|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  **VIỆN ĐIỆN TỬ - VIỄN THÔNG**  logo_128  **ĐỒ ÁN TỐT NGHIỆP**  **Thiết kế phần mềm trên Smart Watch giúp giám sát người dùng**  **BÙI QUANG HƯNG**  hung.bq166244@sis.hust.edu.vn  **Chuyên ngành Điện tử**   |  |  | | --- | --- | | **Giảng viên hướng dẫn:** | PGS. TS. Trịnh Quang Đức  Chữ ký của GVHD | | **Bộ môn:** | Abc abc abc abc abc abc abc abc abc | | **Viện:** | Điện tử - Viễn thông | | **HÀ NỘI, 12/2020** | | |

# LỜI NÓI ĐẦU

Ngày nay, sự phát triển vượt bậc của ngành công nghệ thông tin và điện tử viễn thông đã đem lại những thành tựu to lớn, góp phần vào sự phát triển chung của nhân loại. Những thành tựu đó không chỉ áp dụng trên các lĩnh vực mang tính vĩ mô như khám phá vũ trụ, khám phá các quy luật tự nhiên mà còn trong đời sống hàng ngày của con người. Sự ra đời của hàng loạt những thiết bị thông minh và phần mềm quản lý góp phần cải thiện cuộc sống con người theo chiều hướng tích cực, hiện đại và hiệu quả hơn. Nó len lỏi vào trong từng góc nhỏ của cuộc sống, ứng dụng ngay trong những hoạt động như hỗ trợ con người trong quá trình sản xuất và kiểm soát các hoạt đông của tập thể…

Lập trình trên các thiết bị thông minh hiện cũng đang là xu hướng phát triển của ngành truyền thông và công nghệ thông tin. Với mong muốn được tìm hiểu về lập trình ứng dụng vào đời sống thực tế, em đã chọn đề tài: “Thiết kế phần mềm trên Smart Watch gửi định vị lên website giúp giám sát vị trí người dùng”. Đây là đề tài không mới tuy nhiên nó rất phù hợp với người mới bắt đầu như em, giúp em có những cái nhìn đầu tiên trực quan, cụ thể nhất về cách thức tạo ra một phần mềm phục vụ mong muốn của người dùng. Ngoài ra em cũng có thêm rất nhiều hiểu biết về hệ thống định vị GPS.

Em xin gửi lời cảm ơn chân thành đến thầy Trịnh Quang Đức – giảng viên hướng dẫn, người đã tạo điều kiện, tận tình hướng dẫn và chỉ bảo em trong suốt quá trình nghiên cứu, tìm hiểu và thực hiện đồ án của mình.

Em cũng xin gửi lời cảm ơn đến trường đại học Bách Khoa Hà Nội đã tạo điều kiện cho chúng em học tập. Cảm ơn các thầy cô trong viện Điện tử - Viễn thông đã nhiệt tình giảng dạy, trang bị những kiến thức cơ bản cho chúng em trên con đường phát triển sau này. Cảm ơn các bạn sinh viên lớp CN – Điện tử 1 – K61 đã cùng mình hoàn thành tốt một giai đoạn quan trọng trong cuộc đời.

# LỜI CAM ĐOAN

Tôi là Bùi Quang Hưng, mã số sinh viên 20166244, sinh viên lớp cử nhân Điện tử 1, khóa K61. Người hướng dẫn là TS. Trịnh Quang Đức. Tôi xin cam đoan toàn bộ nội dung được trình bày trong đồ án Thiết kế phần mềm trên Smart Watch gửi định vị lên website giúp giám sát vị trí người dùng là kết quả của quá trình nghiên cứu, tìm hiểu và thực hiện của tôi. Các dữ liệu cũng như phần mã code được viết ra trong đồ án đều là trung thực do tôi tự tay viết và kiểm chứng. Mọi thông tin tham khảo, trích dẫn đều tuân thủ các quy định về sở hữu trí tuệ, các tài liệu tham khảo được liệt kê rõ ràng. Tôi xin chịu hoàn toàn trách nhiệm về các nội dung được viết trong đồ án này.

Hà Nội, ngày 29, tháng 12, năm 2020

**Người cam đoan**

**Bùi Quang Hưng**

# TÓM TẮT ĐỒ ÁN

Với mong muốn được tìm hiểu về lập trình ứng dụng vào đời sống thực tế, em đã chọn đề tài: “Thiết kế phần mềm trên Smart Watch gửi định vị lên website giúp giám sát vị trí người dùng”. Đây là đề tài không mới tuy nhiên nó rất phù hợp với người mới bắt đầu như em, giúp em có những cái nhìn đầu tiên trực quan, cụ thể nhất về cách thức tạo ra một phần mềm phục vụ mong muốn của người dùng. Ngoài ra em cũng có thêm rất nhiều hiểu biết về hệ thống định vị GPS. Ứng dụng này dành cho các phụ huynh hoặc người nhà có thể giám sát được vị trí của con em hoặc cha mẹ mình - những người có năng lực hành vi bị hạn chế hoặc cần kiểm soát, cần theo dõi. Phần mềm được nạp vào đồng hồ thông minh đeo tay. Khi người dùng đeo đồng hồ, phần mềm sẽ định vị vị trí người đeo và gửi lên website nơi người cần giám sát, theo dõi có tài khoản và thông qua đó có thể biết người đeo đang ở đâu. Ngôn ngữ sử dụng để viết phần mềm là Java.

# MỤC LỤC

[LỜI NÓI ĐẦU 2](#_Toc60644931)

[LỜI CAM ĐOAN 3](#_Toc60644932)

[TÓM TẮT ĐỒ ÁN 4](#_Toc60644933)

[MỤC LỤC 5](#_Toc60644934)

[DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT 7](#_Toc60644935)

[DANH MỤC HÌNH VẼ 8](#_Toc60644936)

[DANH MỤC BẢNG BIỂU 9](#_Toc60644937)

[CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI 10](#_Toc60644938)

[1.1 Lập trình phần mềm 10](#_Toc60644939)

[1.1.1 Ngôn ngữ lập trình phần mềm 11](#_Toc60644940)

[1.2 Hệ thống định vị GPS 12](#_Toc60644941)

[1.2.1 GPS là gì? 12](#_Toc60644942)

[1.2.2 Cách hoạt động của GPS 13](#_Toc60644943)

[1.2.3 GPS trên thiết bị di động 14](#_Toc60644944)

[1.3 Lý do chọn đề tài 16](#_Toc60644945)

[1.4 Yêu cầu cần đạt được 16](#_Toc60644946)

[1.5 Ứng dụng vào thực tế 16](#_Toc60644947)

[CHƯƠNG 2. LỰA CHỌN PHƯƠNG ÁN THIẾT KẾ 17](#_Toc60644948)

[2.1 Mô hình triển khai 17](#_Toc60644949)

[2.1.1 Mô hình Client - Server 18](#_Toc60644950)

[2.1.2 Mô hình Client – Server cho ứng dụng 22](#_Toc60644951)

[2.2 Thiết kế phần mềm trên Smart Watch gửi định vị lên web 22](#_Toc60644952)

[2.2.1 Database lưu trữ dữ liệu 22](#_Toc60644953)

[2.2.2 Lập trình ứng dụng trên Smart Watch 25](#_Toc60644954)

[2.2.3 Phần mềm hỗ trợ lập trình Android: Android Studio 26](#_Toc60644955)

[2.2.4 Gửi định vị lên web 27](#_Toc60644956)

[2.3 Giao diện web cho người quản lý 29](#_Toc60644957)

[2.3.1 Giao diện 29](#_Toc60644958)

[CHƯƠNG 3. TRIỂN KHAI PHẦN MỀM 30](#_Toc60644959)

[3.1 Thiết kế database lưu trữ dữ liệu 30](#_Toc60644960)

[3.2 Thiết kế giao diện phần mềm 30](#_Toc60644961)

[3.3 Triển khai code phần server xử lí và code phần app 31](#_Toc60644962)

[3.4 Triển khai code giao diện web 31](#_Toc60644963)

[3.5 Kiểm thử phần mềm: 31](#_Toc60644964)

[CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC 31](#_Toc60644965)

[CHƯƠNG 5. KẾT LUẬN 32](#_Toc60644966)

# DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

# DANH MỤC HÌNH VẼ

[Hình 1.1 Cách hoạt động của GPS 14](#_Toc60642924)

[Hình 2.1 Mô hình Client - Server 17](#_Toc60642925)

# DANH MỤC BẢNG BIỂU

# CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

## 1.1 Lập trình phần mềm

Lập trình là cụm từ rất phổ biến trong xã hội hiện nay, rất quen thuộc đối với sinh viên các ngành công nghệ thông tin và điện tử. Một cách tổng quát đó là việc thiết lập một thứ gì đó làm việc theo trình tự để hoàn thành một công việc hay một mục đích nào đó đã biết trước. Ví du khi bạn tới trường, bạn lấy xe máy ra khỏi nhà, sau đó bạn khóa cửa và chạy xe tới trường, hay khi bạn đói bạn tìm đồ ăn, khi bạn khát bạn tìm nước uống… Đây có thể xem là các hoạt động lập trình đơn giản trong đầu bạn. Trong lập trình có một lĩnh vực gọi là lập trình phần mềm. Tuy nhiên, ngày nay mọi người hay đồng nhất 2 cụm từ lập trình và lập trình phần mềm với nhau, chung ta cần làm rõ vấn đề này.

