

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÁO CÁO ĐỒ ÁN ĐA NGÀNH
ĐỀ TÀI: HỆ THỐNG THEO DÕI ĐỘ ẨM
VÀ HỖ TRỢ TƯỚI TIÊU

GVHD: Trần Khải Thiện

SVTH:

Nguyễn Lê Tiến Đạt 1710959

Võ Văn Đức 1711096

Nguyễn Bảo Huy 1711502

Trần Đình Huy 1711550

Nguyễn Tiến Hùng 1711611

TP.HCM, 2020

Mục Lục

I. Giới Thiệu Đề Tài.....	4
1. Lý do chọn đề tài.....	4
2. Các tính năng:	4
a. Người dùng:.....	4
b. Admin:	5
c. Một số tính năng của hệ thống:	6
II. Phân Công Công Việc	7
III. Tìm Hiểu Công Nghệ.....	9
1. Protocol.....	9
1.1. MQTT.....	9
1.2. WebSocket	10
2. Platform	12
2.1. Node.js.....	12
2.2. EMQ X	14
3. Framework – Express	15
4. Database – MongoDB	17
5. Library	19
5.1. MQTT.js	19
5.2. Socket.io	20
5.3. Mongoose.....	21
5.4. Agenda.....	26
5.5. jQuery	27
5.6. Bootstrap.....	28
6. Giả lập thiết bị.....	30
IV. Thiết Kế Hệ Thống.....	31
1. Thiết kế tổng quan.....	31
2. Thiết kế database.....	31
V. Mô Tả Chi Tiết.....	32
1. Usecase	32
2. Usecase chi tiết.....	32
2.1. Quản lí lịch tưới	32
2.2. Tưới tự động	33

2.3.	Quản lí người dùng.....	33
2.4.	Bản đồ thiết bị.....	33
2.5	Nhận và hiển thị dữ liệu từ thiết bị.....	34
2.6	Quản lý thông tin cá nhân:.....	35
2.7	Hiển thị thông tin tổng quan:	37
2.5.	Quản lí thiết bị.....	39
2.6.	Đăng xuất.....	39
2.7.	Đăng ký.....	40
2.8.	Đăng nhập.....	40
3.	Yêu cầu chức năng	41
3.1.	Hiển thị vị trí thiết bị trên bản đồ.....	41
3.2.	Tìm kiếm thiết bị trên bản đồ.....	42
3.3.	Hiển thị lịch tưới	43
3.4.	Xóa lịch tưới	44
3.5.	Đặt lịch tưới	45
3.6.	Đặt lịch tưới	46
3.7.	Tự động lưu thông số từ cảm biến	46
3.8.	Hiển thị người dùng.....	47
3.9.	Hiển thị phiên đăng nhập người dùng	49
3.10.	Xóa phiên đăng nhập người dùng	50
3.11.	Ràng buộc thiết bị với người dùng.....	51
3.12.	Xóa ràng buộc thiết bị với người dùng.....	51
3.13.	Hiển thị ràng buộc thiết bị với thiết bị.....	53
3.14.	Ràng buộc thiết bị với thiết bị.....	54
3.15.	Xóa Ràng buộc thiết bị với thiết bị	54
3.16.	Cập nhật vị trí thiết bị.....	56
3.17.	Đăng xuất (người dùng).....	57
3.18.	Đăng nhập (admin).....	58
3.19.	Xác thực (admin).....	59
3.20.	Đăng xuất (admin).....	60
3.21	Nhận và hiển thị dữ liệu từ thiết bị:.....	61
3.22	Quản lý thông tin cá nhân:.....	62
3.23	Hiển thị thông tin tổng quan:	65
4.	Yêu cầu phi chức năng	69

VI. Tổng kết:	71
1. Kết quả hiện thực:	71
2. Phương hướng phát triển trong tương lai:	82
3. Kết luận:	82

I. Giới Thiệu Đề Tài

1. Lý do chọn đề tài

Chúng ta đang sống trong thời đại 4.0, hay Cuộc cách mạng công nghiệp lần thứ 4. Cuộc cách mạng này đã và đang tác động đến tất cả các lĩnh vực trong đời sống xã hội hiện nay. Trong đó, nền kinh tế của nhân loại đang đứng trước những sự thay đổi mới về phương thức sản xuất mới, được ứng dụng những thành quả đạt được của cuộc cách mạng công nghiệp lần thứ 4 này. Nó đang phá vỡ hầu hết các nền tảng sản xuất truyền thống trước đó, mà nông nghiệp, một lĩnh vực chịu tác động mạnh đang dần thay đổi theo từng ngày. Hàng loạt các trang trại, nông trại ứng dụng các thiết bị IoT vào vận hành, hoạt động một cách tự động hóa, vừa mang lại năng suất cũng như chất lượng cao, đồng thời tiết kiệm được nhân công.

Trong nông nghiệp, người xưa thường nói: “Nhất nước, nhì phân, tam cần, tứ giống”. Từ lâu, người nông dân đã nhận thức được tầm quan trọng của việc tưới tiêu. Tưới nước cho đúng kỹ thuật, đúng phương pháp sẽ giúp cho cây vận chuyển được chất dinh dưỡng để nuôi cây, cho phép cây sinh trưởng và phát triển tốt, đồng thời còn hạn chế được sâu bệnh gây hại cho cây trồng,... Vì vậy, tưới tiêu như nào cho đúng, cho khoa học luôn là nỗi trăn trở của nông dân / người trồng cây.

Với mong muốn xây dựng, phát triển một nền nông nghiệp tự động hóa một cách chính xác, khoa học và hiện đại phục vụ cho công việc tưới tiêu. Nhóm chúng em đi đến đề tài “ Phát triển hệ thống theo dõi độ ẩm của các cây trồng tại một khu vườn/nông trại, bật, tắt thiết bị tưới nước phù hợp khi độ ẩm không đủ theo ngưỡng cho phép. Ghi nhận hoạt động”. Ứng dụng hướng đến những người trồng cây hộ gia đình, những trang trại. Giúp họ quan sát được nhiệt độ, độ ẩm của đất và hỗ trợ việc tưới tiêu cũng như ghi nhận lại các hoạt động.

2. Các tính năng:

Ứng dụng Easy Plant Tree phục vụ hai đối tượng là người dùng và admin. Các tính năng được xây dựng xoay quanh từng đối tượng riêng biệt.

Cụ thể như sau:

a. Người dùng:

Tính năng	Mô tả tính năng
-----------	-----------------

Đăng ký	<ul style="list-style-type: none"> ✓ Cho phép người dùng tạo tài khoản để sử dụng ứng dụng. ✓ Tài khoản này phải được xác thực qua email.
Đăng nhập	<ul style="list-style-type: none"> ✓ Cho phép người dùng đăng nhập để sử dụng ứng dụng. ✓ Người dùng có thể thay đổi mật khẩu dựa vào email đã đăng ký.
Đăng xuất	<ul style="list-style-type: none"> ✓ Người dùng có thể đăng xuất khỏi ứng dụng
Dashboard	<ul style="list-style-type: none"> ✓ Theo dõi độ ẩm của từng sensor với thời gian thực và hiển thị dưới dạng biểu đồ. Với một số yêu cầu như lazy loading dữ liệu, cập nhật dữ liệu realtime. ✓ Xem độ ẩm trung bình của từng cảm biến trong tuần này và tuần trước. ✓ Xem số lần tưới nước trong ngày của tuần này và tuần trước.
Bản đồ thiết bị	<ul style="list-style-type: none"> ✓ Hiển thị bản đồ. ✓ Tìm kiếm thiết bị trên bản đồ.
Lịch tưới	<ul style="list-style-type: none"> ✓ Hiển thị lịch tưới và cập nhật theo thời gian thực. ✓ Hiển thị lịch sử tưới. ✓ Đặt lịch tưới và giải quyết đụng độ.
Quản lý thông tin cá nhân	<ul style="list-style-type: none"> ✓ Hiển thị và xóa phiên đăng nhập. ✓ Hiển thị thiết bị và quan hệ phụ thuộc các thiết bị. ✓ Đặt ngưỡng độ ẩm cho phép cho các cảm biến.

b. Admin:

Admin có tính năng đăng nhập, đăng xuất tương tự như của người dùng.

Tính năng	Mô tả tính năng
Hiển thị thông tin thiết bị.	<ul style="list-style-type: none"> ✓ Hiển thị danh sách người dùng. ✓ Hiển thị danh sách các thiết bị.

	<ul style="list-style-type: none"> ✓ Thống kê số lượng người dùng và thiết bị. ✓ Tìm kiếm người dùng hoặc thiết bị.
Quản lý người dùng	<ul style="list-style-type: none"> ✓ Hiển thị danh sách người dùng. ✓ Thêm, xóa các thiết bị cho người dùng. ✓ Thêm, xóa các ràng buộc cho thiết bị. ✓ Hiển thị thiết bị trên bản đồ và cập nhật vị trí thiết bị.
Quản lý thiết bị	<ul style="list-style-type: none"> ✓ Hiển thị danh sách cảm biến riêng, motor riêng. ✓ Thêm, xóa thiết bị.

c. Một số tính năng của hệ thống:

Tính năng	Mô tả tính năng
Tưới tự động	<ul style="list-style-type: none"> ✓ Tự động điều khiển motor thích hợp thông qua ngưỡng độ ẩm đã thiết đặt.
Xử lý cảm biến	<ul style="list-style-type: none"> ✓ Kết nối, lắng nghe và lưu thông tin từ sensor gửi về.

II. Phân Công Công Việc

Thành Viên	Nhiệm Vụ	Đánh Giá
Nguyễn Lê Tiên Đạt	<ul style="list-style-type: none"> - Quản lí phân công công việc - Đảm bảo tiến độ - Chịu trách nhiệm hiển thị độ ẩm trong lịch sử (lazy loading), cập nhật độ ẩm theo thời gian thực (real time), thống kê + thiết kế hiện giao diện - Chịu trách nhiệm viết middleware xác thực Websocket cho người dùng thông qua JWT - Thiết kế giao diện đa số tính năng 	Tinh thần trách nhiệm cao, hoàn thành nhiệm vụ được giao trong thời gian quy định
Võ Văn Đức	<ul style="list-style-type: none"> - Lựa chọn công nghệ - Thiết kế hệ thống - Thiết kế cấu trúc project, thiết đặt môi trường bao gồm database, môi trường deploy, giả lập môi trường để hiện thực - Chịu trách nhiệm quản lí lịch tưới bao gồm đặt, xóa, kiểm tra trùng lịch + thiết kế hiện giao diện - Chịu trách nhiệm quản lí người dùng bao gồm thêm, xóa ràng buộc người dùng với thiết bị, thiết bị với thiết bị, đặt vị trí thiết bị + thiết kế hiện giao diện - Chịu trách nhiệm trang thông tin người dùng bao gồm thông tin người dùng, thông tin thiết bị, và đặt ngưỡng cho cảm bị 	Tinh thần trách nhiệm cao, hoàn thành nhiệm vụ được giao trong thời gian quy định
Nguyễn Bảo Huy	<ul style="list-style-type: none"> - Thiết kế + Hiện thực database (mongoose) - Chịu trách nhiệm xử lí thông tin từ sensor gửi bao gồm kiểm tra và lưu vào database - Chịu trách nhiệm hiển thị và tìm kiếm sensor trên bản đồ của người dùng + thiết kế hiện giao diện - Hiện thực model ML (không tích hợp vào app) 	Tinh thần trách nhiệm cao, hoàn thành nhiệm vụ được giao trong thời gian quy định

Nguyễn Tiến Hùng	<ul style="list-style-type: none"> - Chịu trách nhiệm các vấn đề xác thực người dùng bao gồm đăng nhập, đăng xuất, đăng kí, xác thực người dùng sử dụng JWT - Chịu trách nhiệm tưới tự động, cách ra quyết định tưới - Chịu trách nhiệm dịch vụ gửi mail tự động đến người dùng 	Có tinh thần trách nhiệm, hoàn thành công việc tốt mặc dù trễ thời gian quy định
Trần Đình Huy	<ul style="list-style-type: none"> - Chịu trách nhiệm các vấn đề đăng nhập, xác thực admin thông qua JWT - Chịu trách nhiệm kết nối và giao tiếp với MQTT Broker - Trang chủ cho admin bao gồm số lượng thiết bị người dùng, và liệt kê thiết bị và người dùng 	Có tinh thần trách nhiệm cao, hoàn thành công việc đúng hạn nhưng ở mức trung bình

III. Tìm Hiểu Công Nghệ

1. Protocol

1.1. MQTT

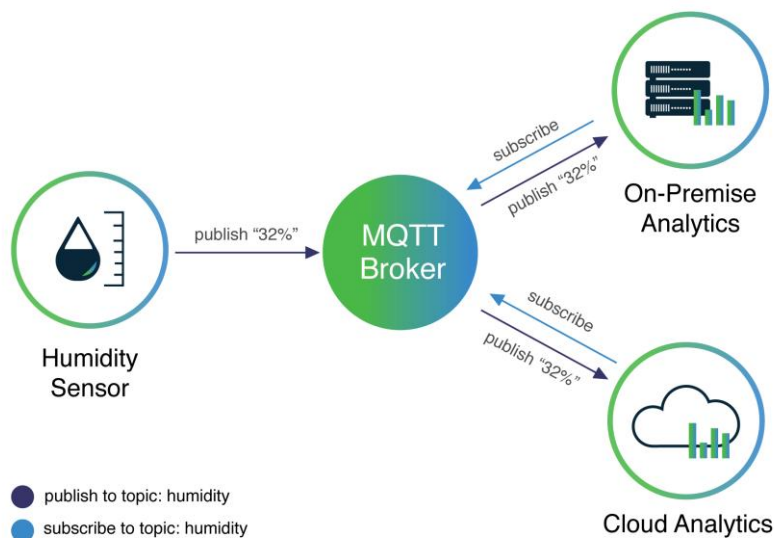
MQTT là viết tắt của MQ Telemetry, MQTT được sáng tạo bởi Dr Andy Stanford-Clark và Arlen Nipper.

MQTT là giao thức truyền thông điệp đơn giản và nhẹ theo mô hình publish/subscribe được thiết kế cho các thiết bị có khả năng hạn chế, các mạng có băng thông thấp, độ trễ cao và không đáng tin cậy. Giao thức được thiết kế nhằm giảm thiểu băng thông mạng, yêu cầu phần cứng thiết bị và đảm bảo độ tin cậy trong truyền thông điệp. Do vậy MQTT trở nên lý tưởng cho các kết nối “machine to machine” hay “internet of things”.

Giao thức định nghĩa hai đối tượng loại trong mạng: message broker và client:

- MQTT Broker là một server nhận tất cả thông điệp là chuyển tiếp đến các client đích.
- MQTT Client là các thiết bị chạy thư viện MQTT và kết nối đến MQTT Broker thông qua network.

Thông tin được tổ chức theo những topic. Khi publisher gửi thông điệp lên MQTT Broker thông qua một topic thì những client subscribe topic đó (subscriber) sẽ được MQTT Broker chuyển tiếp đến tất cả subscriber.



Hình 2.1.1: Thông tin tổ chức theo topic¹

¹ [What is MQTT and Why You Need It in Your IoT Architecture](#)

Giao thức cho phép lựa chọn chất lượng phương thức gửi (QoS) bao gồm:

- QoS 0: Broker/client sẽ gửi dữ liệu đúng 1 lần, quá trình gửi được xác nhận bởi chỉ giao thức TCP/IP.
- QoS 1: Broker/client sẽ gửi dữ liệu với ít nhất 1 lần xác nhận từ đầu kia.
- QoS 2: Broker/client đảm bảo khi gửi dữ liệu thì phía nhận chỉ nhận được đúng 1 lần, quá trình này phải trải qua 4 bước bắt tay

Lựa chọn chất lượng phương thức phụ thuộc vào nhu cầu của ứng dụng điều này tạo nên giao thức MQTT trở nên linh hoạt phù hợp cho những kết nối M2M

1.2. WebSocket

Giao thức WebSocket là một giao thức được sử dụng rộng rãi cho việc phát triển ứng dụng real-time. Những phương thức trước đó để mô phỏng kết nối full-duplex được dựa trên polling, một phương thức đồng bộ mà client gửi request đến server để xem có thông tin không. Client nhận response từ server cả khi mà không có thông tin có sẵn.

Polling chỉ hoạt động tốt khi khoảng thời gian message có sẵn có thể biết trước được. Tuy nhiên, trên thực tế ở hầu hết ứng dụng real-time, tần suất xuất hiện message thường là không thể dự đoán trước. Ngoài ra, polling còn yêu cầu client mở và đóng các kết nối không cần thiết.

