiều ưu điểm so với các ngôn ngữ server-side truyền thống khác.

**b)Hệ thống chương trình :**

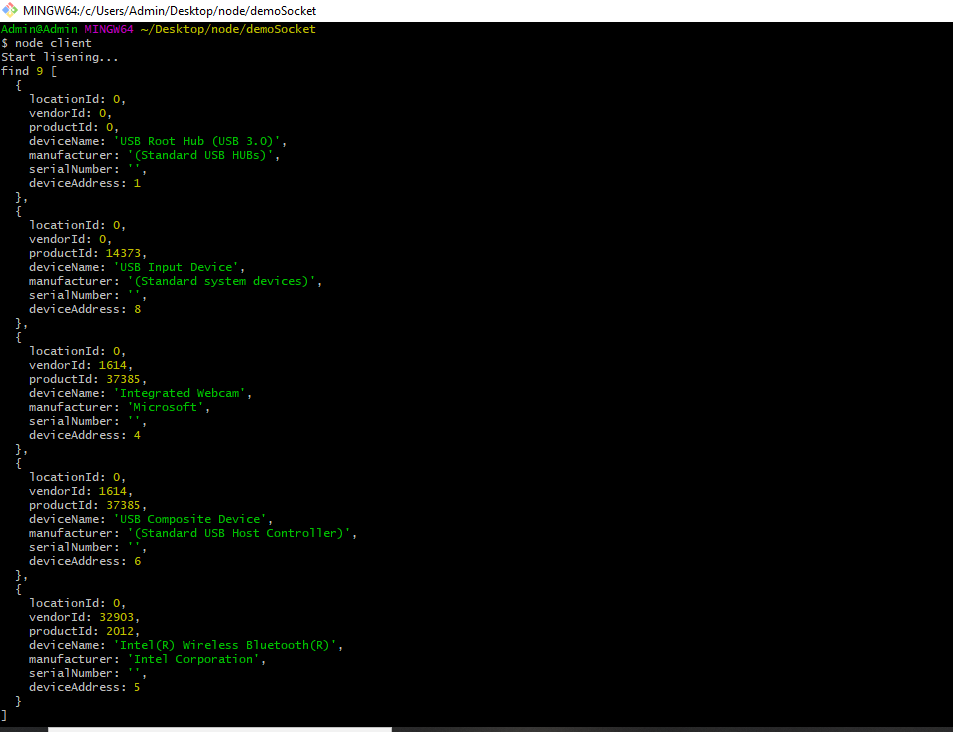


*Hình 7: Sơ đồ khối hệ thống*

* Với mỗi đường truyền bởi một socket khác nhau. Nó được phân biệt và quản lý bởi một socketID
* **Socket.io**cho phép tạo ra các kênh riêng biệt, từ đó tạo ra mối quan hệ riêng giữa các phần như mỗi module riêng lẻ hoặc dựa trên các quyền khác nhau. Từ đó, ta có thể dễ dàng phân định các Client khác nhau. Đây là tính năng quan trọng của đồ án
* **Với mục tiêu là quản lý các Client khác nhau trong hệ thống mạng LAN, việc phân biệt các máy là việc hết sức quan trọng. Do đó, nhờ những tính năng nổi bật của socket.io, ta có thể làm việc đó dễ dàng**

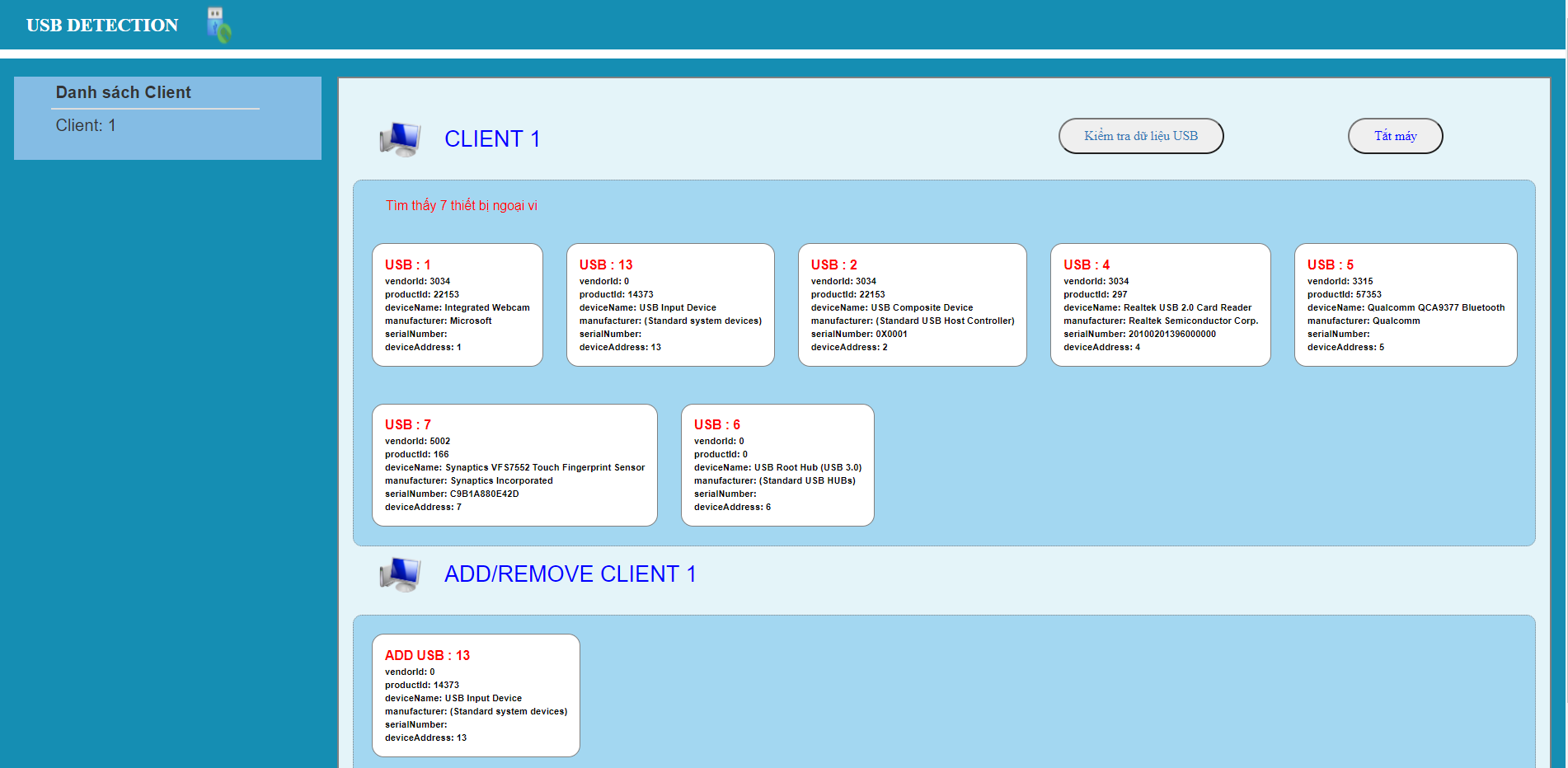
**3.TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ :**