Lập trình phần mềm là việc các lập trình viên sử dụng các ngôn ngữ lập trình và phần mềm hỗ trợ viết ra những đoạn code theo một trình tự để tạo ra những phần mềm/ứng dụng chạy trên các thiết bị như máy tính, điện thoại... nhằm đáp ứng một nhu cầu nào đó của con người: giải trí, học tập, trao đổi thông tin…

Ví dụ lập trình phần mềm có thể làm một số việc như:

1. Thiết lập bóng đèn trong nhà tự bật khi có người trong nhà và buổi tối, tự tắt khi trời sáng hoặc không có người ở nhà.
2. Điều khiển máy bơm tự động tới vườn khi đất khô hay trời nắng nóng.
3. Mọi hệ điều hành, phần mềm trên máy tính, điện thoại đều là sản phẩm của việc lập trình.
4. Giúp mọi người chat với nhau: facebook, Zalo...
5. Làm các game giải trí như: Pubg, liên minh huyền thoại, Dota...

Trong đồ án này, mục đích thiết kế phần mềm của em là kiểm soát, giám sát vị trí của người cần giám sát. Đối tượng giám sát là con em, trẻ nhỏ hoặc người già bị hạn chế năng lực hành vi. Như vậy em cần kết hợp giữa tư duy giải quyết vấn đề với lập trình phần mềm để thiết kế ra phần mềm thỏa mãn nhu cầu trên.

### 1.1.1 Ngôn ngữ lập trình phần mềm

Ngôn ngữ lập trình là một dạng ngôn ngữ máy tính, được dùng để viết ra phần mềm, tập lệnh hoặc các hướng dẫn cho máy tính làm theo. Hiện nay có rất nhiều ngôn ngữ lập trình đang được sử dụng. Mặc dù các ngôn ngữ cũng có điểm chung tương đồng nhưng mỗi ngôn ngữ lại có các cú pháp sử dụng riêng. Công việc của các lập trình viên là họ phải học các quy tắc, cú pháp và cấu trúc ngôn ngữ rồi thực hiện viết mã nguồn trong một trình soạn thảo hoặc IDE và biên dịch code thành ngôn ngữ máy để máy tính có thể hiểu được. Các ngôn ngữ script không yêu cầu trình biên dịch mà sử dụng các trình thông dịch để thực thi script.

Một số ngôn ngữ lập trình phần mềm thông dụng hiện nay có thể kể đến là: C, C++, C#, Java, Python, Javascript… Tùy vào từng lĩnh vực sẽ có các ngôn ngữ lập trình phần mềm riêng. Ví dụ

* Các ứng dụng và chương trình máy tính là những thứ bạn sử dụng để làm việc, học tập, giải trí hằng ngày như: trình duyệt Internet… Các ngôn ngữ thuộc lĩnh vực này có thể kể đến như: C, C#, C++, Java,Visual Basic…
* **Lĩnh vực Trí tuệ nhân tạo**, hay hướng tới tạo ra các nhân vật có thể tương tác trong các trò chơi máy tính, các chương trình đưa ra quyết định, chatbot, phát triển AI: AIML, C, C++, C#, Python…
* Các ngôn ngữ dành cho các nhà phát triển, nghiên cứu, duy trì cơ sở dữ liệu: DBASE, FoxPro, MySQL, SQL, Visual FoxPro…
* **Phát triển trò chơi liên quan đến việc tạo trò chơi trên máy tính hoặc phần mềm giải trí: Java, C++, C, C#...**
* Tạo các trang web, ứng dụng Internet hoặc các tác vụ khác liên quan đến Internet: HDML, HTML, PHP, Python, XML, Java, Java script…

## 1.2 Hệ thống định vị GPS

### 1.2.1 GPS là gì?

Hệ thống Định vị Toàn cầu GPS (*Global Positioning System*) là hệ thống xác định vị trí dựa trên vị trí của các [vệ tinh nhân tạo](https://vi.wikipedia.org/wiki/V%E1%BB%87_tinh) do [Bộ Quốc phòng Hoa Kỳ](https://vi.wikipedia.org/wiki/B%E1%BB%99_Qu%E1%BB%91c_ph%C3%B2ng_Hoa_K%E1%BB%B3) thiết kế, xây dựng, vận hành và quản lý. Trong cùng một thời điểm, tọa độ của một điểm trên mặt đất sẽ được xác định nếu xác định được khoảng cách từ điểm đó đến ít nhất ba vệ tinh. GPS ban đầu chỉ dành cho các mục đích quân sự nhưng từ năm 1980 chính phủ Mỹ đã cho phép sử dụng GPS trong dân sự. GPS hoạt động trong mọi điều kiện thời tiết, mọi nơi trên [Trái Đất](https://vi.wikipedia.org/wiki/Tr%C3%A1i_%C4%90%E1%BA%A5t), 24 giờ một ngày và không mất phí thuê bao hoặc mất tiền trả cho việc thiết lập sử dụng GPS. Tuy nhiên việc mua thiết bị thu tín hiệu và phần mềm nhúng hỗ trợ để sử dụng GPS cũng tiêu tốn một nguồn chi phí không hề nhỏ

Diagram

Description automatically generated

Hình:

GPS là mạng lưới bao gồm 27 vệ tinh quay xung quanh trái đất. Trong số 27 vệ tinh này, 24 vệ tinh đang hoạt động, 3 vệ tinh còn lại đóng vai trò dự phòng trong trường hợp 1 trong số 24 vệ tinh chính bị hư hỏng. Dựa vào cách sắp đặt của các vệ tinh này, khi đứng dưới mặt đất, bạn có thể nhìn được ít nhất là 4 vệ tinh trên bầu trời tại bất kì thời điểm nào.

Phối hợp hoạt động với các vệ tinh quay xung quanh trái đất là 5 trạm theo dõi đặt trên mặt đất: trạm chủ được đặt tại Colorado (Mỹ) và 4 trạm khác (không có người điều khiển) được đặt tại các vị trí rất xa lạ, song lại rất gần với đường xích đạo (trong đó có Hawaii cũng ở Mỹ). Các trạm theo dõi này thu thập dữ liệu từ các vệ tinh và truyền dữ liệu về trạm chủ. Trạm chủ sau đó sẽ xử lý dữ liệu và đưa ra các thay đổi cần thiết để chuyển dữ liệu chuẩn về các vệ tinh GPS.

### 1.2.2 Cách hoạt động của GPS

Các [vệ tinh](https://vi.wikipedia.org/wiki/V%E1%BB%87_tinh) GPS bay vòng quanh Trái Đất hai lần trong một ngày theo một quỹ đạo rất chính xác và phát tín hiệu có thông tin xuống Trái Đất. Các [máy thu GPS](https://vi.wikipedia.org/wiki/M%C3%A1y_thu_GPS) nhận thông tin này và bằng [phép tính lượng giác](https://vi.wikipedia.org/w/index.php?title=Ph%C3%A9p_t%C3%ADnh_l%C6%B0%E1%BB%A3ng_gi%C3%A1c&action=edit&redlink=1) tính được chính xác vị trí của người dùng. Về bản chất máy thu GPS so sánh thời gian tín hiệu được phát đi từ vệ tinh với thời gian nhận được chúng. Sai lệch về thời gian cho biết máy thu GPS ở cách vệ tinh bao xa. Rồi với nhiều quãng cách đo được tới nhiều vệ tinh máy thu có thể tính được vị trí của người dùng và hiển thị lên bản đồ điện tử của máy.

Máy thu phải nhận được tín hiệu của ít nhất ba vệ tinh để tính ra vị trí hai chiều ([kinh độ](https://vi.wikipedia.org/wiki/Kinh_%C4%91%E1%BB%99) và [vĩ độ](https://vi.wikipedia.org/wiki/V%C4%A9_%C4%91%E1%BB%99)) và để theo dõi được chuyển động. Khi nhận được tín hiệu của ít nhất 4 vệ tinh thì máy thu có thể tính được vị trí ba chiều (kinh độ, vĩ độ và [độ cao](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%99_cao)). Một khi vị trí người dùng đã tính được thì máy thu GPS có thể tính các thông tin khác, như tốc độ, hướng chuyển động, bám sát di chuyển, khoảng hành trình, quãng cách tới điểm đến, thời gian [Mặt Trời](https://vi.wikipedia.org/wiki/M%E1%BA%B7t_Tr%E1%BB%9Di) mọc, lặn và nhiều thứ khác nữa.