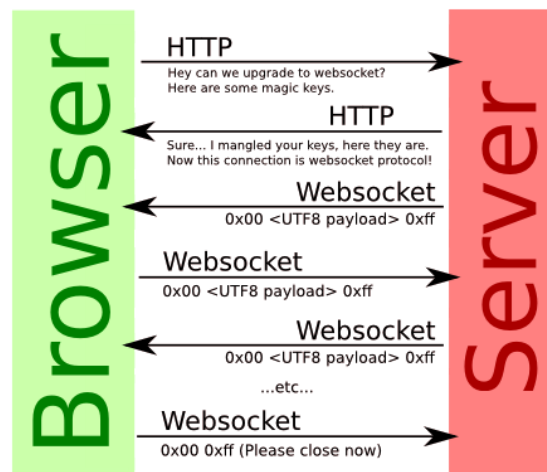
Long polling (hay Comet) là một phương thức giao tiếp phổ biến mà trong đó client mở kết nối đến server trong một giai đoạn nhất định. Nếu server không có thông tin gì, nó sẽ giữ kết nối đến khi có thông tin để trả về cho client hoặc đến khi đạt giới hạn thời gian cho kết nối đó (timeout). Về bản chất, Comet trì hoãn sự hoàn thành của HTTP response cho đến khi server có gì đó để trả về client, kỹ thuật này thường được gọi là hanging-GET hay pending-POST.

Thực tế là việc client phải thường xuyên reconnect đến server khiến long polling không phải là một sự lựa chọn tối ưu cho các ứng dụng real-time.

Giao thức WebSocket được bao hàm ở mục Connectivity trong specification của HTML5. Nó cho phép tạo kết nối full-duplex, hai chiều giữa client và server. Nó cung cấp một cách thức để tạo những kết nối bền bỉ, độ trễ thấp để hỗ trợ giao tiếp giữa client và server. Sử dụng WebSocket, bạn có thể tạo một ứng dụng real-time đúng nghĩa như ứng dụng chat, phối hợp soạn thảo văn bản, giao dịch chứng khoán hay game online nhiều người chơi cùng lúc.

WebSocket API cho phép ứng dụng của bạn kiểm soát giao thức WebSocket và phản hồi lại với những sự kiện được trigger bởi server. Vì API là hướng sự kiện nên khi kết nối full-

duplex được khởi tạo, nếu server có data để gửi về client hoặc khi resource mà ứng dụng quan tâm thay đổi state, nó lập tức gửi thông báo về phía client.



Kết nối WebSocket được khởi tạo bằng việc nâng cấp từ giao thức HTTP sang giao thức WebSocket trong quá trình handshake giữa client và server thông qua cùng một kết nối TCP. Header nâng cấp được bao gồm trong request nhằm thông báo cho server rằng client muốn tạo lập một kết nối WebSocket. Một khi được tạo lập các message WebSocket có thể được truyền đến và đi thông qua các method của WebSocket.

Bản chất bất đồng bộ của WebSocket có nghĩa là một khi một kết nối WebSocket được mở, ứng dụng sẽ lắng nghe những sự kiện (WebSocket Events). Để bắt đầu lắng nghe các sự kiện, bạn có thể thêm hàm callback vào object WebSocket hoặc sử dụng DOM method `addEventListener()` để thêm event listener.

Object WebSocket phát đi 4 sự kiện:

- Open: Server phản hồi lại request để mở kết nối WebSocket. Event này thông báo rằng handshake thành công và kết nối Websocket được khởi tạo. Callback của sự kiện này là `onopen`.
- Message: Client nhận được data chứa trong message từ server. Callback tương ứng của sự kiện này là `onmessage`.
- Error: Sự kiện xảy ra khi có bất kỳ lỗi nào trong kết nối WebSocket. Callback với sự kiện này là `onerror`.
- Close: Kết nối được đóng lại. Callback tương ứng với sự kiện này là `onclose`.

WebSocket Methods, Websocket cung cấp 2 method:

- `send()`: Method `send(data)` truyền data trên kết nối. Nếu vì lý do nào đó, kết nối không tồn tại hoặc bị đóng, method này sẽ trả về exception về tình trạng không hợp lệ của kết nối.

- `close()`: Method `close()` dùng để đóng kết nối hiện tại. Nếu kết nối đã bị đóng từ trước đó (có thể do lỗi kết nối), nó không gây ảnh hưởng gì. Method này nhận hai argument tùy chọn: code (dạng số) và reason (dạng text).

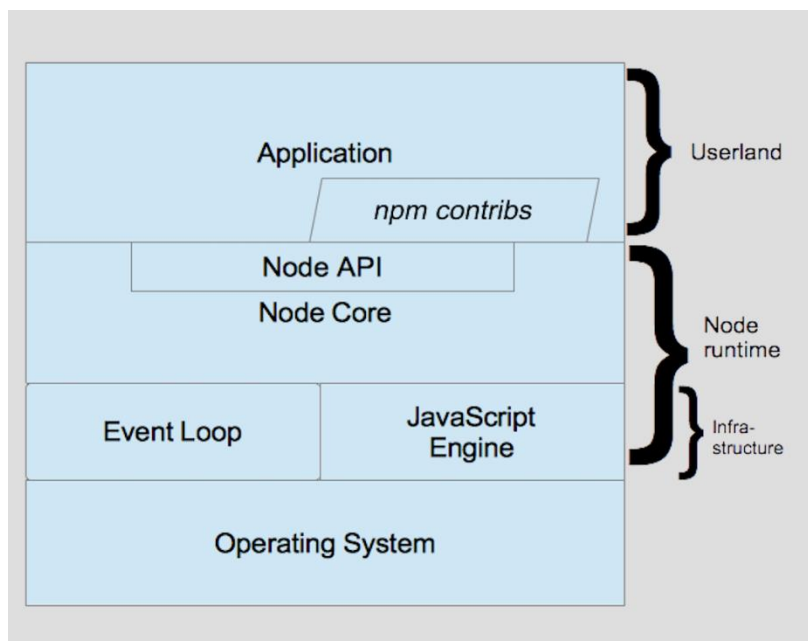
2. Platform

2.1. Node.js

Node.js là *môi trường thực thi JavaScript* được xây dựng trên [Chrome's V8 JavaScript engine](#). Được ra mắt vào năm 2009 bởi [Ryan Dahl](#) và sau đó được cộng đồng developer khắp thế giới đóng góp đến nay với hơn 30000 lượt commit, hơn 70000 star và hơn 17000 lượt fork từ [github](#).

Node.js sử dụng mô hình non-blocking I/O và kiểu lập trình bất đồng bộ, mặc dù JavaScript là ngôn ngữ lập trình đơn luồng nhưng với thiết kế của Node.js làm cho ứng dụng xây dựng trên nền tảng Node.js vẫn có hiệu suất cao.

Sơ lược về kiến trúc của Node.js



Hình 1.1.1: Kiến trúc Node.js – Node.js Course IBM

Node API là tập hợp các module dựng sẵn cung cấp bởi Node.js để cung cấp cho người dùng để xây dựng ứng dụng. Một số module phổ biến: File System, Crypto, Events, Http, Stream,...

Node Core là NodeAPI được hiện thực bằng C++, xây dựng trên thư viện libuv và Javascript Engine.

Javascript Engine được sử dụng cụ thể là Chrome's V8 engine.

Event loop chính là libuv, chịu trách nhiệm có hoạt động bất đồng bộ và non-blocking I/O.

Node.js đặc biệt hữu hiệu cho những ứng dụng không quá phức tạp về mặt tính toán mà nặng về mặt I/O như là truy xuất database hoặc đọc file. Đó cũng chính là lý do nhóm chọn Node.js làm platform phát triển ứng dụng.

Hướng dẫn cài đặt Node.js trên Ubuntu 18.04:

Bước 1: Cài đặt bằng Node Version Control - nvm:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.35.3/install.sh | bash
```

Bước 2: Cập nhật terminal:

Nếu sử dụng bash shell:

```
source ~/.bashrc
```

Nếu sử dụng zsh shell:

```
source ~/.zshrc
```

Bước 3: Cài đặt Node.js bản long time support – lts:

```
nvm install --lts
```

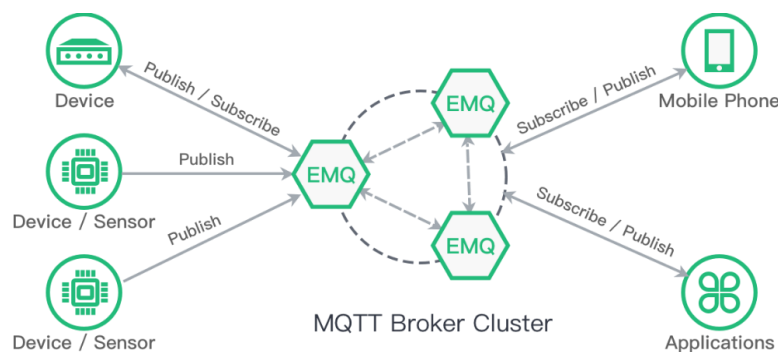
Để kiểm tra Node.js đã được cài đặt ta sử dụng trên terminal: `node -v`

2.2. EMQ X

EMQ X là MQTT message broker mã nguồn mở xây dựng trên Erlang/OTP platform. Sở dĩ được xây dựng trên Erlang/OTP platform vì platform này được đánh giá là rất tốt cho ứng dụng thời gian thực có độ trễ thấp và thích hợp cho hệ phân tán.

EMQ X được thiết kế chịu được lượng kết nối cực lớn và đảm bảo được định tuyến gói tin nhanh và chính xác giữa các thiết bị trong mạng.

EMQ X hỗ trợ các thiết bị có thể listen MQTT sử dụng tcp ở cổng 1883, ws ở cổng 8083 và một số giao thức khác. EMQ X hỗ trợ chất lượng truyền thông điệp QoS 0,1,2.



Hình 1.2.1: Tổng quan EMQ X – EMQX.IO

Hướng dẫn cài đặt và thiết đặt EMQX X lên Ubuntu 18.04 bằng terminal

Bước 1: Cài đặt:

```
curl https://repos.emqx.io/install_emqx.sh | bash
```

Bước 2: Thiết đặt EMQ X Broker khởi động cùng với OS

```
sudo update-rc.d emqx enable
```

Bước 3: Thiết đặt username và password cho plugin emqx_auth_username

Mở file emqx_auth_username.conf tại /etc/emqx/plugins

Thêm vào đầu file và save (thực hiện mở và lưu với quyền root):

```
auth.user.1.username = BKvm2  
auth.user.1.password = Hcmut_CSE_2020
```

Bước 4: Khởi động và load plugin emqx_auth_username:

```
sudo service emqx start && emqx_ctl plugins load emqx_auth_username
```

Sau khi thực hiện các bước trên thì EMQ X sẽ tự khởi động cùng với OS. Để quản lí các kết nối với Borker ta mở trình duyệt dùng url: [https://<ip>: 18083](https://<ip>:18083) đăng nhập với tên đăng nhập admin và mật khẩu public.

3. Framework – Express

Express là một web application framework cho Node.js, nó cung cấp cho chúng những rất nhiều tính năng mạnh mẽ trên nền tảng web. Express rất dễ dàng để phát triển các ứng dụng nhanh dựa trên Node.js cho các ứng dụng Web. Express hỗ trợ các phương thức HTTP và middleware tạo ra 1 API rất mạnh mẽ và sử dụng dễ dàng hơn. Các tính năng của Express framework phải kể đến như:

- Định tuyến mạnh mẽ.
- Tập trung vào hiệu suất cao.
- Phản hồi nhanh chóng.
- Trình trợ giúp HTTP (chuyên hướng, lưu trữ, v.v.).
- Hệ thống hỗ trợ hơn 14 view template engine.
- Đàm phán về nội dung (content negotiation).
- Có thể thực thi để tạo ứng dụng nhanh chóng.

Ưu điểm và khuyết điểm của Express:

- Ưu điểm:
 - o Phát triển ứng dụng nhanh: Express cho phép ta sử dụng cùng một ngôn ngữ là JavaScript ở cả back-end và front-end. Nó cung cấp cho các nhà phát triển

JavaScript cơ hội để trở thành fullstack. Do đó, quá trình phát triển nhanh hơn và dễ dàng hơn vì một người có thể quản lý cả lớp trình bày và truy cập dữ liệu.

- Xử lý yêu cầu I/O: Express là một lựa chọn tuyệt vời cho các ứng dụng xử lý nhiều yêu cầu và thông báo từ người dùng. Node.js cùng với Express.js có khả năng hỗ trợ hàng ngàn hành động đồng thời với nhau.
- Cộng đồng nguồn mở: Express là một trong những framework Node.js được hỗ trợ nhiều nhất. Nó có một cộng đồng mã nguồn mở, vì vậy mã luôn được xem xét và cải thiện.
- Dễ dàng tích hợp các dịch vụ và phần mềm trung gian của bên thứ ba: Express.js là một framework tối giản, vì vậy nhóm phát triển Express đã tạo các gói phần mềm trung gian để giải quyết các vấn đề phát triển khác nhau. Chúng bao gồm các tham số URL, phiên, dữ liệu POST, tiêu đề bảo mật, thư viện, v.v... Express.js không có quy tắc nghiêm ngặt và xác định nào về cách xử lý các tác vụ cụ thể hoặc lựa chọn thành phần nào. Ta có thể sử dụng bất kỳ middleware nào theo thứ tự thuận tiện cho dự án. Ngoài ra, ta có thể quyết định cách cấu trúc ứng dụng của mình.

- Nhược điểm:

- Mặc dù Express.js là một framework rất thuận tiện và dễ sử dụng, nhưng nó có một số nhược điểm nhỏ có thể ảnh hưởng đến quá trình phát triển. Trước khi bắt đầu phát triển với Node.js hoặc Express.js hãy suy nghĩ về quy mô và mức độ phức tạp của dự án. Dưới đây là những vấn đề tiềm ẩn chính mà ta có thể gặp phải và một số cách để tránh chúng.
- Bản chất hướng sự kiện (Event-driven nature): Express.js cũng như Node.js là một framework đơn luồng với vòng lặp sự kiện lắng nghe các sự kiện khác nhau và thực hiện các callback đã đăng ký(registered). Các nhà phát triển đã làm việc với các ngôn ngữ lập trình khác có thể thấy khó hiểu bản chất gọi lại. Điều đó dễ dẫn đến khó khăn trong việc duy trì ,nâng cấp mã nguồn trong tương lai.
- Triết lý của các plugin được gọi là middleware: Express.js được xây dựng trên triết lý này, đó là lý do tại sao điều quan trọng khi làm việc với NodeJs và Express là phải hiểu các khái niệm chính của nó. Ngắn gọn, middleware là một tập hợp con của các hàm chuỗi chạy giữa yêu cầu máy khách và câu trả lời của máy chủ.

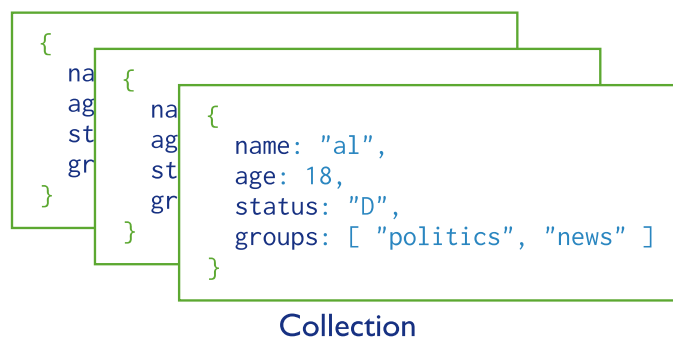
- Khó khăn khi áp dụng cho các dự án lớn: Công cụ này chủ yếu phù hợp cho các dự án vừa và nhỏ. Những dự án lớn sử dụng Express đòi hỏi nguồn nhân lực hiểu biết sâu về Js, điều đó sẽ ảnh hưởng trong việc tuyển dụng nhân viên.

Lý do chọn Express: Đồ án của nhóm thuộc loại vừa và nhỏ thích hợp với công nghệ Express. Ngoài ra, vì NodeJS và ExpressJS được viết bằng JavaScript, một ngôn ngữ rất dễ học và thao tác, framework này có khả năng mở rộng và linh hoạt cao. Ta có thể xây dựng trang web, ứng dụng web và thậm chí các ứng dụng di động tương thích đa nền tảng.

4. Database – MongoDB



MongoDB là một hệ cơ dữ liệu phi quan hệ NoSQL, mã nguồn mở. Thay vì sử dụng các table để lưu trữ dữ liệu giống như các Relational Database, MongoDB sử dụng một khái niệm khác được gọi là collection. Nếu như trong Relational Database, các table được tạo nên từ các records (rows) thì trong MongoDB, các collections sẽ chứa các document. Các document này được lưu trữ dưới định dạng BSON (Binary JSON). Hình 1 ví dụ các documents trong một collection.



Hình 4.1.1: Mô tả một collection.

Điều làm nên đặc biệt của NoSQL và MongoDB là các collection (trong mongoDB) được cấu trúc một cách linh hoạt, cho phép các dữ liệu lưu trữ không cần theo một cấu trúc nhất định, giúp các developer phát triển ứng dụng một cách nhanh chóng, đồng thời hệ thống có thể cập nhật, thay đổi cấu trúc DB một cách linh hoạt.

Một số câu lệnh cơ bản trên MongoDB

CSDL	MySQL	MongoDB
Tạo csdl	CREATE DATABASE test;	<code>use test;</code>
Tạo bảng	CREATE TABLE students (ten_cot - kieu_du_lieu);	<code>db.createCollection('students');</code>
Tạo bản ghi	INSERT INTO studetns ('name', 'gender') VALUES('thanh', 'male');	<code>db.students.insert({ name:'thanh', gender:'male' });</code>
Cập nhật	UPDATE students SET name = 'thanh update' WHERE id = 1;	<code>db.students.update({ _id: 1 },{\$set:{ name: 'thanh update' }});</code>
Xóa bản ghi	DELETE FROM students Where id = 1;	<code>db.students.remove({ _id: 1 });</code>
Tìm kiếm all	SELECT * FROM students;	<code>db.students.find({});</code>
Tìm kiếm	SELECT * FROM students WHERE name = 'thanh';	<code>db.students.find({ name: 'thanh' });</code>

Ưu điểm của mongoDB

- MongoDB sử dụng cơ chế lưu trữ dưới dạng Document BSON, nên mỗi một collection và các documets sẽ có các kích cỡ khác nhau, linh hoạt trong việc lưu trữ dữ liệu lần insert dữ liệu mới có cấu trúc khác với các dữ liệu ban đầu.
- Dữ liệu trong MongoDB không có các ràng buộc Primary Key hay Forgein Key, ... Do đó sẽ không có phép Join như trong Relational Database (phép Join này được hiện thực trên application layer), các phép update hay delete không cần phải kiểm tra ràng buộc mà có thể thực hiện ngay lập tức.
- Vì là một distributed system nên MongoDB rất dễ mở rộng bằng cách thêm các node vào trong hệ thống.
- Trong mỗi document luôn có một trường “_id” do chính mongoDB thêm vào để phân biệt giữa các documents. Trường “_id” này được tự động đánh index để tăng tốc độ truy vấn thông tin nhằm mục đích đạt hiệu suất cao nhất.

Nhược điểm của mongoDB

- Bởi vì không có các ràng buộc nên khi insert dữ liệu chúng ta cần phải cẩn thận.
- Không hỗ trợ phép join và các ràng buộc nên dẫn đến dư thừa dữ liệu.
- MongoDB thuộc lớp lazy evaluation, do đó khi insert/update/delete các document, MongoDB chưa thực hiện ngay lập tức xuống đĩa cứng mà chỉ thực hiện ở trên RAM. Nếu có sự cố như mất điện, ... dẫn đến mất dữ liệu.

Khi nào nên sử dụng MongoDB

Các hệ thống yêu cầu phản hồi nhanh, đáp ứng các request ngay lập tức (real time) thì NoSQL hay MongoDB có thể phù hợp.

5. Library

5.1. MQTT.js

MQTT.js là thư viện MQTT protocol cho client được viết bằng JavaScript cho cả Node.js và browser. MQTT.js bao gồm các phương thức gửi, nhận gọi tin và tự động kết nối lại với message broker.

Các phương thức quan trọng:

- Kết nối đến broker thông qua API `mqtt.connect(url, [options])`