* 1. **Client :**
* Ở phía client, ta chạy ứng dụng USB-Detected. Ứng dụng chạy nền và không xuất hiện giao diện
* Trực tiếp lấy thông tin từng USB mà client đã sử dụng

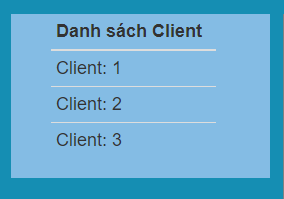


*Hình 8 Ứng dụng usb-detected.exe*

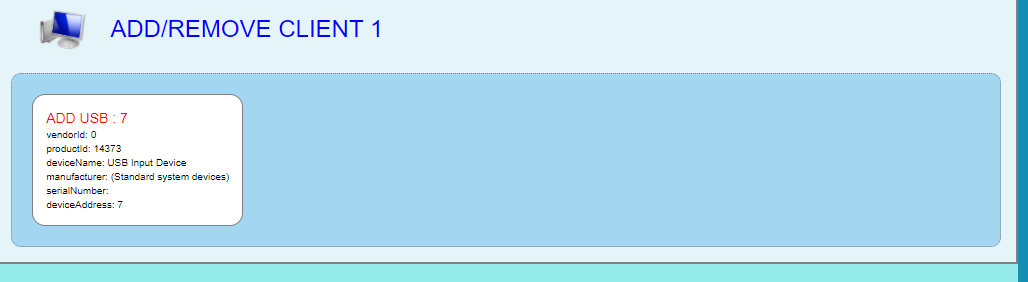
* Ứng dụng chạy trên Command
  1. **Giao diện trình quản lý :**



*Hình 9: Hiển thị thông báo phát hiện thiết bị lưu trữ ngoài*



*Hình 10: Hiền thị danh sách từng Client*



*Hình 11: Các thông số của thiết bị lưu trữ ngoài ở Client*

* Mỗi hành động cắm/rút của Client đều được ghi lại. Bao gồm cả Tên thiết bị, Loại thiết bị và thời gian người ấy thực hiện hành động



*Hình 12: Chức năng dành cho người quản trị*

# KẾT LUẬN CHUNG :

Qua đồ án lần này, nhìn chung nhóm đã có cái nhìn rõ hơn về cách lập trình một hệ thống mạng và hệ điều hành. Đã kiểm thử và thành công trên hệ thống mạng LAN. Thiết kế giao diện dành cho người quản trị. Lắng nghe các sự kiện Thêm/Rút thiết bị và thông báo lên Server tương đối nhanh và không bị lỗi. Đồng thời nhóm cũng đã cải tiến phần mềm với các chức năng như cho phép tắt máy thiết bị và đọc thông tin trong thiết bị lưu trữ của các máy client. Qua đó, người quản trị có thể có thêm nhiều quyền quản lý với các máy trạm của mình hơn, không đơn thuẩn chỉ là việc phát hiện nữa.

# ƯU ĐIỂM, NHƯỢC ĐIỂM :

* Ưu điểm :
* Ứng dụng hiển thị đầy đủ thông tin cho người quản lý
* Dễ dàng cài đặt cho các máy client
* Nhược điểm :
* Vẫn còn ít tính năng dành cho người quản trị
* Hướng phát triển :
* Thêm các tùy chỉnh cho người quản lý như : Gửi thông báo đến Client
* Tạo ứng dụng trên smartphone để người quản lý có thể dễ dàng xem được ứng dụng

# 