### 1.2.3 GPS trên thiết bị di động

[Thiết bị SmartPhone](http://www.thegioididong.com/dtdd?trang=1) hoặc SmartWatch (Android, Windows Phone, iOS) sẽ là một đầu thu GPS, nó sẽ thu dữ liệu từ các vệ tinh GPS ở trên bầu trời. Vậy dữ liệu từ các vệ tinh là gì? Nói một cách đơn giản, mỗi vệ tinh cho ta biết khoảng cách chính xác từ vị trí của mình đến vệ tinh đó hoặc một điểm nào đó trên trái đất.

Cơ chế hoạt động của GPS rất đơn giản, chúng ta có thể tưởng tượng như sau. Trên bản đồ có 3 điểm cố định A, B, C. Dữ liệu GPS cho ta biết khoảng cách lần lượt từ điểm A, B, C đến nơi ta đứng lần lượt là: 1km, 3km và 2km.

Diagram

Description automatically generated

Hình 1.1 Cách hoạt động của GPS

Sau đó ta vẽ 3 đường tròn tâm là A, B, C với bán kính lần lượt là 1km, 2km và 3km, vị trí giao nhau của 3 đường tròn chính là vị trí của ta trên bản đồ

Thiết bị SmartPhone hoặc SmartWatch phải nhận được tín hiệu của ít nhất ba vệ tinh để cho ra vị trí hai chiều và theo dõi được chuyển động của ta. Khi nhận được tín hiệu của ít nhất bốn vệ tinh, máy sẽ cho ra được vị trí ba chiều. Một khi vị trí của ta đã tính được thì thiết bị SmartPhone hay SmartWatch (Android, [Windows Phone](http://www.thegioididong.com/game-ung-dung/windowsphone), iOS) có thể tính các thông tin khác, như tốc độ di chuyển, hướng chuyển động, bám sát di chuyển, khoảng hành trình, khoảng cách đích đến và nhiều thứ khác nữa.

Ví dụ nói trên mô tả một cách đơn giản về cách hoạt động của GPS: mỗi vệ tinh có thể được coi là một người bạn của bạn. Tuy vậy, các vệ tinh hoạt động trên không gian ba chiều, do đó bạn sẽ phải tưởng tượng ra các hình cầu, thay vì các hình tròn. Sử dụng các dữ liệu về khoảng cách tới các vệ tinh, kết hợp với việc bạn chắc chắn đang ở trên bề mặt trái đất, việc tìm ra vị trí chính xác của bạn trở nên dễ dàng.

Để đưa ra vị trí chính xác, rất nhiều thiết bị GPS kết nối tới ít nhất là 4 vệ tinh. Đó là lý do vì sao đôi khi để tìm ra vị trí chính xác của bạn, hệ thống GPS lại mất nhiều thời gian tới vậy. Đó cũng là lý do vì sao đôi khi chúng ta bị mất sóng GPS: do thiết bị có thể đã kết nối tới 1 hoặc 2 vệ tinh, song 2 vệ tinh vẫn là không đủ

Đằng sau cách hoạt động của GPS còn rất nhiều thuật toán cũng như các ứng dụng khoa học khác. Ví dụ, ta sẽ phải tính đến mức độ trễ của tín hiệu giữa vệ tinh và các đầu nhận GPS. Ngoài ra, vì thiết bị di động phải liên tục tính toán để sửa chữa các sai sót trong dữ liệu về vị trí nên tiêu tốn nhiều pin

## 1.3 Lý do chọn đề tài

Với mong muốn được tìm hiểu về lập trình ứng dụng vào đời sống thực tế, em đã chọn đề tài: “Thiết kế phần mềm trên Smart Watch gửi định vị lên website giúp giám sát vị trí người dùng”. Đây là đề tài không mới tuy nhiên nó rất phù hợp với người mới bắt đầu như em, giúp em có những cái nhìn đầu tiên trực quan, cụ thể nhất về cách thức tạo ra một phần mềm phục vụ mong muốn của người dùng. Ngoài ra em cũng có thêm rất nhiều hiểu biết về hệ thống định vị GPS để có thể phục vụ vào công việc sau này.

## 1.4 Yêu cầu cần đạt được

Tuy đây là một phần mềm không quá phức tạp nhưng các yêu cầu kỹ thuật đặt ra là:

1. Chạy được trên các loại đồng hồ thông minh đơn giản hiện nay
2. Đảm bảo tín hiệu được gửi và nhận liên tục để đảm bảo tính chính xác của vị trí người đeo
3. Giao diện phía người giám sát đơn giản, dễ dùng nhưng vẫn đảm bảo tính bảo mật thông tin cho người được giám sát. Phân quyền phía người giám sát

## 1.5 Ứng dụng vào thực tế

Ứng dụng này dành cho các phụ huynh hoặc người nhà có thể giám sát được vị trí của con em hoặc cha mẹ mình - những người có năng lực hành vi bị hạn chế hoặc cần kiểm soát, cần theo dõi. Phần mềm được nạp vào đồng hồ thông minh đeo tay. Khi người dùng đeo đồng hồ, phần mềm sẽ định vị vị trí người đeo và gửi lên website nơi người cần giám sát, theo dõi có tài khoản và thông qua đó có thể biết người đeo đang ở đâu.

# CHƯƠNG 2. LỰA CHỌN PHƯƠNG ÁN THIẾT KẾ

## 2.1 Mô hình triển khai

Về mô hình project sẽ triển khai theo mô hình client-server.

### 2.1.1 Mô hình Client - Server

Mô hình client-server là một mô hình nổi tiếng trong [mạng máy tính](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_m%C3%A1y_t%C3%ADnh), được áp dụng rất rộng rãi và là mô hình của mọi trang web hiện có. Ý tưởng của mô hình này là máy con (đóng vài trò là máy khách) gửi một yêu cầu (request) để máy chủ (đóng vai trò người cung ứng dịch vụ), máy chủ sẽ xử lý và trả kết quả về cho máy khách.

Diagram

Description automatically generated

Hình 2.1 Mô hình Client - Server

Mô hình Client - Server cho phép mạng tập trung các ứng dụng và chức năng tại một hoặc nhiều máy dịch vụ file chuyên dụng. Các máy này trở thành trung tâm của hệ thống. Hệ điều hành của Client - Server cho phép người dùng chia sẻ đồng thời cùng một tài nguyên, không quan trọng vị trí địa lý.

Mô hình client/server như sau: Client/Server là mô hình tổng quát nhất, trên thực tế thì một server có thể được nối tới nhiều server khác nhằm làm việc hiệu quả và nhanh hơn. Khi nhận được 1 yêu cầu từ client, server này có thể gửi tiếp yêu cầu vừa nhận được cho server khác ví dụ như database server vì bản thân nó không thể xử lý yêu cầu này được. Máy server có thể thi hành các nhiệm vụ đơn giản hoặc phức tạp. Ví dụ như một máy chủ trả lời thời gian hiện tại trong ngày, khi một máy client yêu cầu lấy thông tin về thời gian nó sẽ phải gửi một yêu cầu theo một tiêu chuẩn do server định ra, nếu yêu cầu được chấp nhận thì máy server sẽ trả về thông tin mà client yêu cầu. Có rất nhiều các dịch vụ server trên mạng nhưng nó đều hoạt động theo nguyên lý là nhận các yêu cầu từ client sau đó xử lý và trả kết quả cho client yêu cầu. Thông thường chương trình server và client được thi hành trên hai máy khác nhau. Cho dù lúc nào server cũng ở trạng thái sẵn sàng chờ nhận yêu cầu từ client nhưng trên thực tế một tiến trình liên tác qua lại (interaction) giữa client và server lại bắt đầu ở phía client, khi mà client gửi tín hiệu yêu cầu tới server. Các chương trình server thường đều thi hành ở mức ứng dụng (tầng ứng dụng của mạng). Sự thuận lợi của phương pháp này là nó có thể làm việc trên bất cứ một mạng máy tính nào có hỗ trợ giao thức truyền thông chuẩn cụ thể ở đây là giao thức TCP/IP. Với các giao thức chuẩn này cũng giúp cho các nhà sản xuất có thể tích hợp nhiều sản phẩm khác nhau của họ lên mạng mà không gặp phải khó khăn gì. Với các chuẩn này thì các chương trình server cho một dịch vụ nào đấy có thể thi hành trên một hệ thống chia sẻ thời gian (timesharing system) với nhiều chương trình và dịch vụ khác hoặc nó có thể chạy trên chính một máy tính các nhân bình thường. Có thể có nhiều chương server cùng làm một dịch vụ, chúng có thể nằm trên nhiều máy tính hoặc một máy tính. Với mô hình trên chúng ta nhận thấy rằng mô hình client/server chỉ mang đặc điểm của phần mềm không liên quan gì đến phần cứng mặc dù trên thực tế yêu cầu cho một máy server là cao hơn nhiều so với máy client. Lý do là bởi vì máy server phải quản lý rất nhiều các yêu cầu từ các clients khác nhau trên mạng. Ưu và nhược điểm chính Có thể nói rằng với mô hình client/server thì mọi thứ dường như đều nằm trên bàn của người sử dụng, nó có thể truy cập dữ liệu từ xa (bao gồm các công việc như gửi và nhận file, tìm kiếm thông tin...) với nhiều dịch vụ đa dạng mà mô hình cũ không thể làm được. Mô hình client/server cung cấp một nền tảng lý tưởng cho phép tích hợp các kỹ thuật hiện đại như mô hình thiết kế hướng đối tượng, hệ chuyên gia, hệ thông tin địa lý (GIS)... Một trong những vấn đề nảy sinh trong mô hình này đó là tính an toàn và bảo mật thông tin trên mạng. Do phải trao đổi dữ liệu giữa 2 máy ở 2 khu vực khác nhau cho nên dễ dàng xảy ra hiện tượng thông tin truyền trên mạng bị lộ.