```
1. const mqtt = require("mqtt");
2.
3. let client = mqtt.connect("tcp:// 168.63.137.21:1883", {
4.   username: "Bkvm2",
5.   password: "Hcmut_CSE_2020"
6. });
7.
8. client.on("connect", function () {
9.   console.log("connected to mqtt broker");
10. });
```

- Subscribe topic thông qua API `mqtt.subscribe(topic, [options])` và nhận thông tin qua sự kiện "message"

```
1. client.subscribe("Topic/Mois", { qos: 2 }, function (err) {
2.   if (error) {
3.     console.error(err);
4.   }
5. });
6.
7. client.on("message", function (topic, messageBuffer) {
8.   let messageString = messageBuffer.toString();
9.   let messageJSON = JSON.parse(messageString);
10. console.log("topic: ", messageJSON);
11. });
```

- Publish topic thông qua API `mqtt.publish(topic, message, [options])`

```
1. let messageJSON = {
2.   device_id: "Speaker",
3.   values: ["2233", "1"]
4. };
5. let messageString = JSON.stringify(messageJSON);
6.
7. client.publish("Topic/Speaker", messageString, { qos: 2 });
```

5.2. Socket.io

Socket.IO là một bộ thư viện dành cho các ứng dụng web, mobile để phát triển các ứng dụng realtime. Với đặc trưng mạnh mẽ và dễ sử dụng. 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

Hầu hết các trình duyệt hiện nay đã hỗ trợ websocket nên việc sử dụng socket.io trên trình duyệt cũng là đang sử dụng websocket Việc sử dụng socket.io rất đơn giản và giống nhau ở cả client lẫn server nó bao gồm 3 phần chính: Khởi tạo kết nối, lắng nghe event, gửi event.

Riêng ở server thì sẽ không có phần khởi tạo kết nối vì chỉ có client mới cần khởi tạo kết nối đến server. Việc dùng socket.io đồng bộ ở cả client lẫn server, cú pháp cũng khá đơn giản.

Các phương thức được sử dụng:

1. `const server = http.createServer(app);`
2. `const io = require("socket.io")(server);`

Dòng 1 dùng phương thức `createServer` trong HTTP module để khởi tạo server. Dòng 2 để tạo một cá thể socket.io gắn với máy chủ HTTP vừa được khởi tạo.

1. `io.use(authorizationSocket);`
2. `io.setMaxListeners(Infinity);`

Dòng 1 để cấu hình cách xác thực socket khi client kết nối đến server. Khi cài đặt socket.io trong nodejs, mặc định mọi client có thể kết nối tới socket server đó. Để xác thực và ủy quyền cho những người dùng có quyền sử dụng kết nối, ta cần cấu hình thêm. Dòng 2 để tăng số lượng listeners vì mặc định node.js cho phép 10 listeners trên cùng một sự kiện.

```
io.of("/dashboard").on("connect", func);
```

Sử dụng phương thức `of` để khởi tạo và truy xuất đến namespace dashboard. Nếu namespace dashboard đã có khởi tạo thì nó sẽ trả về ngay lập tức. Việc sử dụng namespace rất có ích để tối thiểu số kết nối TCP cùng một lúc. Ngoài ra, namespace chia sẻ chung kết nối WebSockets có thể giúp chúng ta tiết kiệm port socket trên server. Sử dụng phương thức `on` để lắng nghe sự kiện connection, khi sự kiện xảy ra hàm func sẽ được thực hiện.

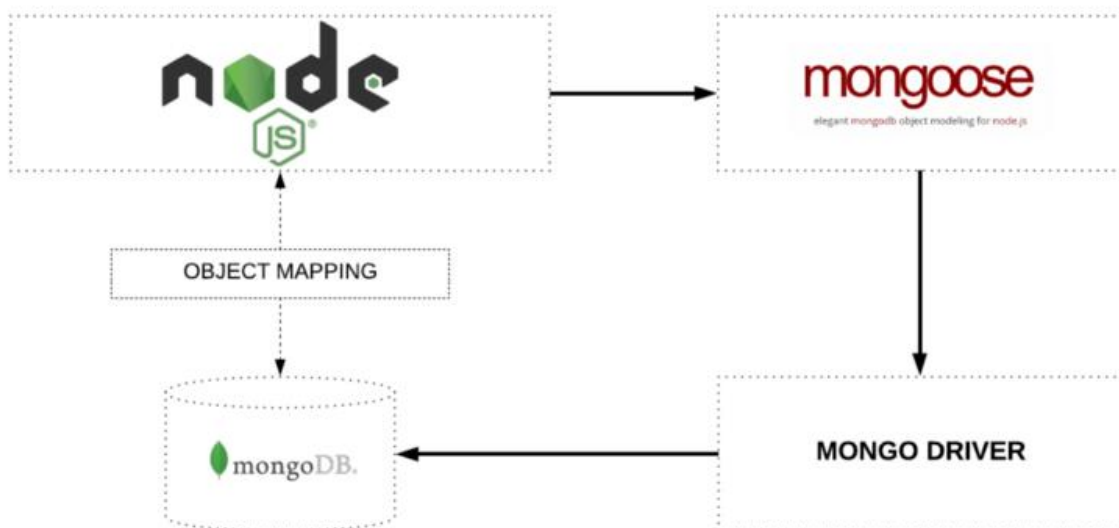
```
const chat = io.connect("/controller");
```

Sử dụng phương thức `connect` của `socket.io` để client khởi tạo kết nối socket đến các namespace đã được tạo ở server.

5.3. Mongoose

mongoose

Mongoose là một thư viện mô hình hóa đối tượng dữ liệu (Object Data Modeling) dùng để query dữ liệu từ MongoDB dành cho Node.js. Nó quản lý mối quan hệ giữa dữ liệu, cung cấp việc xác thực các schema và dùng để chuyển các object trong code thành các objects trong MongoDB và ngược lại.



Hình 5.3.2: Mối quan hệ giữa Node js, mongoose và mongoDB

MongoDB là một schema – less NoSQL document database. Do đó để hỗ trợ những người dùng quen với SQL, mongoose ra đời, đóng vai trò như lớp trung gian giữa ứng dụng và database. Mongoose định nghĩa một khái niệm mới là Mongoose “Schema” là một document data structure.

Sau đây là các phương thức cơ bản của mongoose.

- Database connection

Tạo file `./src/database.js` trong thư mục gốc của project.

Tiếp theo chúng ta thêm một class đơn giản với method kết nối tới database vào trong `./src/database.js`.

```
1. let mongoose = require('mongoose');
2.
3. const server = '127.0.0.1:27017'; // REPLACE WITH YOUR DB SERVER
4. const database = 'fcc-Mail'; // REPLACE WITH YOUR DB NAME
5.
6. class Database {
7.   constructor() {
8.     this._connect()
9.   }
10.
11.  _connect() {
12.    mongoose.connect(`mongodb://${server}/${database}`)
13.      .then(() => {
14.        console.log('Database connection successful')
15.      })
16.      .catch(err => {
17.        console.error('Database connection error')
18.      })
19.  }
20. }
21.
22. module.exports = new Database()
```

- Mongoose Schema vs Model

Mongoose schema định nghĩa cấu trúc của document như default values, validators, ... Trong khi đó Mongoose model cung cấp một interface tới database để create, query, update, delete các records.

Tạo một Mongoose model cơ bản bao gồm 3 phần chính:

Referencing Mongoose

```
let mongoose = require('mongoose')
```

Defining the Schema

Schema sẽ định nghĩa các properties của một document thông qua object `new mongoose.Schema`.

```
1. let emailSchema = new mongoose.Schema({
2.   email: String
3. })
```

Ở đây, một property vừa định nghĩa được gọi là email với schema type là String. Nếu trường này nhận vào một kiểu không phải String nó sẽ báo lỗi.

Những Schema Types sau đây được cho phép trong Mongoose:

- Array
- Boolean
- Buffer
- Date
- Mixed
- Number
- ObjectId
- String

Mixed và ObjectId được định nghĩa trong

```
require('mongoose').Schema.Types
```

Exporting a Model

Chúng ta cần gọi constructor của model trên Mongoose instance và truyền cho nó tên của collection và tham chiếu tới schema vừa định nghĩa.

```
module.exports = mongoose.model('Email', emailSchema)
```

Kết hợp đoạn code trên vào `./src/models/email.js` để định nghĩa contents của một model email đơn giản.

```
1. let mongoose = require('mongoose')
2.
3. let emailSchema = new mongoose.Schema({
4.   email: String
5. })
6.
7. module.exports = mongoose.model('Email', emailSchema)
```


- Basic Operations

Create Record

Tạo một instance của model email và lưu nó vào trong database.

```
1. let EmailModel = require('./email')
2.
3. let msg = new EmailModel({
4.   email: 'ADA.LOVELACE@GMAIL.COM'
5. })
6.
7. msg.save()
8.   .then(doc => {
9.     console.log(doc)
10.  })
11.  .catch(err => {
12.    console.error(err)
13.  })
```

Kết quả được trả về là một document:

```
{
  _id: 5a78fe3e2f44ba8f85a2409a,
  email: 'ada.lovelace@gmail.com',
  __v: 0
}
```

Fetch Record

Để tìm những record mà chúng ta vừa tạo, sử dụng find method và truyền email mà chúng ta cần tìm.

```
1. EmailModel
2.   .find({
3.     email: 'ada.lovelace@gmail.com' // search query
4.   })
5.   .then(doc => {
6.     console.log(doc)
7.   })
8.   .catch(err => {
9.     console.error(err)
10.  })
```

Kết quả trả về.

```
{
  _id: 5a78fe3e2f44ba8f85a2409a,
  email: 'ada.lovelace@gmail.com',
  __v: 0
}
```

```
}
```

Update Record

Chúng ta sẽ sửa record ở trên bằng cách thay đổi địa chỉ email và thêm một field khác vào nó.

```
EmailModel
.findOneAndUpdate(
  {
    email: 'ada.lovelace@gmail.com' // search query
  },
  {
    email: 'theoutlander@live.com' // field: values to update
  },
  {
    new: true,           // return updated doc
    runValidators: true  // validate before update
  })
.then(doc => {
  console.log(doc)
})
.catch(err => {
  console.error(err)
})
```

Kết quả trả về:

```
{
  _id: 5a78fe3e2f44ba8f85a2409a,
  email: 'theoutlander@live.com',
  __v: 0
}
```

Delete Record

Chúng ta sẽ dùng phương thức `findOneAndRemove` để xóa một record. Phương thức này sẽ trả về document nguyên bản vừa xóa.

```
1. EmailModel
2. .findOneAndRemove({
3.   email: 'theoutlander@live.com'
4. })
5. .then(response => {
6.   console.log(response)
7. })
8. .catch(err => {
9.   console.error(err)
10. })
```

Query Building

Mongoose có rất nhiều API để xử lý các phép toán phức tạp được hỗ trợ bởi MongoDB. Hãy xem xét query sau mà chúng ta có thể tăng dần các thành phần truy vấn.

Trong ví dụ này, chúng ta sẽ:

- Tìm tất cả user
- Skip 100 records đầu tiên
- Giới hạn kết quả trả về chỉ 10 records
- Sắp xếp kết quả theo field firstName
- Chọn firstName
- Thực hiện query

```
1. UserModel.find()           // find all users
2.   .skip(100)               // skip the first 100 items
3.   .limit(10)               // limit to 10 items
4.   .sort({firstName: 1})    // sort ascending by firstName
5.   .select({firstName: true}) // select firstName only
6.   .exec()                  // execute the query
7.   .then(docs => {
8.     console.log(docs)
9.   })
10.  .catch(err => {
11.    console.error(err)
12.  });
```

5.4. Agenda

Agenda là thư viện nhỏ nhẹ định thời cho Node.js và đáng tin cậy. Agenda cũng cấp những tính năng mạnh mẽ như định thời có độ ưu tiên, đồng thời và lặp lại. Agenda hỗ trợ database mongoDB trong việc quản lý jobs.

Tạo và kết nối database:

```
1. async function initAgenda(url){
2.   let agenda = new Agenda();
3.   agenda.database(url, 'agenda', { useUnifiedTopology: true });
4.   await agenda.start();
5.   return agenda;
6. }
7.
8. initAgenda("MONGODB_URL")
9. .catch(function (agenda){
10.   console.log("Connected to MongoDB");
11. }).catch(function(error){
12.   console.error(error);
13. });
```

Định nghĩa một job:

```
1. agenda.define("watering", function(job){
2.   let {intensity} = job.attrs.data;
3.   console.log(intensity);
4. });
```

Đặt lịch thực hiện tại một thời điểm theo job đã định nghĩa:

```
1. let data = { intensity: 5000 };
2. agenda.schedule("in 2 minutes", "watering", data);
```

Quản lý jobs, thông qua query dạng mongo native:

```
1. agenda.jobs({name: "watering"});
```

5.5. jQuery

jQuery là thư viện đa trình duyệt được thiết kế để đơn giản hóa lập trình phía máy người sử dụng HTML được viết từ JavaScript. jQuery giúp xây dựng các chức năng bằng Javascript dễ dàng, nhanh và giàu tính năng hơn. jQuery được tích hợp nhiều module khác nhau từ module hiệu ứng cho đến module truy vấn selector. Các module phổ biến của jQuery bao gồm:

- Ajax – xử lý Ajax.
- Attributes – Xử lý các thuộc tính của đối tượng HTML.
- Effect – xử lý hiệu ứng.
- Event – xử lý sự kiện.
- Form – xử lý sự kiện liên quan tới form.

- DOM – xử lý Data Object Model.
- Selector – xử lý luồng lách giữa các đối tượng HTML.

Với các đặc tính nổi trội như đã nêu trên, nhóm chọn jQuery do nó cung cấp nhiều API hữu ích, tiện lợi, đồng thời hoạt động tốt trên nhiều loại trình duyệt khác nhau. Mà cụ thể là jQuery 3.4.1.

Hướng dẫn thêm jQuery 3.4.1 vào dự án.

Bước 1: Lưu file jquery-3.4.1.min.js vào máy tính từ đường dẫn <https://code.jquery.com/jquery-3.4.1.min.js>

Bước 2: Khai báo sử dụng jQuery, trỏ nguồn tới vị trí chứa file jquery-3.4.1.min.js mà bạn vừa tải về tương tự như sau.

```
<script src="/public/js/jquery/jquery-3.4.1.min.js"></script>
```

5.6. Bootstrap

Bootstrap là một framework bao gồm các HTML, CSS và JavaScript template miễn phí, mã nguồn mở hướng vào việc phát triển website chuẩn responsive. Bootstrap bao gồm các HTML templates, CSS templates và Javascript tạo ra những cái cơ bản có sẵn như: typography, forms, buttons, tables, grid, navigation, modals, image carousels và nhiều thứ khác. Trong bootstrap có thêm các plugin Javascript trong nó. Bootstrap có những đặc điểm nổi bật như sau:

- Dễ dàng thao tác: Cơ chế hoạt động của Bootstrap là dựa trên xu hướng mã nguồn mở HTML, CSS và Javascript. Người dùng cần trang bị kiến thức cơ bản 3 mã này mới có thể sử dụng Bootstrap hiệu quả. Bên cạnh đó, các mã nguồn này cũng có thể dễ dàng thay đổi và chỉnh sửa tùy ý.
- Tùy chỉnh dễ dàng: Bootstrap được tạo ra từ các mã nguồn mở cho phép designer linh hoạt hơn. Giờ đây có thể lựa chọn những thuộc tính, phần tử phù hợp với dự án họ đang theo đuổi. CDN Bootstrap còn giúp bạn tiết kiệm dung lượng vì không cần tải mã nguồn về máy.
- Chất lượng sản phẩm đầu ra tốt: Bootstrap là sáng tạo của các lập trình viên giỏi trên khắp thế giới. Bootstrap đã được nghiên cứu và thử nghiệm trên các thiết bị. Được kiểm tra nhiều lần trước khi đưa vào sử dụng. Do đó, khi chọn Bootstrap, bạn có thể tin rằng mình sẽ tạo nên những sản phẩm với chất lượng tốt nhất.
- Độ tương thích cao: Điểm cộng lớn nhất của Bootstrap là khả năng tương thích với mọi trình duyệt và nền tảng. Đây là một điều cực kì quan trọng và cần thiết trong trải nghiệm người dùng. Sử dụng Grid System cùng với hai bộ tiền xử lý Less và Sass, Bootstrap mặc định hỗ trợ Responsive và ưu tiên cho các giao diện trên thiết bị

di động hơn. Bootstrap có khả năng tự động điều chỉnh kích thước trang website theo khung browser. Mục đích để phù hợp với màn hình của máy tính để bàn, tablet hay laptop.

Nhóm lựa chọn bootstrap do dễ sử dụng do được xây dựng cơ bản trên HTML, CSS và Javascript. Mặc khác nó có tính năng Responsive trên các thiết bị phones, tablets, desktops và tương thích với nhiều trình duyệt (Chrome, Firefox, Internet Explorer, Safari, and Opera). Cụ thể là Bootstrap-v4.4.1

Hướng dẫn thêm Bootstrap-v4.4.1 vào dự án.

Bước 1: Tải Bootstrap-v4.4.1 về máy tính từ đường dẫn <https://getbootstrap.com/docs/4.4/getting-started/download/>

Bước 2: Tiến hành giải nén file vừa tải về và tìm file bootstrap.min.css trong thư mục css. Nhúng file bootstrap.min.css vào dự án.

```
<link rel="stylesheet" href="/public/css/bootstrap/bootstrap.min.css">
```

Tuy nhiên, nhiều thành phần trong bootstrap yêu cầu sử dụng JavaScript để hoạt động, ví dụ như: Alerts, dismissing, modals, dropdowns, navbar, scrollspy,... Vì vậy cần phải nhúng thêm file bootstrap.bundle.min.js trong thư mục js vào dự án và ở phía dưới jQuery.