**Client** trong mô hình client/server, người ta còn định nghĩa cụ thể cho một máy client là một máy trạm mà chỉ được sử dụng bởi 1 người dùng với để muốn thể hiện tính độc lập cho nó. Máy client có thể sử dụng các hệ điều hành bình thường như Win9x, DOS, OS/2... Bản thân mỗi một client cũng đã được tích hợp nhiều chức năng trên hệ điều hành mà nó chạy, nhưng khi được nối vào một mạng LAN, WAN theo mô hình client/server thì nó còn có thể sử dụng thêm các chức năng do hệ điều hành mạng (NOS) cung cấp với nhiều dịch vụ khác nhau (cụ thể là các dịch vụ do các server trên mạng này cung cấp), ví dụ như nó có thể yêu cầu lấy dữ liệu từ một server hay gửi dữ liệu lên server đó... Thực tế trong các ứng dụng của mô hình client/server, các chức năng hoạt động chính là sự kết hợp giữa client và server với sự chia sẻ tài nguyên, dữ liệu trên cả hai máy Vai trò của client Trong mô hình client/server, client được coi như là người sử dụng các dịch vụ trên mạng do một hoặc nhiều máy chủ cung cấp và server được coi như là người cung cấp dịch vụ để trả lời các yêu cầu của các clients. Điều quan trọng là phải hiểu được vai trò hoạt động của nó trong một mô hình cụ thể, một máy client trong mô hình này lại có thể là server trong một mô hình khác. Ví dụ cụ thể như một máy trạm làm việc như một client bình thường trong mạng LAN nhưng đồng thời nó có thể đóng vai trò như một máy in chủ (printer server) cung cấp dịch vụ in ấn từ xa cho nhiều người khác (clients) sử dụng. Client được hiểu như là bề nổi của các dịch vụ trên mạng, nếu có thông tin vào hoặc ra thì chúng sẽ được hiển thị trên máy client.

**Server Server** còn được định nghĩa như là một máy tính nhiều người sử dụng (multiuser computer). Vì một server phải quản lý nhiều yêu cầu từ các client trên mạng cho nên nó hoạt động sẽ tốt hơn nếu hệ điều hành của nó là đa nhiệm với các tính năng hoạt động độc lập song song với nhau như hệ điều hành UNIX, WINDOWS... Server cung cấp và điều khiển các tiến trình truy cập vào tài nguyên của hệ thống. Các ứng dụng chạy trên server phải được tách rời nhau để một lỗi của ứng dụng này không làm hỏng ứng dụng khác. Tính đa nhiệm đảm bảo một tiến trình không sử dụng toàn bộ tài nguyên hệ thống. Vai trò của server. Như chúng ta đã bàn ở trên, server như là một nhà cung cấp dịch vụ cho các clients yêu cầu tới khi cần, các dịch vụ như cơ sở dữ liệu, in ấn, truyền file, hệ thống... Các ứng dụng server cung cấp các dịch vụ mang tính chức năng để hỗ trợ cho các hoạt động trên các máy clients có hiệu quả hơn. Sự hỗ trợ của các dịch vụ này có thể là toàn bộ hoặc chỉ một phần thông qua IPC. Để đảm bảo tính an toàn trên mạng cho nên server này còn có vai trò như là một nhà quản lý toàn bộ quyền truy cập dữ liệu của các máy clients, nói cách khác đó là vai trò quản trị mạng. Có rất nhiều cách thức hiện nay nhằm quản trị có hiệu quả, một trong những cách đang được sử dụng đó là dùng tên Login và mật khẩu

**Ưu điểm của mô hình Client - Server**

* Client server có khả năng chống quá tải mạng
* Client server đảm bảo toàn vẹn dữ liệu khi có sự cố xảy ra
* Dễ dàng mở rộng hệ thống mạng
* Chỉ cần chung định dạng giao tiếp mà không cần chung nền tảng là có thể hoạt động được
* Client server cho phép tích hợp các kỹ thuật hiện đại như GIS, mô hình thiết kế hướng đối tượng…
* Với mô hình Client server, người dùng có thể truy cập dữ liệu từ xa, thực hiện các thao tác gửi, nhận file hay tìm kiếm thông tin đơn giản

Ví dụ:

1. Web server: Người dùng nhập địa chỉ website => client gửi yêu cầu đến web server => web server gửi toàn bộ nội dung của website về cho client
2. Mail server: Người dùng soạn mail => client gửi đến mail server => mail server tiếp nhận, lưu trữ, tìm kiếm địa chỉ email được gửi đến & gửi mail đi
3. File server: File server nhận tập tin từ phía client, lưu trữ và truyền tập tin đi. Người dùng có thể upload, download các tập tin lên server qua giao thức FTP hay web browser

**Nhược điểm của mô hình Client - Sever**

Bên cạnh những ưu điểm kể trên, Client server vẫn còn tồn tại một số nhược điểm sau:

* Cần bảo trì, bảo dưỡng server thường xuyên.
* Khả năng bảo mật thông tin mạng là một hạn chế nữa của Client server. Bởi vì, nguyên lý hoạt động của Client server là trao đổi dữ liệu giữa server và client ở 2 khu vực địa lý khác nhau. Trong quá trình trao đổi dữ liệu, khả năng thông tin mạng bị lộ là điều dễ xảy ra.

### 2.1.2 Mô hình Client – Server cho ứng dụng

Áp dụng vào thực tế của đồ án này em sẽ thiết kế mô hình Client – Server như sau

Phần Client:

* Phần mềm lấy định vị trên Smart Watch: Ứng dụng này sẽ lấy định vị thông qua GPS gửi lên Server bằng các giao thức truyền dữ liệu
* Giao diện kiểm soát, giám sát vị trí trên web

Phần Server:

* Lưu trữ thông tin gửi lên từ phía client vào database.
* Chuyển thông tin để hiển thị lên Google Map khi người dùng web yêu cầu.

## 2.2 Thiết kế phần mềm trên Smart Watch gửi định vị lên web

### 2.2.1 Database lưu trữ dữ liệu

Như ta đã biết, Database (cơ sở dữ liệu) là một tập hợp những data (dữ liệu) có liên quan đến nhau. Database được duy trì dưới dạng một tập hợp các tập tin trong hệ điều hành hay được lưu trữ trong các hệ quản trị cơ sở dữ liệu.

Database có vai trò vô cùng quan trọng khi làm việc với hệ thống dữ liệu. Chúng giúp người dùng thành công trong việc kết nối các dữ liệu. Người dùng có thể truy cập hệ cơ sở dữ liệu nhanh chóng và dễ dàng hơn. Database chính là cơ sở nguồn để người dùng có thể truy xuất ra những thông tin cần thiết.

  Đặc điểm chính của Database chính là truy xuất ra những thông tin, dữ liệu bằng nhiều phương thức khác nhau. Các nội dung truy xuất được đảm bảo toàn vẹn dữ liệu ở mức độ cao. Đồng thời, nguồn thông tin khi xuất ra hoàn toàn không bị trùng lặp, nếu có thì xác suất cũng rất thấp. Một cơ sở dữ liệu Database cho phép nhiều người dùng đồng thời truy cập trong cùng một thời gian.

**Ứng dụng của Database**

Hệ thống cơ sở dữ liệu đóng vai trò vô cùng quan trong trong thời đại thông tin – kỹ thuật như hiện nay. Chức năng chính của Database có thể kể đến là: lưu trữ, truy xuất và cập nhật dữ liệu cùng nhiều ứng dụng khác.

Database giúp quản lý các dịch vụ bảo mật và phục hồi hệ thống quản trị dữ liệu, giúp thực thi các ràng buộc bên trong hệ cơ sở này. Đồng thời, quản lý và kiểm soát tất cả các máy khách kết nối, truy cập vào hệ thống dữ liệu ở hệ thống nguồn. Hỗ trợ xử lý tất cả các truy cập dữ liệu và các chức năng điều khiển khác.