```
<script src="/public/js/bootstrap/bootstrap.bundle.min.js"></script>
```

6. Giả lập thiết bị

Project giả lập thiết bị [devicesfaker](https://bitbucket.org/vanducvo/devicesfaker)² sử dụng công nghệ Node.js, Websocket, MQTT protocol

Tính năng:

- Giả lập cảm biến độ ẩm đất và gửi đến broker theo thời gian thiết đặt.
- Giả lập loa nhận message gửi đến thay đổi thông số
- Dashboard theo dõi độ ẩm và điều khiển loa

Devices Faker

Soil Moisture	status	value
Mois	1	715

Speaker	status	value
Speaker	0	0

Speaker ID

Speaker

Level

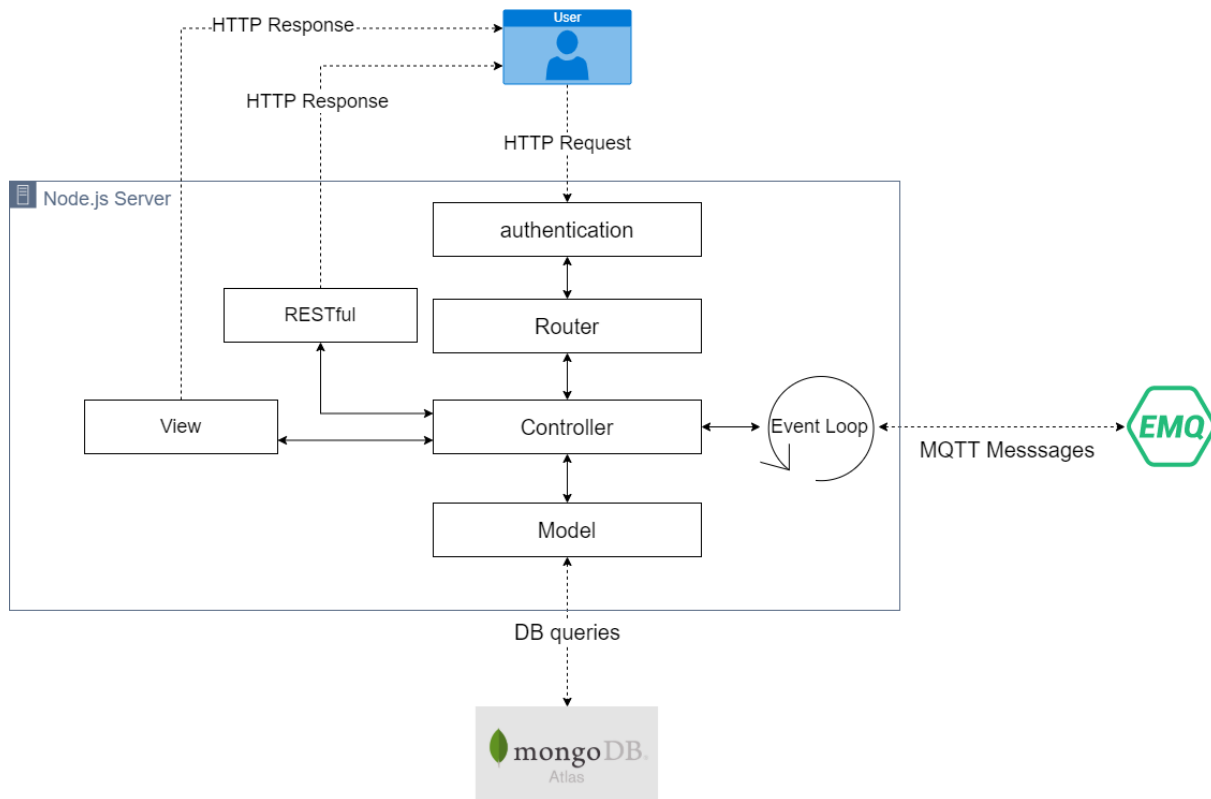
☐ OFF

Control Motor

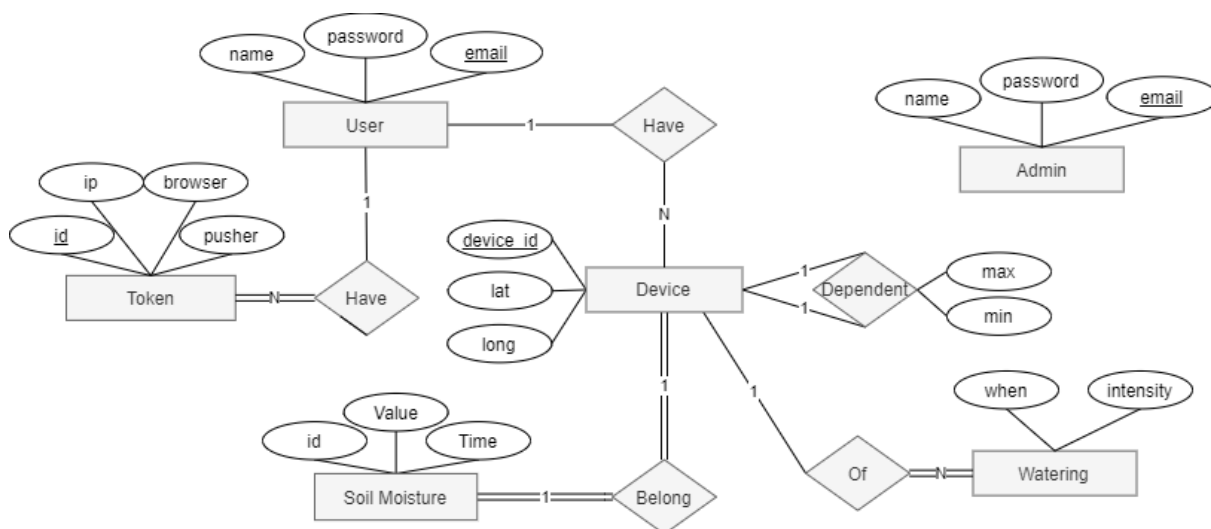
² <https://bitbucket.org/vanducvo/devicesfaker>

IV. Thiết Kế Hệ Thống

1. Thiết kế tổng quan

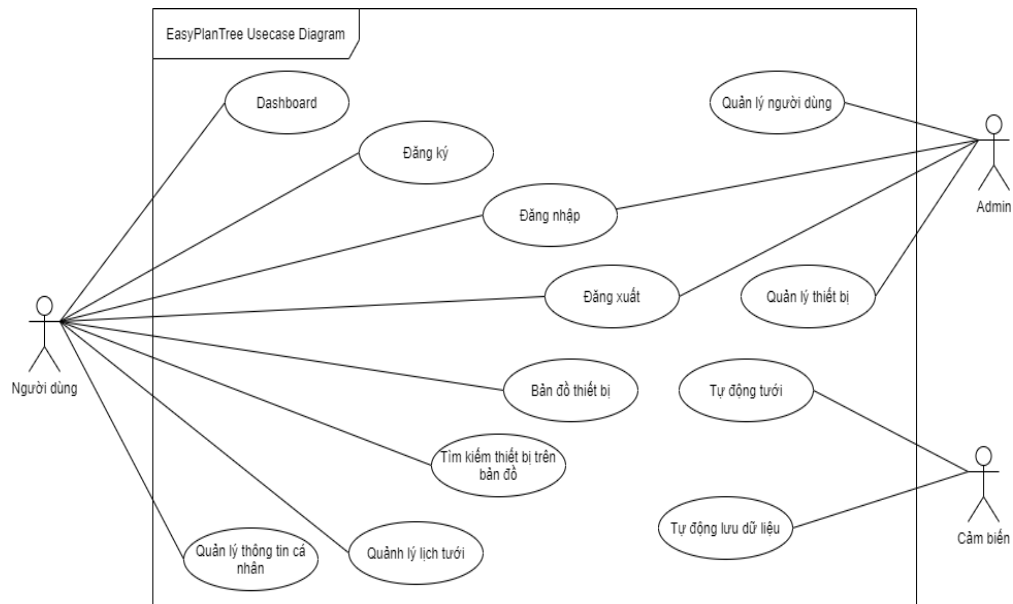


2. Thiết kế database



V. Mô Tả Chi Tiết

1. Usecase



2. Usecase chi tiết

2.1. Quản lí lịch tưới

EasyPlantTree: Quản lí lịch tưới	
Tác nhân	Người dùng
Mô tả	Cho phép người dùng hiển thị lịch tưới trong quá khứ và tương lai, đặt lịch tưới , xóa lịch tưới trong tương lai cho từng thiết bị riêng biệt
Dữ liệu	Thông tin lịch tưới: Ngày giờ tưới, cường độ tưới, thiết bị tưới (motor)
Tác nhân kích thích	
Phản hồi	Lịch tưới đã được đặt hoặc xóa
Lời bình	Cập nhật hiển thị lịch tưới hiện tại và quá khứ theo thời gian thực

2.2. Tưới tự động

EasyPlantTree: Tưới tự động	
Tác nhân	Cảm biến
Mô tả	Khi thông cảm biến không nằm trong khoảng cho phép thì thực hiện tưới 1000ml/phút trong 20s, không thực hiện tưới nếu có lịch tưới trong 30s sắp tới.
Dữ liệu	Thông số cảm biến
Tác nhân kích thích	
Phản hồi	Bật motor
Lời bình	Ngưỡng thông số cho phép do người dùng thiết đặt cho từng cảm biến

2.3. Quản lí người dùng

EasyPlantTree: Quản lí người dùng	
Tác nhân	Admin
Mô tả	Cho phép admin có quyền thêm, xóa thiết bị cho người dùng; ràng buộc các thiết bị; đặt vị trí các thiết bị; xem, xóa phiên đăng nhập người dùng;
Dữ liệu	Thông tin người dùng liên quan
Tác nhân kích thích	
Phản hồi	Thay đổi cài đặt người dùng
Lời bình	Admin có quyền quản lí tất cả các người dùng.

2.4. Bản đồ thiết bị

EasyPlantTree: Bản đồ thiết bị	
Tác nhân	Người dùng
Mô tả	Người dùng có thể xem tổng thể vị trí các thiết bị và có thể tìm vị trí cụ thể của thiết bị trên bản đồ
Dữ liệu	Thông tin vị trí thiết bị
Tác nhân kích thích	
Phản hồi	Vị trí thiết bị trên bản đồ
Lời bình	

2.5 Nhận và hiển thị dữ liệu từ thiết bị.

a. Theo dõi độ ẩm của từng cảm biến theo thời gian thực.

Easy Plant Tree: Theo dõi độ ẩm của từng cảm biến theo thời gian thực	
Tác nhân	Người dùng
Mô tả	Dựa vào dữ liệu về độ ẩm trong cơ sở dữ liệu, lập biểu đồ theo dõi độ ẩm đất cho từng cảm biến trong 3 giờ gần nhất. Cập nhật dữ liệu theo thời gian thực.
Dữ liệu	Thông tin độ ẩm của các cảm biến trong 3 giờ gần nhất
Tác nhân kích thích	Khi người dùng đăng nhập thành công vào hệ thống.
Phản hồi	Hiện thị trên dashboard của website
Lời bình	Chủ trang trại có thể chuyển đổi theo dõi giữa các cảm biến.

b. Xem độ ẩm trung bình của từng cảm biến trong tuần này và tuần trước.

Easy Plant Tree: Xem độ ẩm trung bình của từng cảm biến trong tuần này và tuần trước	
Tác nhân	Người dùng
Mô tả	Dựa vào dữ liệu về độ ẩm trong cơ sở dữ liệu, lập biểu đồ cột độ ẩm trung bình trong tuần này và tuần trước.
Dữ liệu	Dữ liệu về độ ẩm trong tuần này và tuần trước.
Tác nhân kích thích	Khi người dùng đăng nhập thành công vào hệ thống.

Phản hồi	Hiện thị trên dashboard của website
Lời bình	Chủ trang trại có thể chuyển đổi theo dõi giữa tuần này và tuần trước.

c. Xem số lần tưới trong ngày tương ứng với tuần này và tuần trước.

Easy Plant Tree: Xem số lần tưới trong ngày tương ứng với tuần này và tuần trước	
Tác nhân	Người dùng
Mô tả	Thống kê số lần tưới nước các ngày trong tuần của tuần này và tuần trước và biểu diễn dưới dạng biểu đồ cột.
Dữ liệu	Dữ liệu về lịch sử tưới nước trong tuần này và tuần trước.
Tác nhân kích thích	Khi người dùng đăng nhập thành công vào hệ thống.
Phản hồi	Hiện thị trên dashboard của website
Lời bình	Chủ trang trại có thể chuyển đổi theo dõi giữa tuần này và tuần trước.

2.6 Quản lý thông tin cá nhân:

a. Hiện thị thông tin người dùng.

Easy Plant Tree: Hiện thị thông tin người dùng	
Tác nhân	Người dùng
Mô tả	Hiện thị thông tin người dùng gồm: Họ và tên, email.
Dữ liệu	Dữ liệu về người dùng.
Tác nhân kích thích	Khi người dùng bấm vào button Quản lý tài khoản
Phản hồi	Hiện thị trên trang cá nhân của người dùng.
Lời bình	

b. Hiện thị phiên đăng nhập.

Easy Plant Tree: Hiện thị phiên đăng nhập	
Tác nhân	Người dùng
Mô tả	Hiện thị thông tin lịch sử các phiên đăng nhập gồm: ID phiên đăng nhập, IP, thông tin về trình duyệt sử dụng.
Dữ liệu	Dữ liệu về lịch sử đăng nhập.
Tác nhân kích thích	Khi người dùng bấm vào button Quản lý tài khoản

Phản hồi	Hiện thị trên trang cá nhân của người dùng.
Lời bình	

c. Xóa phiên đăng nhập:

Easy Plant Tree: Xóa phiên đăng nhập	
Tác nhân	Người dùng
Mô tả	Người dùng có thể xóa các phiên đăng nhập trước. Tuy nhiên không thể xóa phiên đăng nhập hiện tại.
Dữ liệu	Dữ liệu về lịch sử đăng nhập.
Tác nhân kích thích	Khi người dùng bấm vào button xóa phiên đăng nhập
Phản hồi	Cập nhật lại trang cá nhân của người dùng.
Lời bình	

d. Hiện thị thiết bị và quan hệ phụ thuộc các thiết bị.

Easy Plant Tree: Hiện thị thiết bị và quan hệ phụ thuộc các thiết bị	
Tác nhân	Người dùng
Mô tả	Hiện thị danh sách các thiết bị và ràng buộc của nó. Các nội dung hiển thị bao gồm: Tên thiết bị, kinh độ, vĩ độ, loại cảm biến, các thiết bị phụ thuộc, ngưỡng cho phép.
Dữ liệu	Dữ liệu về thiết bị và dữ liệu về tính ràng buộc của thiết bị đó.
Tác nhân kích thích	Khi người dùng bấm vào button Quản lý tài khoản
Phản hồi	Hiện thị trên trang cá nhân của người dùng.
Lời bình	

e. Đặt ngưỡng độ ẩm cho phép cho cảm biến.

Easy Plant Tree: Đặt ngưỡng độ ẩm cho phép cho cảm biến	
Tác nhân	Người dùng
Mô tả	Người dùng có thể điều chỉnh ngưỡng của cảm biến độ ẩm.
Dữ liệu	Dữ liệu về tính ràng buộc của thiết bị đó.
Tác nhân kích thích	Khi người dùng bấm vào button Sửa hoặc Xóa ngưỡng thiết bị.
Phản hồi	Hiện thị lại ngưỡng mới trên trang cá nhân của người dùng.
Lời bình	

2.7 Hiện thị thông tin tổng quan:

a. Thống kê số lượng thiết bị:

Easy Plant Tree: Thống kê số lượng thiết bị	
Tác nhân	Admin
Mô tả	Thống kê số lượng thiết bị.
Dữ liệu	Dữ liệu về thiết bị.
Tác nhân kích thích	Khi admin đăng nhập thành công vào hệ thống.
Phản hồi	Hiện thị trên dashboard của admin.
Lời bình	

b. Thống kê số lượng người dùng:

Easy Plant Tree: Thống kê số lượng người dùng	
Tác nhân	Admin
Mô tả	Thống kê số lượng người dùng.
Dữ liệu	Dữ liệu về người dùng.
Tác nhân kích thích	Khi admin đăng nhập thành công vào hệ thống.
Phản hồi	Hiện thị trên dashboard của admin.
Lời bình	

c. Hiện thị danh sách người dùng:

Easy Plant Tree: Hiện thị danh sách người dùng	
Tác nhân	Admin
Mô tả	Hiện thị bảng danh sách thông tin người dùng gồm: họ và tên, email.
Dữ liệu	Dữ liệu về người dùng.
Tác nhân kích thích	Khi admin đăng nhập thành công vào hệ thống.
Phản hồi	Hiện thị trên dashboard của admin.
Lỗi bình	

d. Hiện thị danh sách thiết bị:

Easy Plant Tree: Hiện thị danh sách thiết bị	
Tác nhân	Admin
Mô tả	Hiện thị bảng danh sách thông tin các thiết bị của hệ thống gồm: tên thiết bị, loại thiết bị.
Dữ liệu	Dữ liệu về thiết bị.
Tác nhân kích thích	Khi admin đăng nhập thành công vào hệ thống.
Phản hồi	Hiện thị trên dashboard của admin.
Lỗi bình	

e. Tìm kiếm người dùng:

Easy Plant Tree: Tìm kiếm người dùng	
Tác nhân	Admin
Mô tả	Admin có thể tìm kiếm người dùng trong bảng danh sách người dùng bằng cách tìm theo tên hoặc tìm theo email. Kết quả tìm kiếm được cập nhật động theo từ khóa tìm kiếm.
Dữ liệu	
Tác nhân kích thích	Admin nhập từ khóa cần tìm kiếm trong bảng danh sách người dùng
Phản hồi	Hiện thị lại danh sách người dùng có liên quan đến từ khóa
Lỗi bình	

f. Tìm kiếm thiết bị:

Easy Plant Tree: Tìm kiếm thiết bị	
Tác nhân	Admin
Mô tả	Admin có thể tìm kiếm thiết bị trong bảng danh sách các thiết bị bằng cách tìm theo tên hoặc tìm theo phân loại thiết bị. Kết quả tìm kiếm được cập nhật động theo từ khóa tìm kiếm.
Dữ liệu	
Tác nhân kích thích	Admin nhập từ khóa cần tìm kiếm trong bản danh sách thiết bị.
Phản hồi	Hiện thị lại danh sách người dùng có liên quan đến từ khóa
Lời bình	

2.5. Quản lí thiết bị

Easy Plant Tree: Quản lý thiết bị.	
Tác nhân	Admin.
Mô tả	Cho phép quản lý xem các thông tin về thiết bị của người dùng. Ngoài ra, quản lý còn có thể thêm và xoá các thiết bị trong hệ thống.
Dữ liệu	Thông tin về các thiết bị.
Tác nhân kích thích	
Phản hồi	Thay đổi cài đặt của người dùng.
Lời bình	

2.6. Đăng xuất

Easy Plant Tree: Đăng xuất	
Tác nhân	Người dùng, Admin

Mô tả	Cho phép người dùng, admin đăng xuất khỏi tài khoản đã đăng nhập.
Dữ liệu	
Tác nhân kích thích	
Phản hồi	Đăng nhập thành công.
Lời bình	

2.7. Đăng ký

Easy Plant Tree: Đăng ký.	
Tác nhân	Người dùng
Mô tả	Cho phép người dùng đăng ký tài khoản để sử dụng ứng dụng. Khi đăng ký sẽ có email đến để yêu cầu xác thực.
Dữ liệu	Email đăng ký, mật khẩu.
Tác nhân kích thích	
Phản hồi	Đã xác thực thành công.
Lời bình	Nếu email đã được đăng ký thì email đó sẽ không được đăng ký tiếp nữa.

2.8. Đăng nhập

Easy Plant Tree: Đăng nhập	
Tác nhân	Người dùng, Admin
Mô tả	Cho phép người dùng, admin đăng nhập để sử dụng ứng dụng với tài khoản email đã đăng ký.

Dữ liệu	Email đăng nhập, mật khẩu.
Tác nhân kích thích	
Phản hồi	Đăng nhập thành công.
Lỗi bình	

3. Yêu cầu chức năng

3.1. Hiện thị vị trí thiết bị trên bản đồ

Tên usecase	Hiện thị vị trí sensor trên bản đồ
Mô tả	Người dùng bấm vào mục “Bản đồ thiết bị” để xem vị trí của các sensor trên bản đồ
Điều kiện tiên quyết	Vị trí của các sensor
Quy trình thông thường	<ol style="list-style-type: none"> 1. Người dùng bấm vào mục “Bản đồ thiết bị”. 2. Server truy vấn tất cả các thiết bị thuộc người dùng kể cả thiết bị không có vị trí 3. Tạo trang gồm các danh sách các thiết bị và phản hồi về cho người dùng 4. Hiện thị lên trình duyệt và thông qua danh sách đánh dấu vị trí thiết bị lên bản đồ
Ngoại lệ	
Quy trình thay thế	<p>Ngoại lệ 1: Bước 2</p> <p>32. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500</p>

3.2. Tìm kiếm thiết bị trên bản đồ

Tên usecase	Tìm kiếm thiết bị trên bản đồ
Mô tả	Người dùng xem vị trí cụ thể của từng sensor trên bản đồ
Điều kiện tiên quyết	Người dùng đang ở trang “Bản đồ thiết bị”
Quy trình thông thường	<ol style="list-style-type: none">1. Người dùng bấm vào nút “tìm kiếm” ở bên cạnh các sensor trên mục danh sách thiết bị.2. Hệ thống sẽ hiển thị vị trí cụ thể sensor mà người dùng đã chọn trên bản đồ.
Ngoại lệ	
Quy trình thay thế	2a. Nếu thiết bị không có vị trí thì không hiển thị thiết bị trên bản đồ

3.3. Hiển thị lịch tưới

Tên use case	Hiển thị lịch tưới
Mô tả	Khi người dùng vào trang “Hẹn giờ tưới” server sẽ hiển thị tất cả lịch tưới sắp tới và đã xảy ra. Khi có lịch tưới xảy ra sẽ cập nhật realtime.
Điều kiện tiên quyết	Người dùng đăng nhập
Quy trình thông thường	<ol style="list-style-type: none">1. Người dùng truy cập trang hẹn lịch tưới2. Server truy vấn tất cả lịch tưới của người dùng3. Server tạo trang và phản hồi cho người dùng gồm tất cả thông tin về lịch tưới sắp tới và đã xảy ra, các motor của người dùng4. Hiển thị trang cho người dùng
Ngoại lệ	Ngoại lệ 1: Bước 4 3b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500
Quy trình thay thế	<ol style="list-style-type: none">1. Lịch tưới được thực hiện dưới server2. Gửi lịch tưới đã thực hiện cho client thông qua WebSocket3. Thực hiện đưa lịch tưới đã thực hiện vào lịch tưới đã xảy ra

3.4. Xóa lịch tưới

Tên use case	Xóa lịch tưới
Mô tả	Người dùng xóa lịch tưới đã đặt.
Điều kiện tiên quyết	Người dùng đăng nhập và vào trang “Hẹn lịch tưới”
Quy trình thông thường	<ol style="list-style-type: none">1. Người dùng bấm xóa lịch tưới cần xóa2. Gửi thông tin lịch tưới cho Server3. Server tạo truy vấn xóa lịch tưới trong Module Agenda4. Gửi thông tin lịch tưới đã xóa đến người dùng5. Cập nhật lịch tưới sắp tới
Ngoại lệ	Ngoại lệ 1: Bước 4 3b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500
Quy trình thay thế	

3.5. Đặt lịch tưới

Tên use case	Đặt lịch tưới
Mô tả	Người dùng xóa lịch tưới đã đặt.
Điều kiện tiên quyết	Người dùng đăng nhập và vào trang “Hẹn lịch tưới”
Quy trình thông thường	<ol style="list-style-type: none">1. Người dùng bấm xóa lịch tưới cần xóa2. Gửi thông tin lịch tưới cho Server3. Server tạo truy vấn xóa lịch tưới trong Module Agenda4. Gửi thông tin lịch tưới đã xóa đến người dùng5. Cập nhật lịch tưới sắp tới
Ngoại lệ	Ngoại lệ 1: Bước 4 3b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500
Quy trình thay thế	

3.6. Đặt lịch tưới

Tên use case	Đặt lịch tưới
Mô tả	Người dùng đặt lịch tưới
Điều kiện tiên quyết	Người dùng đăng nhập và vào trang “Hẹn lịch tưới”
Quy trình thông thường	<ol style="list-style-type: none">1. Người dùng điền các thông tin cần thiết vào form bao gồm: ngày, giờ, thiết bị, cường độ2. Người dùng bấm “Bơm”3. Nếu lịch đặt trong quá khứ báo lỗi, nếu không yêu cầu xác nhận từ người dùng4. Người dùng xác nhận sau đó gửi lịch đến Server5. Server kiểm tra thiết bị thuộc sở hữu người dùng, kiểm tra lịch có bị chồng lên lịch khác6. Kiểm tra thất bại, phản hồi lỗi, phản hồi lịch tưới7. Nếu thành công cập nhật lịch tưới nếu thất bại hiện thông báo nguyên nhân lỗi
Ngoại lệ	Ngoại lệ 1: Bước 5 3b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500
Quy trình thay thế	

3.7. Tự động lưu thông số từ cảm biến

Tên use case	Tự động lưu thông số từ cảm biến
Mô tả	Khi
Điều kiện tiên quyết	Người dùng đăng nhập và vào trang “Hẹn lịch tưới”
Quy trình thông thường	<ol style="list-style-type: none"> 1. Người dùng điền các thông tin cần thiết vào form bao gồm: ngày, giờ, thiết bị, cường độ 2. Người dùng bấm “Bơm” 3. Nếu lịch đặt trong quá khứ báo lỗi, nếu không yêu cầu xác nhận từ người dùng 4. Người dùng xác nhận sau đó gửi lịch đến Server 5. Server kiểm tra thiết bị thuộc sở hữu người dùng, kiểm tra lịch có bị chồng lên lịch khác 6. Kiểm tra thất bại, phản hồi lỗi, phản hồi lịch tưới 7. Nếu thành công cập nhật lịch tưới nếu thất bại hiện thông báo nguyên nhân lỗi
Ngoại lệ	<p>Ngoại lệ 1: Bước 5</p> <p>3b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500</p>
Quy trình thay thế	

3.8. Hiện thị người dùng

Tên use case	Hiện thị người dùng
Mô tả	Hiện thị tất cả người dùng ứng dụng, cùng với thông tin thiết bị liên quan đến người dùng
Điều kiện tiên quyết	Admin đăng nhập
Quy trình thông thường	<ol style="list-style-type: none"> 1. Người dùng truy cập trang “quản lí người dùng” 2. Server truy vấn tất cả người dùng, thiết bị của người dùng 3. Tạo nội dung trang và phản hồi về cho người dùng 4. Hiện thị nội dung trang lên trình duyệt
Ngoại lệ	<p>Ngoại lệ 1: Bước 3</p> <p>3b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500</p>
Quy trình thay thế	

3.9. Hiện thị phiên đăng nhập người dùng

Tên use case	Hiện thị phiên đăng nhập người dùng
Mô tả	Khi Admin bấm “Thông tin” thuộc mỗi người dùng sẽ load thông tin phiên đăng nhập từ server và cập nhật vào giao diện
Điều kiện tiên quyết	Admin đăng nhập và vào trang “quản lí người dùng”
Quy trình thông thường	<ol style="list-style-type: none">1. Admin bấm chọn “Thông tin” tương ứng với người dùng2. Gửi yêu cầu3. Tạo nội dung trang và phản hồi về cho người dùng4. Hiện thị nội dung trang lên trình duyệt
Ngoại lệ	Ngoại lệ 1: Bước 3 3b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500
Quy trình thay thế	

3.10. Xóa phiên đăng nhập người dùng

Tên use case	Xóa phiên đăng nhập người dùng
Mô tả	Admin có quyền chọn xóa phiên đăng nhập từ người dùng
Điều kiện tiên quyết	Admin đăng nhập và vào trang “quản lí người dùng”, đang hiển thị phiên đăng nhập từ người dùng
Quy trình thông thường	<ol style="list-style-type: none">1. Admin bấm chọn “Xóa” tương ứng với phiên đăng nhập2. Gửi yêu cầu xóa đến Server3. Tạo truy vấn xóa đến database4. Trả về thông tin đã xóa5. Nếu xóa thành công cập nhật lên giao diện người dùng
Ngoại lệ	Ngoại lệ 1: Bước 3 3b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500
Quy trình thay thế	

3.11. Ràng buộc thiết bị với người dùng

Tên use case	Ràng buộc thiết bị với người dùng
Mô tả	Admin có quyền thêm thiết bị cho người dùng.
Điều kiện tiên quyết	Admin đăng nhập và vào trang “quản lý người dùng
Quy trình thông thường	<ol style="list-style-type: none">1. Admin bấm chọn “+” trên trường “thiết bị”2. Chọn loại thiết bị3. Tự động gửi yêu cầu danh sách thiết bị khả dụng4. Tạo truy vấn đến database lấy tất cả thiết bị khả dụng5. Cập nhật các lựa chọn thiết bị6. Admin chọn thiết bị cần thêm sau đó nhấp “Thêm”7. Gửi yêu cầu thêm thiết bị cho người dùng đến Server8. Cập nhật người sở hữu cho thiết bị9. Cập nhật lên giao diện cho admin
Ngoại lệ	<p>Ngoại lệ 1: Bước 3</p> <p>3b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500</p> <p>Ngoại lệ 1: Bước 8</p> <p>8b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500</p>

3.12. Xóa ràng buộc thiết bị với người dùng

Tên use case	Xóa ràng buộc thiết bị với người dùng
Mô tả	Admin có quyền xóa ràng buộc giữa thiết bị và người dùng
Điều kiện tiên quyết	Admin đăng nhập và vào trang “quản lí người dùng”
Quy trình thông thường	<ol style="list-style-type: none"> 1. Admin bấm chọn “X” tương ứng thiết bị trên người dùng 2. Yêu cầu xác nhận xóa tới Admin 3. Nếu Đồng ý, Gửi yêu cầu xóa đến Server 4. Tạo truy vấn xóa đến database 5. Trả về thông tin đã xóa 6. Nếu xóa thành công cập nhật lên giao diện người dùng
Ngoại lệ	<p>Ngoại lệ 1: Bước 3</p> <p>3b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500</p>
Quy trình thay thế	

3.13. Hiện thị ràng buộc thiết bị với thiết bị

Tên use case	Hiện thị ràng buộc thiết bị với thiết bị
Mô tả	Admin có thể xem thiết bị phụ thuộc vào thiết bị nào?
Điều kiện tiên quyết	Admin đăng nhập và vào trang “quản lí người dùng
Quy trình thông thường	<ol style="list-style-type: none">1. Admin bấm chọn vào thiết bị muốn xem ràng buộc ở mục “thiết bị” của mỗi người dùng2. Tự động gửi yêu cầu lấy danh sách các thiết bị ràng buộc3. Hiện thị modal và cập nhật thiết bị phụ thuộc trên modal này
Ngoại lệ	Ngoại lệ 1: Bước 2 2b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500
Quy trình thay thế	

3.14. Ràng buộc thiết bị với thiết bị

Tên use case	Ràng buộc thiết bị với thiết bị
Mô tả	Admin có thể ràng buộc thiết bị với thiết bị cụ thể là cảm biến với motor với mục đích tự động tưới
Điều kiện tiên quyết	Admin đăng nhập, vào trang “quản lí người dùng và chọn thiết bị cần ràng buộc
Quy trình thông thường	<ol style="list-style-type: none">1. Admin bấm chọn “+” trong trường thiết bị phụ thuộc2. Tự động gửi yêu cầu lấy danh sách các thiết bị thuộc người dùng mà chưa ràng buộc3. Hiện thị các thiết bị có thể ràng buộc4. Admin chọn “Ràng buộc”5. Gửi yêu cầu đến Server6. Server tạo truy vấn tạo ràng buộc giữa hai thiết bị7. Nếu thành công phản hồi và cập nhật thiết bị ràng buộc.
Ngoại lệ	<p>Ngoại lệ 1: Bước 2</p> <p>2b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500</p> <p>Ngoại lệ 1: Bước 7</p> <p>7b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500</p>
Quy trình thay thế	