Ngoài ra, Database còn cung cấp tính năng kiểm soát đồng thời, giúp bảo mật chặt chẽ hơn. Tạo môi trường đa người dùng với điều kiện kết nối an toàn. Nhiều người có thể truy cập đồng thời và truy xuất được những dữ liệu cần thiết

**Các mô hình Database phổ biến hiện nay**

* Cơ sở dữ liệu Database dạng File: Database dạng File là dạng thường gặp nhất và cũng là mô hình phổ biến nhất hiện nay. Có thể dễ dàng thấy những tập tin, những thông tin được lưu trữ, chia sẻ và truy cập dưới dạng File. Cơ sở dữ liệu dạng File thường gặp nhất chính là \*.mdb Foxpro. Bên cạnh đó, cơ sở dữ liệu này còn bao gồm những định dạng khác như text, dạng file ascii, dạng file \*.dbf…
* Cơ sở dữ liệu Database quan hệ: Đây cũng là một trong những mô hình dữ liệu được sử dụng rộng rãi nhất hiện nay. Ở mô hình này, các thực thể khác nhau, tức các nguồn dữ liệu khác nhau sẽ cùng được quy hợp, lưu trữ cùng một bảng dữ liệu và các dữ liệu này phải có quan hệ với nhau. Một vài cơ sở dữ liệu nổi tiếng có hỗ trợ Database quan hệ có thể kể đến như: Oracle, MS SQL Server, MySQL…
* Cơ sở dữ liệu hướng đối tượng: Hệ cơ sở dữ liệu này cũng là một bảng dữ liệu thuần. Tuy nhiên, Database bổ sung thêm các trường hợp hướng đối tượng khác như: hành vi đối tượng nhằm thể hiện hành vi của đối tượng. Các đối tượng này cũng được phân cấp rất rõ ràng, mỗi cấp được gọi là một lớp dữ liệu. Hiểu một cách đơn giản hơn, chúng là tập hợp các nhóm đối tượng trong cùng một bảng và được thể hiện bằng dòng dữ liệu. Những hệ quản trị cơ sở dữ liệu hỗ trợ Database này bao gồm: MS SQL Server, Postgres, Oracle
* Database bán cấu trúc: Cơ sở dữ liệu Database bán cấu trúc có thể lưu trữ nhiều dạng dữ liệu khác nhau. Và các thông tin, dữ liệu này thường được lưu trữ dưới định dạng XML. Ở mô hình này, các thông tin mô tả dữ liệu và đối tượng đều được trình bày trong các thẻ tag. Nhờ vào ưu điểm nổi bật này, hệ cơ sở dữ liệu vừa được mở rộng, vừa tiện lợi giúp người dùng có thể dễ dàng truy xuất những thông tin cần thiết. Theo các chuyên gia nghiên cứu, Database bán cấu trúc có thể sẽ là hướng đi mới đầy tiềm năng trong các lĩnh vực ứng dụng công nghệ dữ liệu.

**Lựa chọn cơ sở dữ liệu**

Dựa vào yêu cầu của đề tài, em đã cân nhắc giữa việc sử dụng SQL và NoSQL để lưu dữ liệu. Em nghĩ sử dụng SQL lưu dữ liệu theo dạng bảng không phải là giải pháp tốt trong quá trình phát triển.Lí do đó là MongoDB cho phép xây dựng ứng dụng nhanh ,xử lí dữ liệu đa dạng linh hoạt .Việc phát triển được đơn giản hóa cho các ngôn ngữ hiện đại và loại bỏ lớp ánh xạ đối tượng phức tạp ORM.Hơn thế nữa, MongoDB cũng có thể được mở rộng trong và giữa nhiều trung tâm dữ liệu phân tán, cung cấp các cấp độ sẵn có và khả năng mở rộng mới,

[Cơ](https://vi.wikipedia.org/wiki/NoSQL) sở dữ liệu [NoSQL](https://vi.wikipedia.org/wiki/NoSQL) thường rất nhanh, không yêu cầu lược đồ bảng cố định, tránh các hoạt động tham gia bằng cách lưu trữ dữ liệu không chuẩn hóa và được thiết kế để mở rộng theo chiều ngang. Thông tin truy xuất từ Database sẽ sử dụng API phía Server

Cơ sở dữ liệu NoSQL là lựa chọn cực kỳ thích hợp cho nhiều ứng dụng hiện đại, ví dụ như di động, web và trò chơi đòi hỏi phải sử dụng cơ sở dữ liệu cực kỳ thiết thực, linh hoạt, có khả năng thay đổi quy mô và hiệu năng cao để đem đến cho người dùng trải nghiệm tuyệt vời.

* Linh hoạt: Cơ sở dữ liệu NoSQL thường cung cấp các sơ đồ linh hoạt giúp công đoạn phát triển nhanh hơn và có khả năng lặp lại cao hơn. Mô hình dữ liệu linh hoạt biến cơ sở dữ liệu NoSQL thành lựa chọn lý tưởng cho dữ liệu không được tổ chức thành cấu trúc hoặc có cấu trúc chưa hoàn chỉnh.
* Khả năng thay đổi quy mô: Cơ sở dữ liệu NoSQL thường được thiết kế để tăng quy mô bằng cách sử dụng các cụm phần cứng được phân phối thay vì tăng quy mô bằng cách bổ sung máy chủ mạnh và tốn kém. Một số nhà cung cấp dịch vụ đám mây xử lý các hoạt động này một cách không công khai dưới dạng dịch vụ được quản lý đầy đủ.
* Hiệu năng cao: Cơ sở dữ liệu NoSQL được tối ưu hóa theo các mô hình dữ liệu cụ thể và các mẫu truy cập giúp tăng hiệu năng cao hơn so với việc cố gắng đạt được mức độ chức năng tương tự bằng cơ sở dữ liệu quan hệ.
* Cực kỳ thiết thực: Cơ sở dữ liệu NoSQL cung cấp các API và kiểu dữ liệu cực kỳ thiết thực được xây dựng riêng cho từng mô hình dữ liệu tương ứng.

### 2.2.2 Lập trình ứng dụng trên Smart Watch

Hiện nay có 2 nền tảng chính hỗ trợ lập trình ứng dụng trên thiết bị di động là Android và iOS. Sau khi tìm hiểu, cân nhắc em đã lựa chọn nền tảng Android để triển khai phần ứng dụng trên Smart Watch bởi một số lý do sau:

Như chúng ta đã biết, hiện này có đến hơn 1 nửa nhân loại sử dụng các thiết bị thông minh như điện thoại để giao tiếp với nhau qua các mạng không dây. Số người dùng này sẽ ngày càng tăng lên và thiết bị sẽ ngày càng thông minh với nhiều chức năng và dịch vụ hấp dẫn. Vì thế, việc lập trình trên các thiết bị như điện thoại, đồng hồ ngày càng phổ biến và phát triển mạnh mẽ. Từ nền tảng mã nguồn mở, Google đã cho ra mắt Android chạy trên các thiết bị di động. Android có rất nhiều công cụ và dụng cụ miễn phí để nghiên cứu và phát triển phần mềm trên nền tảng của nó.

Android là nền tảng phần mềm dựa trên mã nguồn mở Linux OS (Kernel 2.6) cho máy di động và các phần mềm trung gian (middleware) để hỗ trợ các ứng dụng mà người dùng cần đến. Android được xây dựng để cho phép các nhà phát triển tạo ra các ứng dụng hấp dẫn, tận dụng các tính năng mà thiết bị di động đã cung cấp. Nó được xây dựng để thực sự mở, sử dụng máy ảo tùy chỉnh được thiết kế để tối ưu hóa bộ nhớ và tài nguyên phần cứng trong một môi trường di động.

Android không phân biệt giữa các ứng dụng lõi của thiết bị di động và ứng dụng của bên thứ ba. Tất cả có thể được xây dựng để truy cập bằng khả năng của thiết bị di động cung cấp cho người dùng với một dải rộng các ứng dụng và dịch vụ. Với các thiết bị xây dựng trên Android, người dùng có thể hoàn toàn thích ứng với thiết bị di động đến lợi ích của họ. Với Android, một nhà phát triển có thể xây dựng ưng dụng cho phép người dùng xem vị trí của bạn bè, người thân của họ và được cảnh báo khi họ có đang ở trong một vùng phụ cận cho họ một cơ hội để kết nối.