3.15. Xóa Ràng buộc thiết bị với thiết bị

Tên use case	Xóa ràng buộc thiết bị với thiết bị
Mô tả	Admin có thể ràng buộc thiết bị với thiết bị
Điều kiện tiên quyết	Admin đăng nhập, vào trang “quản lí người dùng và chọn thiết bị cần xóa ràng buộc
Quy trình thông thường	<ol style="list-style-type: none"> 1. Admin bấm chọn “x” trong trường thiết bị phụ thuộc để xóa các thiết bị phụ thuộc 2. Yêu cầu xác nhận xóa từ Admin 3. Nếu đồng ý, Server tạo truy vấn xóa ràng buộc 4. Phản hồi kết quả xóa 5. Cập nhật giao diện
Ngoại lệ	<p>Ngoại lệ 1: Bước 2</p> <p>2b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500</p>
Quy trình thay thế	

3.16. Cập nhật vị trí thiết bị

Tên use case	Cập nhật vị trí thiết bị
Mô tả	Admin có thể cập nhật vị trí cho thiết bị
Điều kiện tiên quyết	Admin đăng nhập, vào trang “quản lí người dùng và chọn thiết bị cần cập nhật vị trí
Quy trình thông thường	<ol style="list-style-type: none"> 1. Admin bấm chọn “sửa” 2. Sau đó chọn vị trí trên bản đồ 3. Chọn Lưu 4. Gửi vị trí đến Server 5. Server tạo truy vấn cập nhật vị trí cho thiết bị 6. Phản hồi cho người dùng thiết bị cập nhật vị trí 7. Cập nhật giao diện người dùng
Ngoại lệ	<p>Ngoại lệ 1: Bước 5</p> <p>5b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500</p>
Quy trình thay thế	

3.17. Đăng xuất (người dùng).

Tên use case	Đăng xuất.
Mô tả	Người dùng đăng xuất khỏi ứng dụng khi có nhu cầu.
Điều kiện tiên quyết	Người dùng phải có tài khoản trước đó và đang ở trạng thái đăng nhập.
Hậu điều kiện	Người dùng đăng xuất thành công.
Quy trình thông thường	<ol style="list-style-type: none">1. Người dùng click chọn vào avatar ở bên góc phải trên cùng .2. Sau đó chọn vào nút đăng xuất.3. Server nhận yêu cầu đăng xuất từ người dùng và phản hồi.4. Tiến hành xóa phiên đăng nhập5. Đặt thiết đặt xóa cookie, chuyển tiếp đến trang đăng nhập người dùng
Ngoại lệ	
Quy trình thay thế	

3.18. Đăng nhập (admin).

Tên use case	Đăng nhập (admin).
Mô tả	Admin đăng nhập vào trang riêng dành cho admin để quản lý ứng dụng.
Điều kiện tiên quyết	Admin có tài khoản admin đồng thời truy cập đúng địa chỉ dành cho admin
Hậu điều kiện	Admin đăng nhập thành công.
Quy trình thông thường	<ol style="list-style-type: none">1. Admin vào trang dành cho admin quản lý.2. Điền email và mật khẩu chọn nút “đăng nhập”.3. Server nhận yêu cầu và xác thực.4. Nếu đúng đặt cookie vào header và chuyển tiếp đến trang chủ admin
Ngoại lệ	Ngoại lệ 1: Bước 4 3b. Xác thực không thành công trả về lỗi
Quy trình thay thế	

3.19. Xác thực (admin).

Tên use case	Xác thực (admin).
Mô tả	Xác thực mọi request đến từ client Admin thông qua cookie.
Điều kiện tiên quyết	Admin đã đăng nhập
Quy trình thông thường	<ol style="list-style-type: none">1. Request từ client Admin2. Server kiểm tra tính hợp lệ cookie3. Nếu hợp lệ tiếp tục thực hiện request, không hợp lệ từ chối request, trả về trang login cho admin
Ngoại lệ	Ngoại lệ 2: Bước 4 2b. Không thể kiểm tra tính hợp lệ từ chối request, trả về trang login cho admin
Quy trình thay thế	

3.20. Đăng xuất (admin).

Tên use case	Đăng xuất (admin).
Mô tả	Admin đăng xuất khỏi tài khoản admin tại trang web dành cho admin khi có nhu cầu.
Điều kiện tiên quyết	Admin phải đang ở trạng thái đăng nhập tại trang web dành cho admin
Hậu điều kiện	Admin đăng xuất thành công.
Quy trình thông thường	<ol style="list-style-type: none">1. Admin Logout bên góc phải trên cùng.2. Server đặt lệnh xóa cookie vào header và gửi đến người dùng và chuyển tiếp đến trang đăng nhập Admin
Ngoại lệ	
Quy trình thay thế	

3.21 Nhận và hiển thị dữ liệu từ thiết bị:

a. Theo dõi độ ẩm của từng cảm biến theo thời gian thực.

Tên use case	Theo dõi độ ẩm của từng cảm biến theo thời gian thực
Mô tả	Dựa vào dữ liệu về độ ẩm trong cơ sở dữ liệu, lập biểu đồ theo dõi độ ẩm đất cho từng cảm biến trong 3 giờ gần nhất. Cập nhật dữ liệu theo thời gian thực.
Điều kiện tiên quyết	Người dùng đã đăng nhập thành công
Quy trình thông thường	5. Gửi yêu cầu truy vấn dữ liệu 6. Nhận dữ liệu, cập nhật và hiển thị lại đồ thị.
Ngoại lệ	
Quy trình thay thế	

b. Xem độ ẩm trung bình của từng cảm biến trong tuần này và tuần trước.

Tên use case	Theo dõi độ ẩm của từng cảm biến theo thời gian thực
Mô tả	Dựa vào dữ liệu về độ ẩm trong cơ sở dữ liệu, lập biểu đồ cột độ ẩm trung bình trong tuần này và tuần trước.
Điều kiện tiên quyết	Người dùng đã đăng nhập thành công
Quy trình thông thường	1. Gửi yêu cầu truy vấn dữ liệu 2. Nhận dữ liệu, tính toán dựa trên dữ liệu và hiển thị đồ thị.
Ngoại lệ	
Quy trình thay thế	

c. Xem số lần tưới trong ngày tương ứng với tuần này và tuần trước.

Tên use case	Xem số lần tưới trong ngày tương ứng với tuần này và tuần trước
Mô tả	Thống kê số lần tưới nước các ngày trong tuần của tuần này và tuần trước và biểu diễn dưới dạng biểu đồ cột.
Điều kiện tiên quyết	Người dùng đã đăng nhập thành công
Quy trình thông thường	<ol style="list-style-type: none">1. Gửi yêu cầu truy vấn dữ liệu2. Nhận dữ liệu, tính toán dựa trên dữ liệu và hiển thị đồ thị.
Ngoại lệ	
Quy trình thay thế	

3.22 Quản lý thông tin cá nhân:

a. Hiển thị thông tin người dùng.

Tên use case	Hiển thị thông tin người dùng
Mô tả	Hiển thị thông tin người dùng gồm: Họ và tên, email.
Điều kiện tiên quyết	Người dùng đã đăng nhập thành công
Quy trình thông thường	<ol style="list-style-type: none">1. Gửi yêu cầu truy vấn dữ liệu2. Nhận dữ liệu và hiển thị trên trang cá nhân.
Ngoại lệ	
Quy trình thay thế	

b. Hiển thị phiên đăng nhập.

Tên use case	Hiển thị phiên đăng nhập
Mô tả	Hiển thị thông tin lịch sử các phiên đăng nhập gồm: ID phiên đăng nhập, IP, thông tin về trình duyệt sử dụng.
Điều kiện tiên quyết	Người dùng đã đăng nhập thành công
Quy trình thông thường	<ol style="list-style-type: none">1. Gửi yêu cầu truy vấn dữ liệu2. Nhận dữ liệu và hiển thị trên trang cá nhân.
Ngoại lệ	
Quy trình thay thế	

c. Xóa phiên đăng nhập.

Tên use case	Xóa phiên đăng nhập
Mô tả	Người dùng có thể xóa các phiên đăng nhập trước. Tuy nhiên không thể xóa phiên đăng nhập hiện tại.
Điều kiện tiên quyết	Người dùng đã đăng nhập thành công
Quy trình thông thường	<ol style="list-style-type: none">1. Nhấn button xóa ứng với phiên đăng nhập mà người dùng muốn xóa.2. Xóa dữ liệu phiên đăng nhập và cập nhật lại danh sách phiên đăng nhập.
Ngoại lệ	
Quy trình thay thế	

d. Hiển thị thiết bị và quan hệ phụ thuộc các thiết bị.

Tên use case	Hiển thị thiết bị và quan hệ phụ thuộc các thiết bị
Mô tả	Hiển thị danh sách các thiết bị và ràng buộc của nó. Các nội dung hiển thị bao gồm: Tên thiết bị, kinh độ, vĩ độ, loại cảm biến, các thiết bị phụ thuộc, ngưỡng cho phép.
Điều kiện tiên quyết	Người dùng đã đăng nhập thành công
Quy trình thông thường	<ol style="list-style-type: none">1. Gửi yêu cầu truy vấn dữ liệu2. Nhận dữ liệu và hiển thị trên trang cá nhân.
Ngoại lệ	
Quy trình thay thế	

e. Đặt ngưỡng độ ẩm cho phép cho cảm biến.

Tên use case	Đặt ngưỡng độ ẩm cho phép cho cảm biến
Mô tả	Người dùng có thể điều chỉnh ngưỡng của cảm biến độ ẩm.
Điều kiện tiên quyết	Người dùng đã đăng nhập thành công
Quy trình thông thường	<ol style="list-style-type: none">0. Người dùng bấm button Xóa để xóa cài đặt ngưỡng cho thiết bị.1. Người dùng bấm button Sửa để điều chỉnh ngưỡng của thiết bị2. Người dùng nhấn button Cập nhật để cập nhật lại ngưỡng của thiết bị.
Ngoại lệ	
Quy trình thay thế	

3.23 Hiện thị thông tin tổng quan:

a. Thống kê số lượng thiết bị:

Tên use case	Thống kê số lượng thiết bị
Mô tả	Thống kê số lượng thiết bị.
Điều kiện tiên quyết	Admin đã đăng nhập thành công
Quy trình thông thường	<ol style="list-style-type: none">1. Gửi yêu cầu truy vấn dữ liệu2. Nhận dữ liệu và hiển thị trên trang dashboard.
Ngoại lệ	
Quy trình thay thế	

b. Thống kê số lượng người dùng:

Tên use case	Thống kê số lượng người dùng
Mô tả	Thống kê số lượng người dùng.
Điều kiện tiên quyết	Admin đã đăng nhập thành công
Quy trình thông thường	<ol style="list-style-type: none">1. Gửi yêu cầu truy vấn dữ liệu2. Nhận dữ liệu và hiển thị trên trang dashboard.
Ngoại lệ	
Quy trình thay thế	

c. Hiện thị danh sách người dùng:

Tên use case	Hiện thị danh sách người dùng
Mô tả	Hiện thị bảng danh sách thông tin người dùng gồm: họ và tên, email.
Điều kiện tiên quyết	Admin đã đăng nhập thành công
Quy trình thông thường	<ol style="list-style-type: none">1. Gửi yêu cầu truy vấn dữ liệu2. Nhận dữ liệu và hiển thị trên trang dashboard.
Ngoại lệ	
Quy trình thay thế	

d. Hiện thị danh sách thiết bị:

Tên use case	Hiển thị danh sách thiết bị
Mô tả	Hiển thị bảng danh sách thông tin các thiết bị của hệ thống gồm: tên thiết bị, loại thiết bị.
Điều kiện tiên quyết	Admin đã đăng nhập thành công
Quy trình thông thường	<ol style="list-style-type: none"> 1. Gửi yêu cầu truy vấn dữ liệu 2. Nhận dữ liệu và hiển thị trên trang dashboard.
Ngoại lệ	
Quy trình thay thế	

e. Tìm kiếm người dùng:

Tên use case	Tìm kiếm người dùng
Mô tả	Admin có thể tìm kiếm người dùng trong bảng danh sách người dùng bằng cách tìm theo tên hoặc tìm theo email. Kết quả tìm kiếm được cập nhật động theo từ khóa tìm kiếm.
Điều kiện tiên quyết	Admin đã đăng nhập thành công
Quy trình thông thường	<ol style="list-style-type: none"> 0. Bấm vào dropdown để chọn cách tìm kiếm (user hoặc email) 1. Nhập từ khóa tìm kiếm 2. Hiển thị các kết quả tìm kiếm liên quan
Ngoại lệ	
Quy trình thay thế	

f. Tìm kiếm thiết bị:

Tên use case	Tìm kiếm thiết bị
Mô tả	Admin có thể tìm kiếm thiết bị trong bảng danh sách các thiết bị bằng cách tìm theo tên hoặc tìm theo phân loại thiết bị. Kết quả tìm kiếm được cập nhật động theo từ khóa tìm kiếm.
Điều kiện tiên quyết	Admin đã đăng nhập thành công