Android cung cấp truy cập đến một loạt các thư viện công cụ hữu ích và có thể được sử dụng để xây dựng các ứng dụng phong phú. Ngoài ra, Android bao gồm một tập hợp đầy đủ công cụ đã được xây dựng công phu, với việc cung cấp nền tảng phát triển với năng suất cao và cái nhìn sâu sắc vào các ứng dụng

Ngoài ra còn vì một số lý do khách quan, do nguồn kinh phí có hạn nên chưa thể tiếp cận

### 2.2.3 Phần mềm hỗ trợ lập trình Android: Android Studio

Ở trong khuôn khổ đồ án này em sẽ chọn Android Studio là môi trường lập trình cho ứng dụng. Android Studio là một phầm mềm bao gồm các bộ công cụ khác nhau dùng để phát triển ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như các loại điện thoại smartphone, các tablet... Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/deploy (trong đó có trình giả lập simulator để giả lập môi trường của thiết bị điện thoại hoặc tablet trên máy tính) cho phép các lập trình viên có thể nhanh chóng phát triển các ứng dụng từ đơn giản tới phức tạp. Một số tính năng chính

* Một hệ thống xây dựng Gradle linh hoạt
* Trình mô phỏng nhanh và tính năng phong phú
* Instant Run để đẩy các thay đổi vào ứng dụng đang chạy của bạn mà không cần xây dựng một APK mới
* Mẫu mã và tích hợp GitHub để giúp bạn xây dựng các tính năng ứng dụng phổ biến và nhập mã mẫu
* Các công cụ và khuôn khổ thử nghiệm mở rộng
* Lint công cụ để bắt hiệu suất, khả năng sử dụng, tương thích phiên bản, và các vấn đề khác
* Tích hợp hỗ trợ [Google Cloud Platform](https://cloud.google.com/tools/android-studio/docs/) , giúp dễ dàng tích hợp Google Cloud Messaging và App Engine
* Hỗ trợ tích hợp sâu [Firebase](https://danglong.name.vn/firebase-la-gi/) vào trong các ứng dụng chỉ sau một click chuột.

### 2.2.4 Gửi định vị lên web

Việc gửi thông tin định vị em sẽ sử dụng Socket.IO hoặc giao thức MQTT (hoặc AMQP). Để xây dựng một ứng dụng realtime cần sử dụng Socket.IO. Socket.IO sẽ giúp các bên ở những địa điểm khác nhau kết nối với nhau, truyền dữ liệu ngay lập tức thông qua server trung gian. Socket.IO có thể được sử dụng trong nhiều ứng dụng như chat, game online, cập nhật kết quả của một trận đấu đang xảy ra...

Socket.IO không phải là một ngôn ngữ, mà chỉ là 1 công cụ giúp thực hiện những ứng dụng realtime. Vì thế, không thể sử dụng Socket.IO để thay thế hoàn toàn cho một ngôn ngữ, mà phải sử dụng kết hợp với một ngôn ngữ khác. Ngôn ngữ đó có thể là PHP, Asp.net, Nodejs…

Cấu trúc một ứng dụng realtime sử dụng socket bao gồm 2 phần: phía server, phía client.

* Phía server: Đây là nơi sẽ triển khai Socket.IO. Ngôn ngữ để dựng server có thể là PHP, Asp.net, Nodejs... Tuy nhiên, tùy vào ngôn ngữ lựa chọn mà cách cấu trúc server khác nhau. Ở đây em sử dụng Nodejs để dựng server vì như vậy có thể cài trực tiếp Socket.IO đơn giản hơn băng thư viện “socket.io”.
* Phía client: Ở phía client sẽ xây dựng giao diện người dùng. Ở đây sử dụng Reactjs

Những thứ cần chuẩn bị để sử dụng Socket.IO

* Cài đặt Socket.IO bằng câu lệnh: npm I - save socketio. Sau khi hoàn thành bước cài đặt ở trên, hệ thống sẽ tạo ra file package.json. Đây là file dùng để cài đặt cấu hình server.
* Cần đặt những package cần thiết trên server để có thể làm web và ứng dụng realtime.

Code sử dụng socket.io phía server

const express = require('express')

const app = express()

const server = require('http').createServer(app)

const io = require('socket.io')(server)

io.on('connection', (socket) => {

console.log("New client connected" + socket);

socket.emit('request', /\* \*/); // emit an event to the socket

io.emit('broadcast', /\* \*/); // emit an event to all connected sockets

socket.on('reply', function(){

/\* \*/

}); // listen to the event

socket.on('disconnect', () => {

socket.to(roomId.broadcast.emit('user-disconnected', userId)

})

})

## 2.3 Giao diện web cho người quản lý

### 2.3.1 Giao diện

Đây là phần giao diện cho phía quản lý trên web với các chức năng cơ bản là đăng nhập vào hệ thống và giám sát, theo dõi vị trí hiện tại của người đeo đồng hồ

Trong thời gian hiện nay, xu hướng sử dụng Javascrips để code hoàn toàn từ Back-end (với Nodejs) và Front-end (sử dụng Angular,Reactjs,VueJS) . Đặc biệt là ReactJS nổi lên vì tính đơn giản dễ hiểu, dễ tiếp cận. **ReactJS** là một thư viện JavaScript có tính hiệu quả và linh hoạt để xây dựng các thành phần giao diện người dùng (UI) có thể sử dụng lại. **ReactJS** giúp phân chia các UI phức tạp thành các thành phần nhỏ (*được gọi là component*). Nó được tạo ra bởi Jordan Walke, một kỹ sư phần mềm tại Facebook. **ReactJS** ban đầu được phát triển và duy trì bởi Facebook và sau đó được sử dụng trong các sản phẩm của mình như **WhatsApp & Instagram**. ReactJS được dùng để xây dựng các ứng dụng [*single page application*] (SPA). Một trong những điểm hấp dẫn của ReacJS là nó không chỉ được xây dựng bên phía clients mà còn sử dụng được bên phía server. Đây không phải là 1 framework như Angular mà là 1 lirbary dành cho Javascrips chính vì thế hoàn toàn có thể sử dụng cho web và App Moblie.

Tuy nhiên, do tính chất được nhiều lập trình viên công nhận của Javascrips là nó khác bất logic và có nhiều chuẩn để biên dịch (Es5, Es6…) nên em sử dụng Parcel-bundler để đóng gói code cho tiện triển khai và ngoài ra cũng có thể giúp code có thể triển khai trên nhiều trình duyệt. Lí do em chọn Parcel-bundler là vì nó triển khai nhanh đơn giản, ít phải config, phù hợp với quy mô project đồng thời có trải nhiệm Hot Module Replacement

# CHƯƠNG 3. TRIỂN KHAI PHẦN MỀM

## 3.1 Thiết kế database lưu trữ dữ liệu

Như đã trình bày ở chương 2, em sẽ sử dụng NoSQL để lưu trữ dữ liệu

Database sẽ bao gồm 3 phần Colection: Account, Track, Location

+ Account sẽ lưu trữ thông tin xác thực của người dùng.

+ Track lưu thông tin về xác định đối tượng được gửi lên , đây là 1 Document của Collection Account

+ Location lưu trữ thông tin định vị trong từng khoàng thời gian được liên kết với Track thông qua ObjectID

## Thiết kế giao diện phần mềm

## Giao Diện Web

Giao diện web sử dụng ReactJS , bao gồm 2 layout chính là : AuthLayout và DashboardLayout

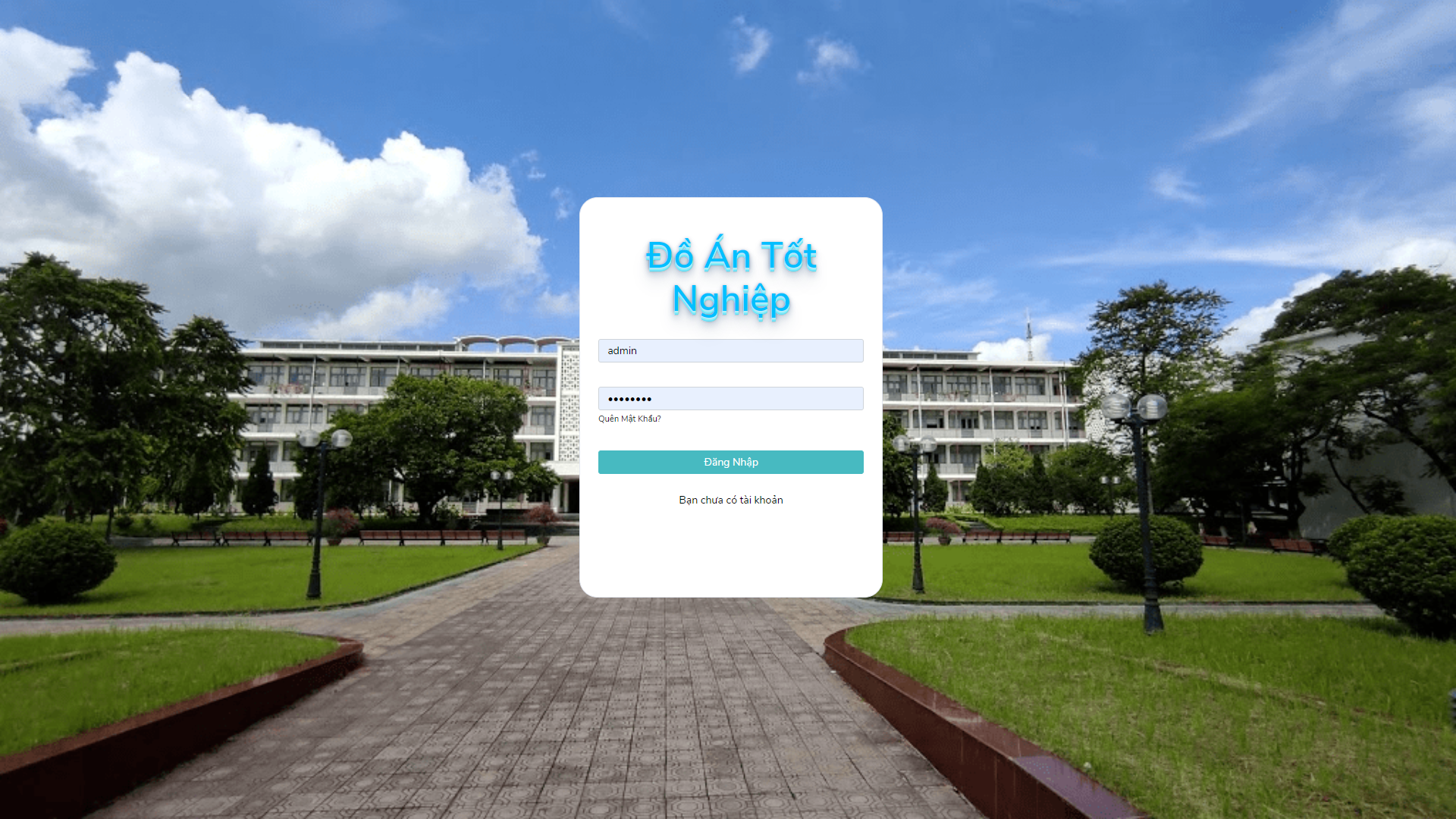
AuthLayout được route đến đầu tiên để kiểm tra quyền đăng nhập trên hệ thống và gồm 3 component con là : Đăng nhập – SingIn ,Đăng Kí – SingUp, Quên mật Khẩu – ForgetPasswork

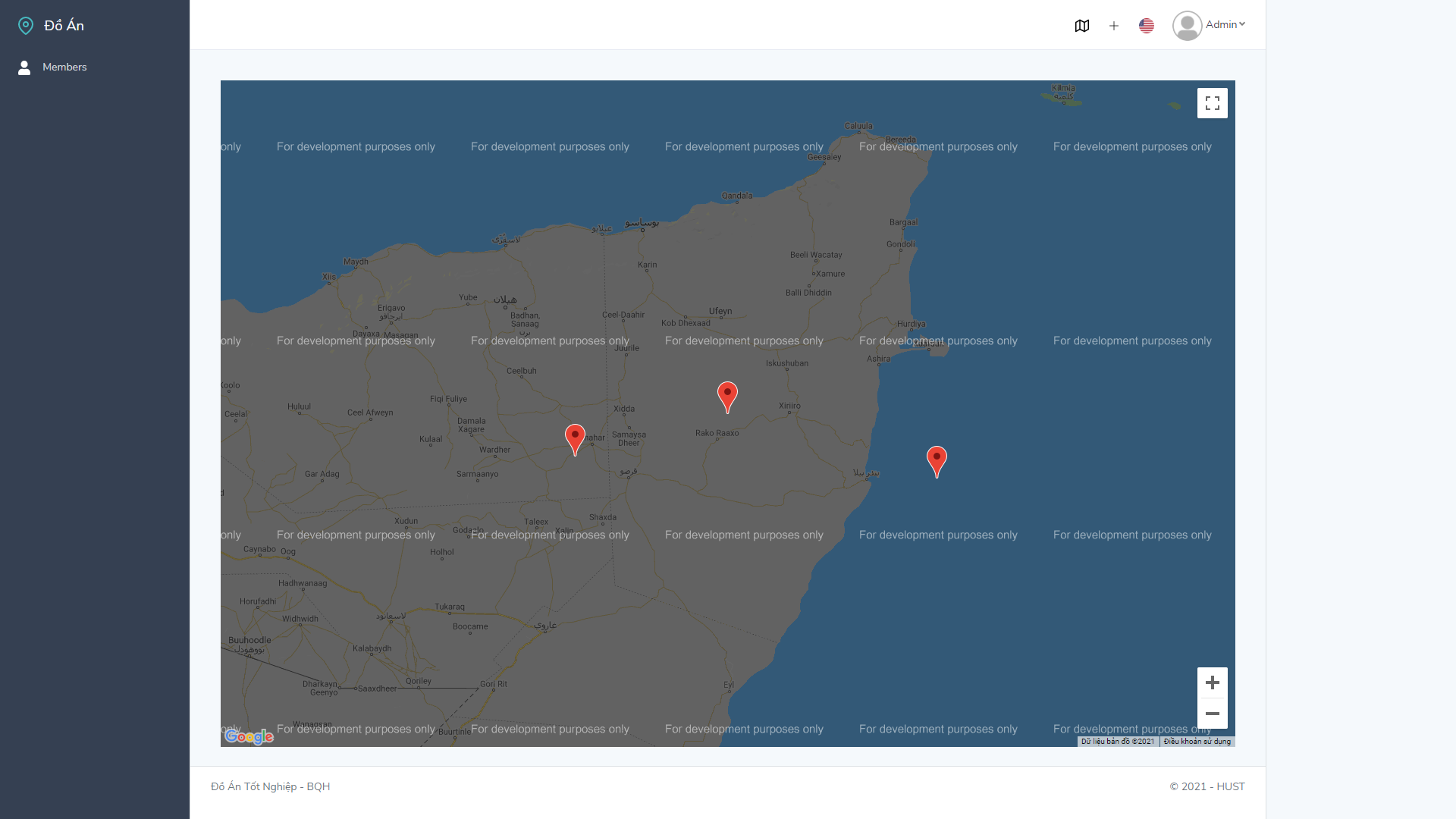
DashboardLayout cấu tạo từ 3 phần SideBar ,NavBar,Main.

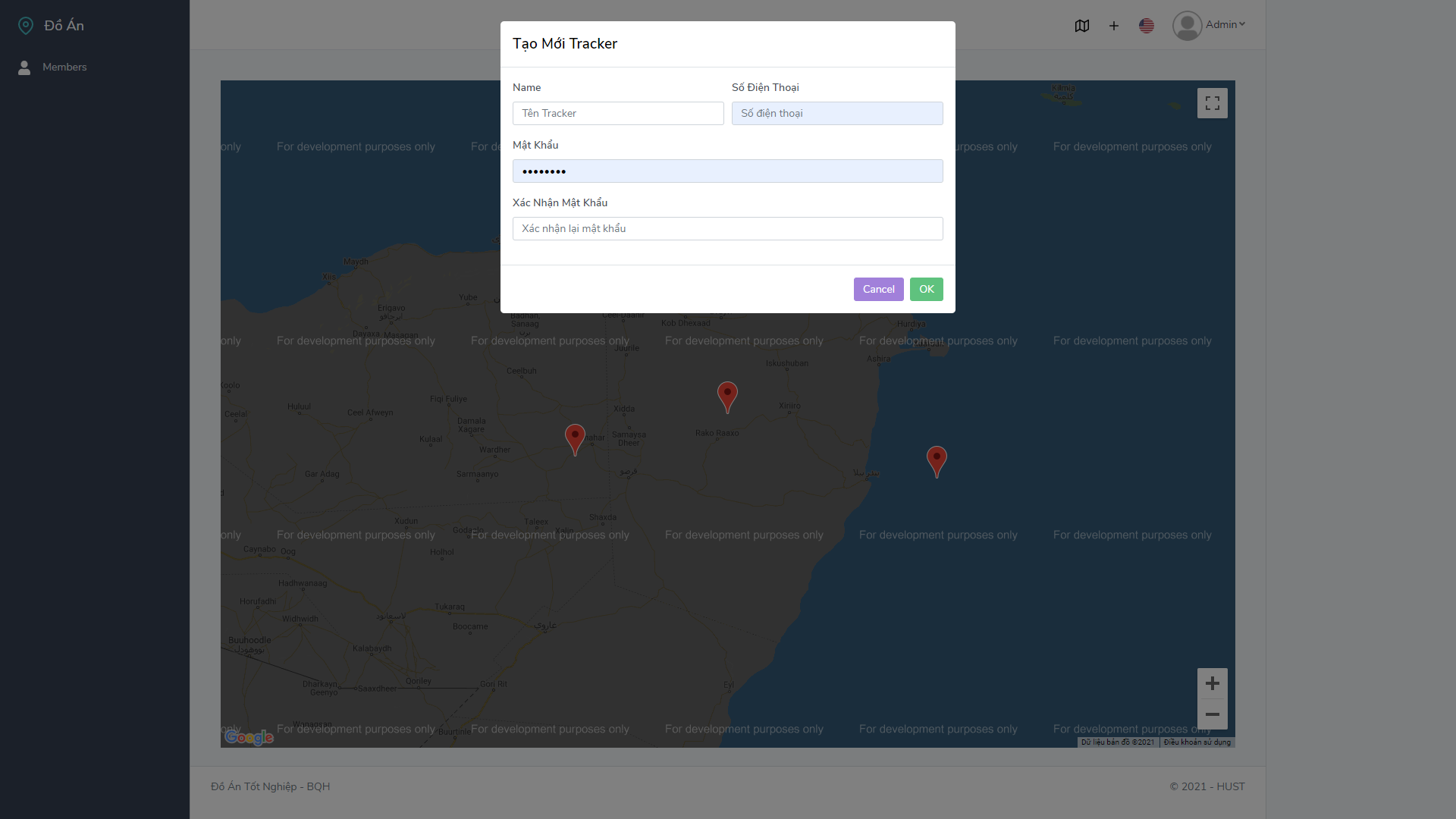
SideBar bên trái cùng và là nơi hiển thị danh sách các tracker

NavBar trên cùng hiển thị các chức năng của người dùng

Main sẽ như 1 sublayout để set-up view 2 component con là Map và InfoUser



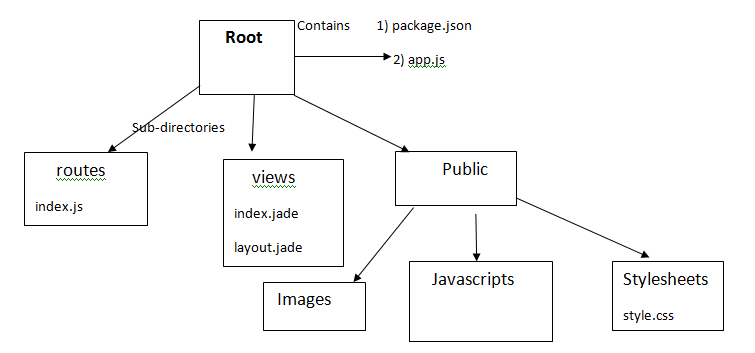




## Triển khai code phần server xử lí và code phần app

* + 1. **Phần server**

Server sử dụng Nodejs với framework Express :



Cấu trúc của Express rất đơn giản, bao gồm 3 phần :

* File package.json : đây là file cấu hình chứa các thư viện phụ thuộc,các thông tin của chương trình ,các command line được định nghĩa để sử dụng,…
* File app.js là main chính của chương trình chứa các thông tin được cấu hình và chạy xuyên suốt chương trình.
* Các thư mục khác : là các forder và các file config(vd:.env,.git) tạo ra tùy theo parterm do người viết định nghĩa

Phần server được thiết kế theo mô hình MVC ,tuy nhiên phần View được tách riêng biệt ra và dữ liệu được trả về bằng RESTfull API .

* + - 1. Mô hình MVC

MVC (Model-View-Controller) là một mô hình thiết kế kinh điển trong việc thiết kế phần mềm gồm 3 phần M – V – C.Mỗi phần có 1 chức năng riêng rẽ và tách biệt:



Các thành phần trong MVC:

* Model (M) : hiểu đơn giản đó là phần lưu trữ có nhiệm vụ tương tác với database để lấy dữ liệu.Nó được controller gọi lên và xử lý trả về cho controller trả lại cho bên View
* View (V) :là phần tương tác của user với controller .Nó sẽ cho người dùng thấy được dữ liệu và là nơi để người dùng yêu cầu dữ liệu
* Controller (C) : là nhân chính của mô hình nơi tiếp nhận xử lí và trả lại thông tin giữa M và V

Mô hình MVC là 1 mô hình sử dụng rất nhiều ,khá phổ biến và có nhiều biến thể được xây dựng lại .Ưu điểm của nó có thế kể đến đó là :

* Tính độc lập , nhất quán của luồng dữ liệu giúp việc sửa đổi bảo trì,tìm lỗi dễ dàng
* Mô hình rất đơn giản, phân chia rõ ràng dễ tiếp cận
* Giúp bảo mật hệ thống

Nhưng đối với các mô hình nhỏ thì nó lại quá cồng kềnh để triển khai và với những mô hình được coi là phức tạp nó cũng không đạt được nhiều tính khả dụng

* + - 1. RESTfull API

REST (**RE**presentational **S**tate **T**ransfer) được đưa ra vào năm 2000, trong luận văn tiến sĩ của Roy Thomas Fielding (đồng sáng lập giao thức HTTP). Nó là một dạng chuyển đổi cấu trúc dữ liệu, là một phong cách kiến ​​trúc cho việc thiết kế các ứng dụng có kết nối. Nó sử dụng HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

**API** (**A**pplication **P**rogramming **I**nterface) là giao diện lập trình ứng dụng giúp tạo ra các phương thức kết nối với các thư viện và ứng dụng khác nhau.

REST API là một ứng dụng chuyển đổi cấu trúc dữ liệu có các phương thức để kết nối với các thư viện và ứng dụng khác. REST API không được xem là một công nghệ, nó là một giải pháp để tạo ra các ứng dụng web services thay thế cho các kiểu khác như SOAP, WSDL (Web Service Definition Language),…

**RESTful API** là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.



Các API chức năng cơ bản

|  |  |  |  |
| --- | --- | --- | --- |
| Tên API | Chức năng | Hearder | Body |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* + - 1. CORS

CORS (Cross-Origin Resource Sharing) là một kĩ thuật được sinh ra để làm cho việc tương tác giữa client và server được dễ dàng hơn, nó cho phép JavaScript ở một trang web có thể tạo request lên một REST API được host ở một domain khác.

* Cơ chế hoạt động:

Trong trường hợp đơn giản nhất, phía client (tức là cái web app chạy ở browser đó) sẽ tạo request GET, POST, PUT, HEAD, .. để yêu cầu server làm một việc gì đó. Những request này sẽ được đính kèm một header tên là Origin để chỉ định origin của client code (giá trị của header này chính là domain của trang web).

Server sẽ xem xét Origin để biết được nguồn này có phải là nguồn hợp lệ hay không. Nếu hợp lệ, server sẽ trả về response kèm với header “ Access-Control-Allow-Origin”. Header này sẽ cho biết xem client có phải là nguồn hợp lệ để browser tiếp tục thực hiện quá trình request.

Trong trường hợp thông thường, Access-Control-Allow-Origin sẽ có giá trị giống như Origin, một số trường hợp giá trị của Access-Control-Allow-Origin sẽ nhìn giống giống như [Regex](https://codeaholicguy.com/2017/10/31/tai-sao-developer-can-biet-regular-expression/) hay chỉ đơn giản là \*, tuy nhiên thì cách dùng \* thường được coi là không an toàn, ngoại trừ trường hợp API của bạn được public hoàn toàn và ai cũng có thể truy cập được.

Và như thế, nếu không có header Access-Control-Allow-Origin hoặc giá trị của nó không hợp lệ thì browser sẽ chặn lại.

* Xử lý CORS

Cách đơn giản nhất đó chính là sử dụng thư viện cors bằng các sử dụng câu lệnh “npm install cors” và thêm nó vào trong file app.js

const cors = require('cors')

const express = require('express')

const app = express()

app.use(cors())

* + - 1. Sử dụng Socket.io
      2. Một số thư viện sử dụng trong express
      3. Sử dụng Socket.io

* + 1. **Phần App Mobile**

## 3.4 Triển khai code giao diện web

## 3.5 Kiểm thử phần mềm:

# CHƯƠNG 4. KẾT QUẢ ĐẠT ĐƯỢC

# CHƯƠNG 5. KẾT LUẬN

Sau thời gian làm đồ án, được sự hướng dẫn tận tình của thầy **Trịnh Quang Đức** cũng như sự giúp đỡ của các thầy giáo khác trong bộ môn, em đã hoàn thành những yêu cầu và nhiệm vụ của Đồ án tốt nghiệp.

Trong đồ án này em đã thực hiện được một số kết quả:

1. Tổng quan về cách lập trình một phần mềm ứng dụng
2. Thiết kế được cơ sở dữ liệu phân quyền cho người dùng
3. Hiểu rõ hơn cách thức hoạt động của hệ thống định vị toàn cầu GPS

Bên cạnh đó đồ án vẫn còn có hạn chế:

1. Chưa đưa ra được sự cải tiến đáng kể, hữu ích hơn cho phần mềm

Trong quá trình thực hiện đồ án không thể tránh khỏi những thiếu sót, em mong các thầy giáo chỉ bảo để sửa chữa, rút kinh nghiệm để khi ra trường trở thành một cử nhân có trình độ vững vàng hơn.

Một lần nữa em xin chân thành cảm ơn!