Quy trình thông thường	<ol style="list-style-type: none"> 0. Bấm vào dropdown để chọn cách tìm kiếm (loại hoặc id) 1. Nhập từ khóa tìm kiếm

	2. Hiển thị các kết quả tìm kiếm liên quan
Ngoại lệ	
Quy trình thay thế	

3.21. Quản lí thiết bị

Tên use case	Quản lý thiết bị.
Mô tả	Cho phép quản lý xem các thông tin về thiết bị của người dùng. Ngoài ra, quản lý còn có thể thêm và xoá các thiết bị của người dùng.
Điều kiện tiên quyết	Quản lý đã đăng nhập thành công vào hệ thống và đang ở trang “Quản lý thiết bị”.
Quy trình thông thường	<ol style="list-style-type: none"> 1. Quản lý 2. Điền email và mật khẩu đã đăng ký của mình vào. 3. Chọn nút đăng nhập. 4. Server nhận yêu cầu và xác thực. 5. Người dùng nhận phản hồi đăng nhập thành công.
Ngoại lệ	<p>Ngoại lệ 1: Bước 4</p> <p>4b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500.</p> <p>Ngoại lệ 2: Bước 5.</p> <p>5b. Không tồn tại tài khoản hoặc đăng nhập sai.</p>
Quy trình thay thế	<p>4b. Khi bị mất kết nối người dùng cần reload lại trang.</p> <p>5b. Nhập lại thông tin đăng nhập.</p>

3.22. Đăng ký

Tên use case	Đăng ký.
Mô tả	Người dùng sẽ đăng ký tài khoản để sử dụng ứng dụng.
Điều kiện tiên quyết	Người dùng phải có tài khoản email và email đó chưa được đăng ký.
Hậu điều kiện	Người dùng đăng ký thành công.
Quy trình thông thường	<ol style="list-style-type: none">1. Người dùng truy cập vào trang đăng nhập.2. Bấm vào “Sign up for an account”.3. Nhập email đăng ký.4. Nhập tên đầy đủ cũng như là mật khẩu.5. Server nhận yêu cầu và kiểm tra phản hồi.6. Kiểm tra email để xác thực đăng ký tài khoản.
Ngoại lệ	Ngoại lệ 1: Bước 4 4b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500.
Quy trình thay thế	3b. Khi mất kết nối, người dùng cần reload lại trang.

3.23. Đăng nhập (Người dùng)

Tên use case	Đăng nhập (Người dùng).
Mô tả	Người dùng dùng tài khoản đã đăng ký để đăng nhập và sử dụng ứng dụng.
Điều kiện tiên quyết	Người dùng phải có tài khoản email đã đăng ký.
Hậu điều kiện	Người dùng đăng nhập thành công.
Quy trình thông thường	<ol style="list-style-type: none">1. Người dùng truy cập vào trang đăng nhập2. Điền email và mật khẩu đã đăng ký của mình vào.3. Chọn nút đăng nhập.4. Server nhận yêu cầu và xác thực, lưu thông tin người dùng, đặt cookie vào header5. Người dùng nhận phản hồi đăng nhập thành công.
Ngoại lệ	Ngoại lệ 1: Bước 4 4b. Không thể truy vấn do kết nối với database bị lỗi trả về lỗi HTTP 500. Ngoại lệ 2: Bước 5. 5b. Không tồn tại tài khoản hoặc đăng nhập sai.
Quy trình thay thế	

4. Yêu cầu phi chức năng

Mô tả	Độ đo
Thời gian phản hồi của máy bơm nhanh.	Độ trễ từ lúc bật/ tắt cho đến lúc máy bơm thực hiện dưới 1 phút.

Giá trị độ ẩm dự đoán gần chính xác	Sai số không quá 10 (đơn vị độ ẩm)
Phân tích, hiển thị thông số thiết bị nhanh	Không quá 3s
Cập nhật điều chỉnh khoảng giá trị cho phép của các cảm biến.	Không quá 5s
Thời gian cập nhật tương tác giữa dữ liệu của sensor và trạng thái mạch motor.	Không quá 5s.

VI. Tổng kết:

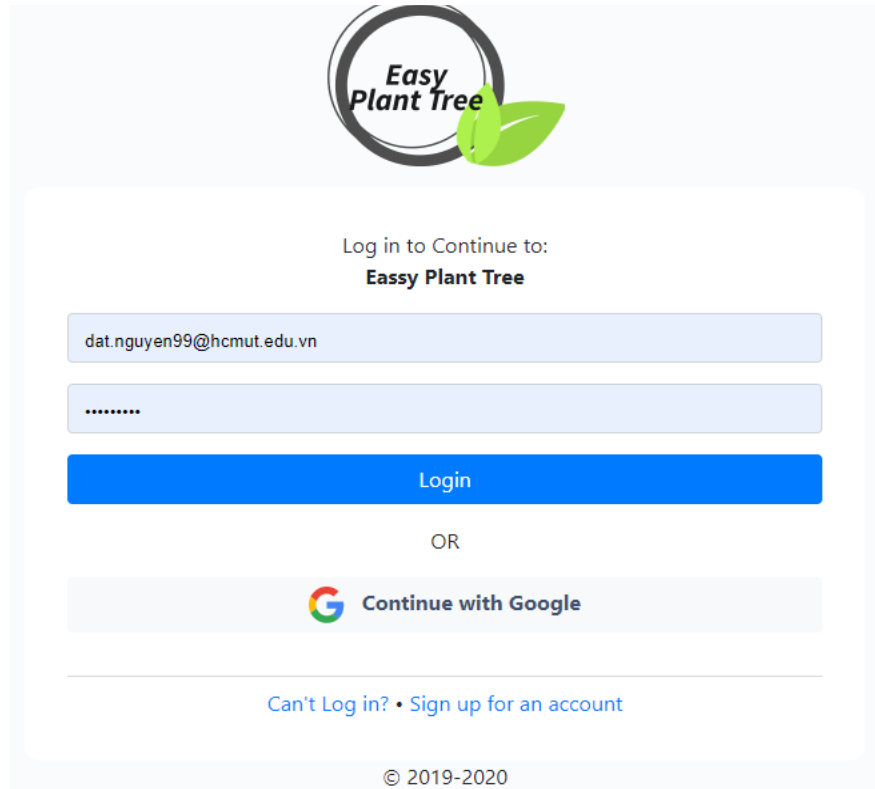
1. Kết quả hiện thực:

Nhóm đã xây dựng hết các chức năng được đặc tả như ở các phần bên trên và deploy lên heroku: <https://easyplanttree.herokuapp.com/>

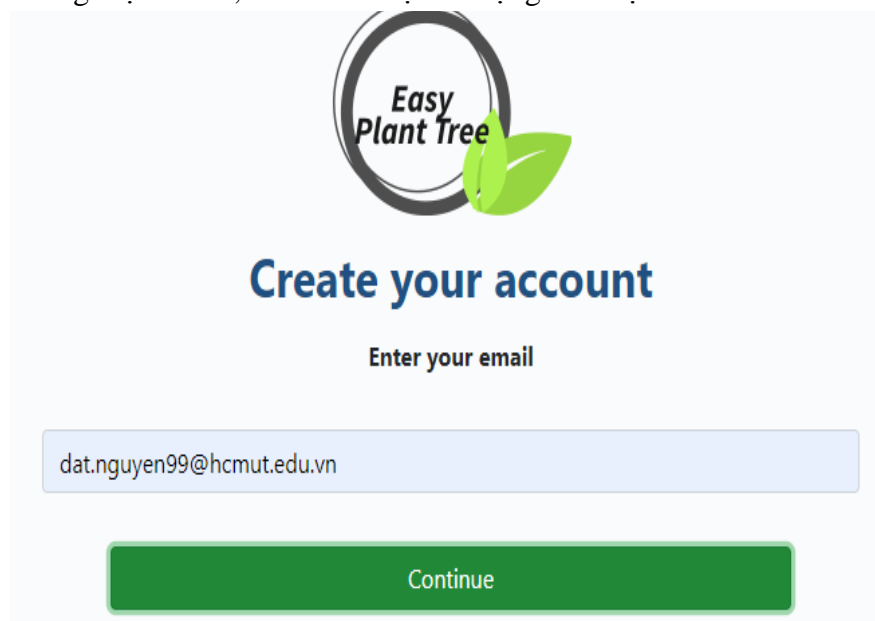
Sau đây là một số hình ảnh demo ứng dụng.

a) Nhóm chức năng đăng nhập, đăng ký, đăng xuất cho người dùng:

- Chức năng đăng nhập



- Chức năng đăng ký: Chức năng đăng ký có sử dụng Certificate authority để chứng thực email, và có thời hạn sử dụng nhất định.





Create your account

Enter your email

dat.nguyen99@hcmut.edu.vn

Nguyễn Lê Tiến Đạt

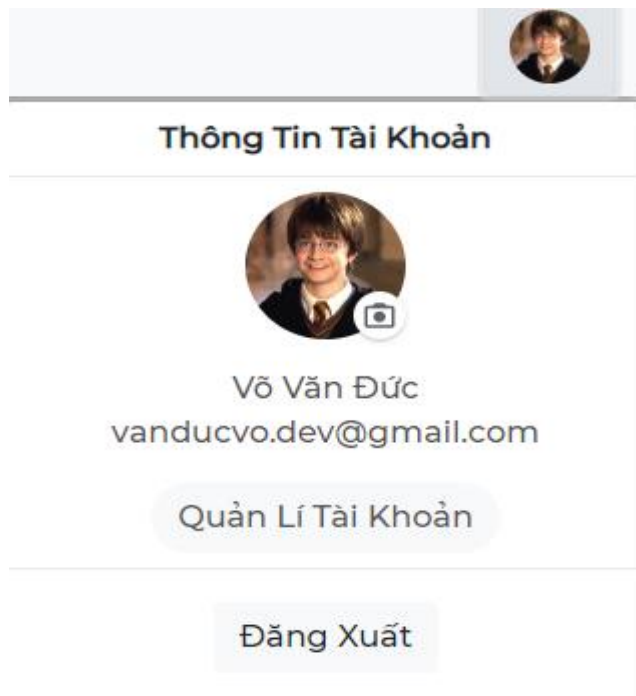
.....|

Create Account



Check Mail to Verify Account!

- Sau khi đăng nhập thành công, người dùng có thể đăng xuất.



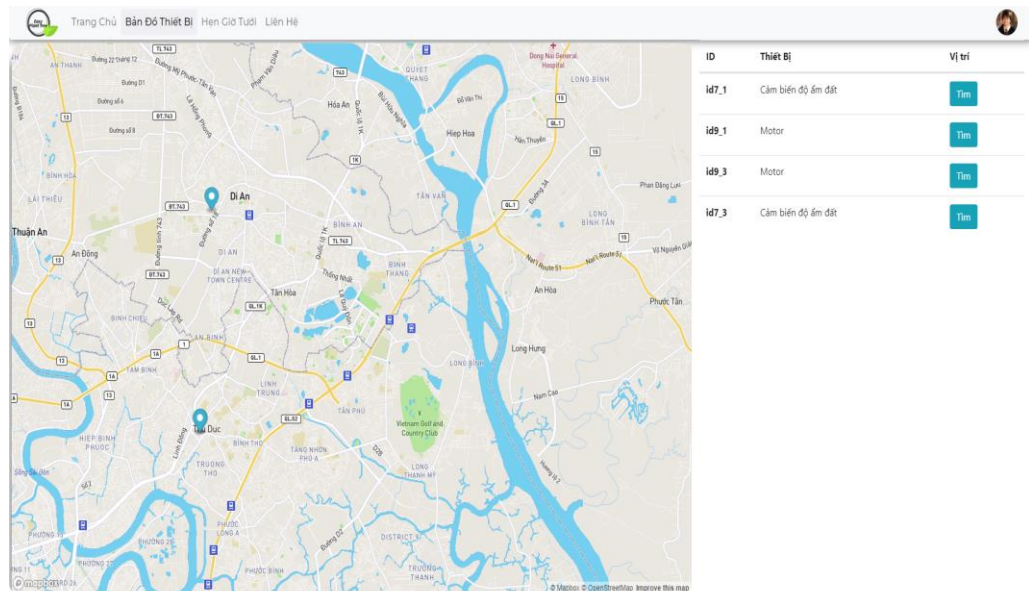
b) Nhóm chức năng về hiển thị :

- Hiển thị độ ẩm theo thời gian thực trong 3 giờ gần nhất của cảm biến độ ẩm đất cũng như độ ẩm trung bình của cảm biến theo tuần với hai tùy chọn là tuần này và tuần trước, lịch sử tưới các ngày trong tuần với hai tùy chọn là tuần này và tuần trước.

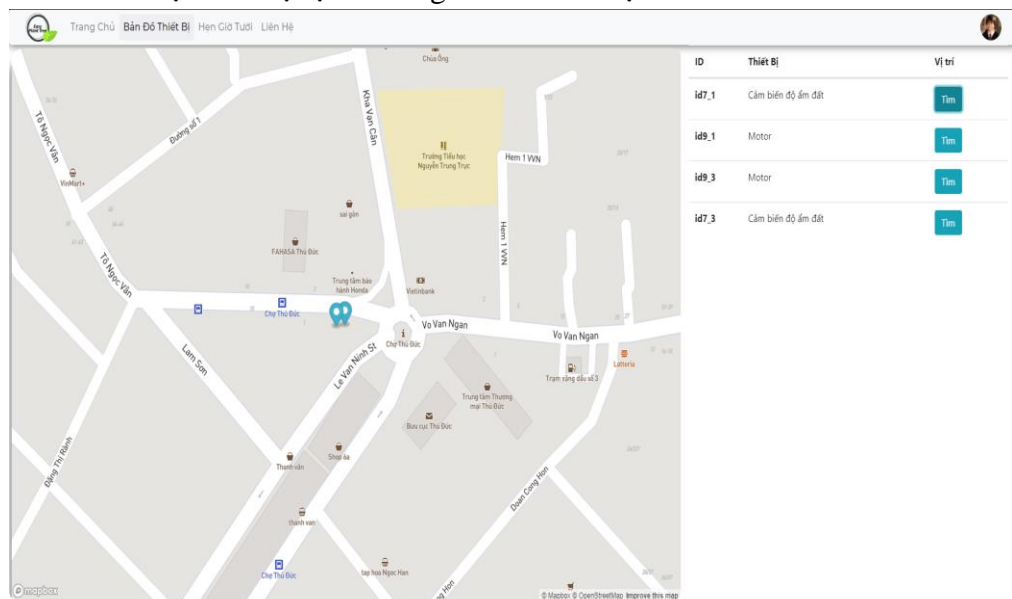


c) Nhóm chức năng về định vị thiết bị trên bản đồ:

- Hiển thị vị trí các thiết bị trên bản đồ:

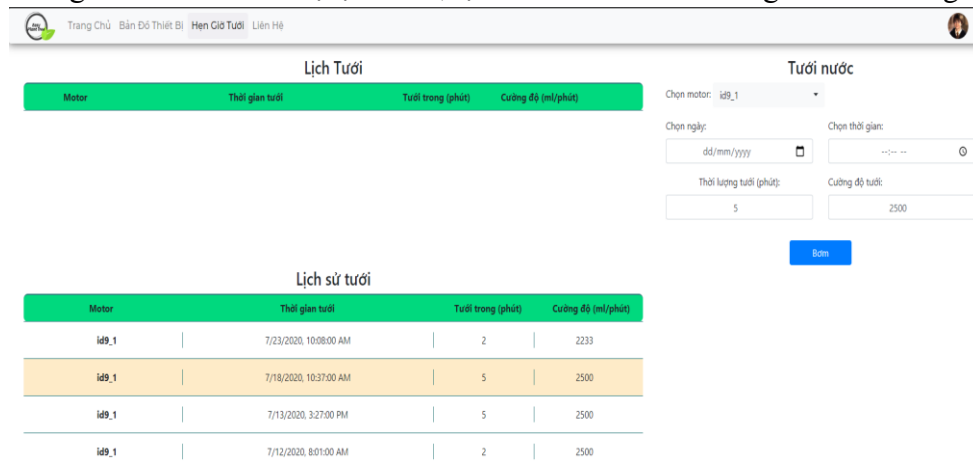


- Tìm kiếm một thiết bị cụ thể bằng cách nhấn chọn button “Tìm”:



- d) Nhóm chức năng về tưới nước:

- Trang tưới nước hiển thị lịch tưới, lịch sử tưới và chức năng tưới thủ công.



- Để tưới thủ công, người dùng chọn ngày giờ tưới, cường độ tưới, thời gian tưới và nhấn chọn button “Bơm” để bắt đầu tưới nước. Hệ thống sẽ yêu cầu người dùng xác nhận đặt lịch tưới này.

The screenshot shows the 'Lịch Tưới' (Irrigation Schedule) and 'Tưới nước' (Watering) sections. A modal dialog titled 'Xác Nhận Đặt Lịch' (Confirm Scheduling) is displayed in the center. The dialog contains the following information:

- Thiết bị: id9_1
- Vào Lúc: 20:00:00, 30/7/2020
- Tưới trong: 3 phút
- Cường độ: 2500

At the bottom of the dialog are two buttons: 'Đóng' (Close) and 'Đồng ý' (Agree). In the background, the 'Tưới nước' section shows a form with 'Chọn motor: id9_1', 'Chọn ngày: 30/07/2020', 'Chọn thời gian: 08:00 CH', 'Thời lượng tưới (phút): 3', and 'Cường độ tưới: 2500'. A blue 'Bơm' (Pump) button is visible.

- Chọn “Đóng” để hủy xác nhận. Chọn “Đồng ý” để xác nhận lịch tưới, lúc này, lịch tưới sẽ được cập nhật lại vào bảng lịch tưới. Người dùng có thể xóa lịch tưới đó nếu không muốn tưới nữa.

The screenshot shows the 'Lịch Tưới' (Irrigation Schedule) and 'Tưới nước' (Watering) sections. The 'Lịch tưới' table has been updated with a new entry. The 'Tưới nước' section remains the same as in the previous screenshot.

Motor	Thời gian tưới	Tưới trong (phút)	Cường độ (ml/phút)
id9_1	20:00:00, 30/7/2020	3	2500

Motor	Thời gian tưới	Tưới trong (phút)	Cường độ (ml/phút)
id9_1	7/23/2020, 10:08:00 AM	2	2233
id9_1	7/18/2020, 10:37:00 AM	5	2500
id9_1	7/13/2020, 3:27:00 PM	5	2500
id9_1	7/12/2020, 8:01:00 AM	2	2500

e) Nhóm chức năng về quản lý tài khoản:

- Người dùng chọn mục “Quản lý tài khoản” ở phía trên mục “Đăng xuất”. Nhóm chức năng này bao gồm: hiển thị thông tin cá nhân, lịch sử các phiên đăng nhập, thông tin về các thiết bị và ràng buộc.

[Trang Chủ](#)
[Bản Đồ Thiết Bị](#)
[Hẹn Giờ Tưới](#)
[Liên Hệ](#)

Tên

Vũ Văn Đức

Email

vanducvo.dev@gmail.com

Phiên Đăng Nhập

5f1a861cd974e80017b71669

IP: 42.119.158.49

Browser: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/83.0.4103.61 Chrome/83.0.4103.61 Safari/537.36

Xóa

5f216ea1f64a2a0017e11993

IP: 1.59.206.225

Browser: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.89 Safari/537.36

Xóa

5f216f6cf64a2a0017e1199a

IP: 14.241.237.196

Browser: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36

Xóa

Thông tin thiết bị và ràng buộc

ID: id7_1

Lat: 10.85116618942132

Long: 106.75467036647501

Loại: Cảm biến độ ẩm đất

Phụ thuộc:

ID: id9_1

Lat: 10.8511660850442383

Long: 106.75462699649461

Loại: Motor

Ngưỡng cho phép:

Nhỏ nhất

Lớn nhất

Sửa

Xóa

ID: id7_3

Lat: 10.808559450555828

Long: 106.75826462968064

Loại: Cảm biến độ ẩm đất

Phụ thuộc:

ID: id9_3

Loại: Motor

Ngưỡng cho phép:

Nhỏ nhất

Lớn nhất

- Người dùng có thể xóa các phiên đăng nhập trước đây, có thể thiết lập các ngưỡng để phục vụ cho việc tưới tự động.

Ngưỡng cho phép:

Nhỏ nhất

Lớn nhất

12

123

Sửa

Xóa

f) Nhóm chức năng về đăng nhập, đăng xuất cho admin:

- Tương tự như chức năng đăng nhập, đăng xuất của người dùng.

ADMIN PAGE - EASY PLANT TREE

Log in to Continue to:
Eassy Plant Tree - ADMIN PAGE

vanducvo.dev@gmail.com

.....

Login

[Can't Log in?](#) • [Contact To Your Admin](#)

g) Nhóm chức năng về hiển thị các thông tin cơ bản cho admin:

- Trang chủ của admin hiển thị thống kê về số lượng người dùng và thiết bị, bảng danh sách người dùng, thiết bị.

76

Trang Chủ | Quản Lý Người Dùng | Quản Lý Thiết Bị | Logout Admin: Võ Văn Đức

Số người dùng 1 | Số thiết bị 7

Danh sách người dùng

Email:

Khách Hàng	Email
Võ Văn Đức	vanducvo.dev@gmail.com

Danh sách thiết bị

Loại:

ID	Loại
id7_1	Cảm biến độ ẩm đất
id9_1	Motor
id9_2	Motor
id7_2	Cảm biến độ ẩm đất
id9_3	Motor
id9_4	Motor
id7_3	Cảm biến độ ẩm đất

- Admin có thể tìm kiếm người dùng, thiết bị bằng cách gõ từ khóa tìm kiếm phù hợp với chế độ tìm kiếm, ví dụ như nhập email người dùng với chế độ tìm kiếm theo email,...

Trang Chủ | Quản Lý Người Dùng | Quản Lý Thiết Bị | Logout Admin: Võ Văn Đức

Số người dùng 1 | Số thiết bị 7

Danh sách người dùng

Email:

Khách Hàng	Email
------------	-------

Danh sách thiết bị

ID:

ID	Loại
id7_1	Cảm biến độ ẩm đất
id7_2	Cảm biến độ ẩm đất
id7_3	Cảm biến độ ẩm đất

h) Nhóm chức năng về quản lý người dùng cho admin:

- Admin có thể tìm kiếm người dùng bằng cách chọn chiến lược tìm kiếm mà mình muốn (tìm theo email hoặc tên người dùng).

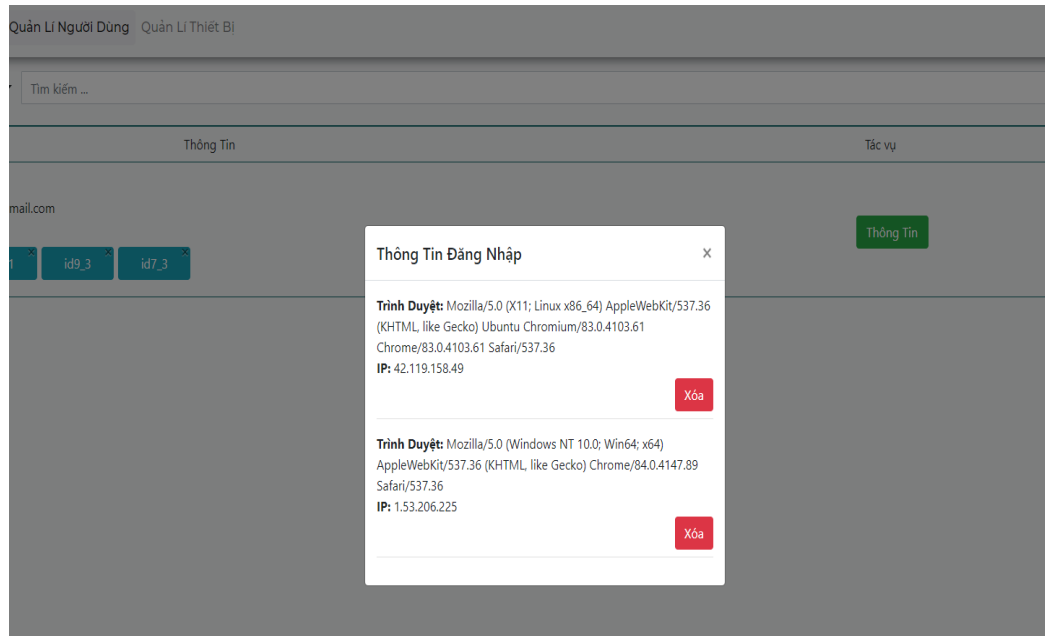
Email:

Thông Tin

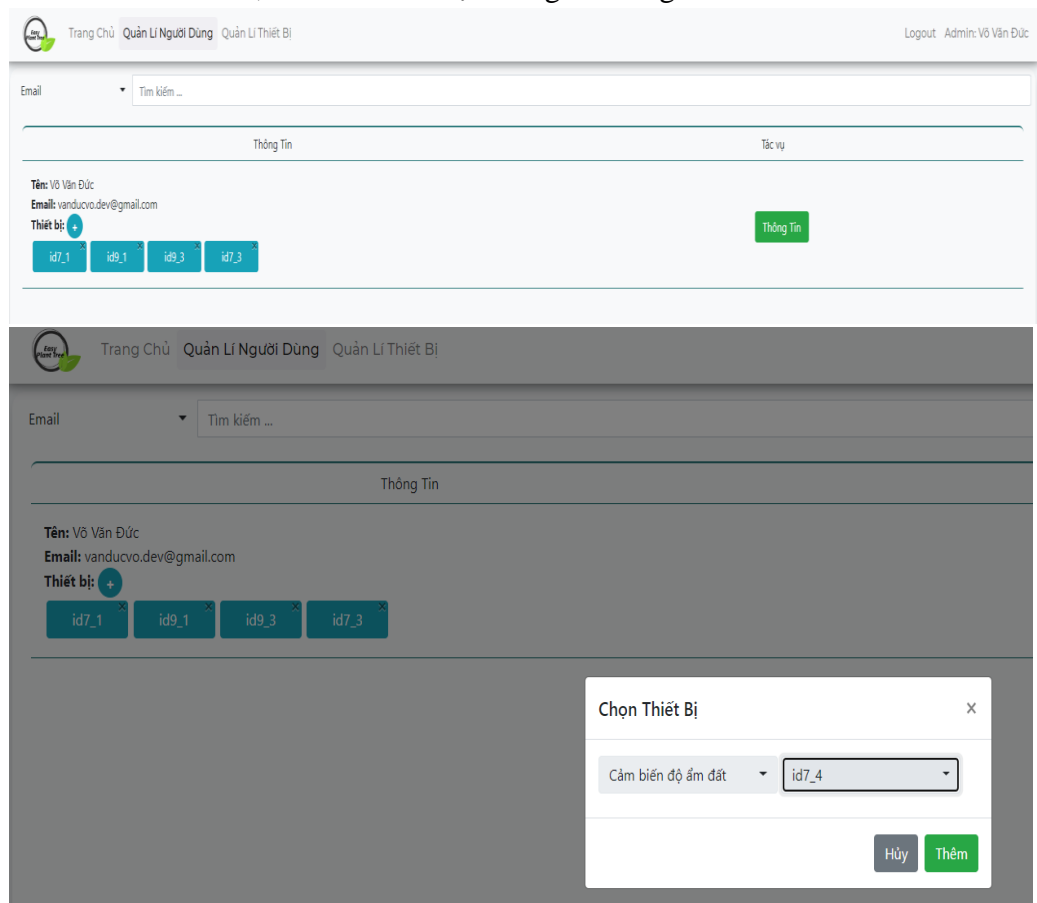
Tên: Võ Văn Đức
Email: vanducvo.dev@gmail.com
Thiết bị: +

id7_1 x id9_1 x id9_3 x id7_3 x

- Admin có thể xem “Thông tin” về các phiên đăng nhập của người dùng.

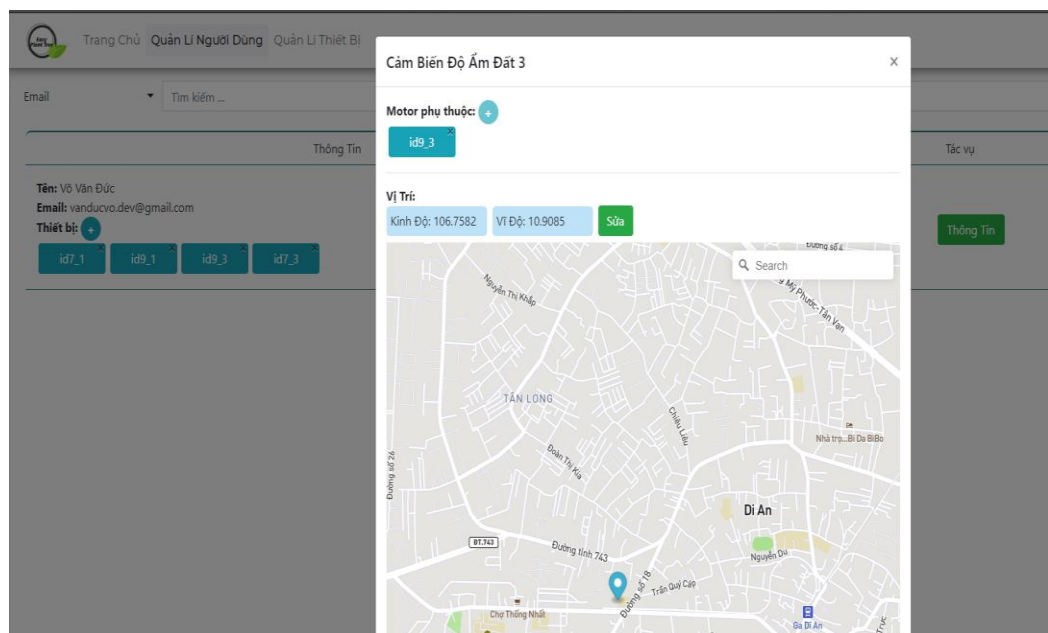


- Admin có thể thêm, xóa các thiết bị cho người dùng.





- Để cài đặt vị trí cũng như ràng buộc cho thiết bị, người dùng nhấn vào thiết bị đó và tiến hành thêm các ràng buộc cũng như thiết lập vị trí cho thiết bị đó.



Cảm Biến Độ Ẩm Đất 3



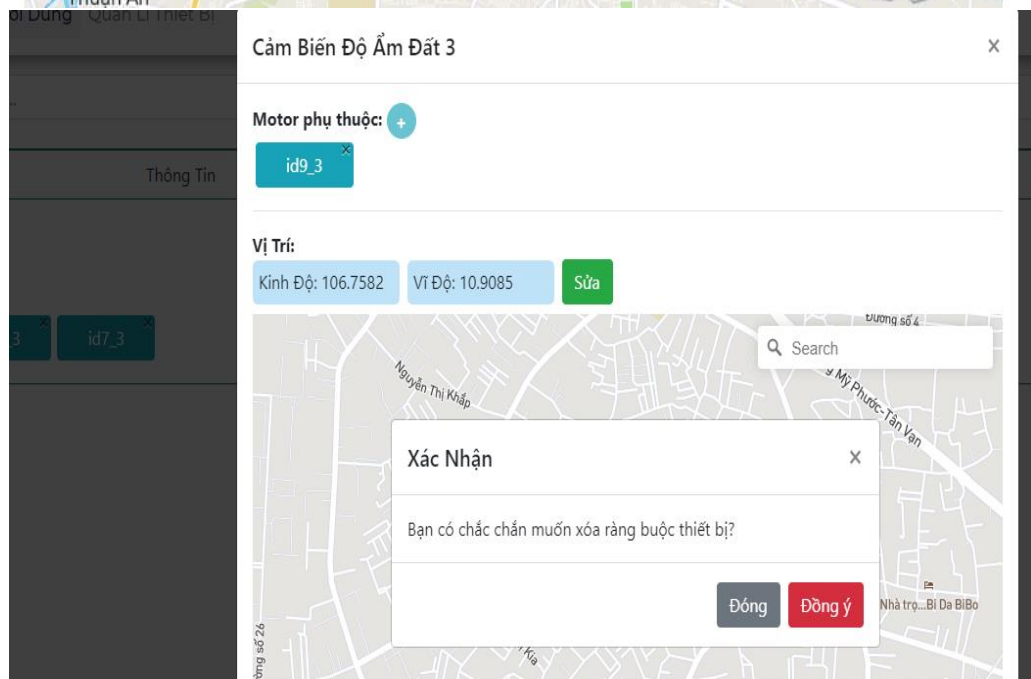
Motor phụ thuộc: +

Vị Trí:

Kinh Độ: 106.7582

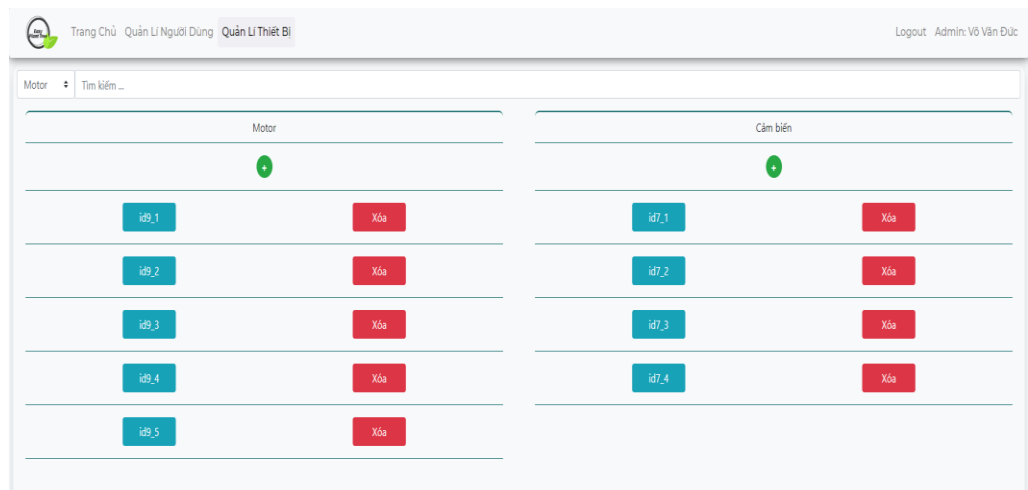
Vĩ Độ: 10.9085

Sửa

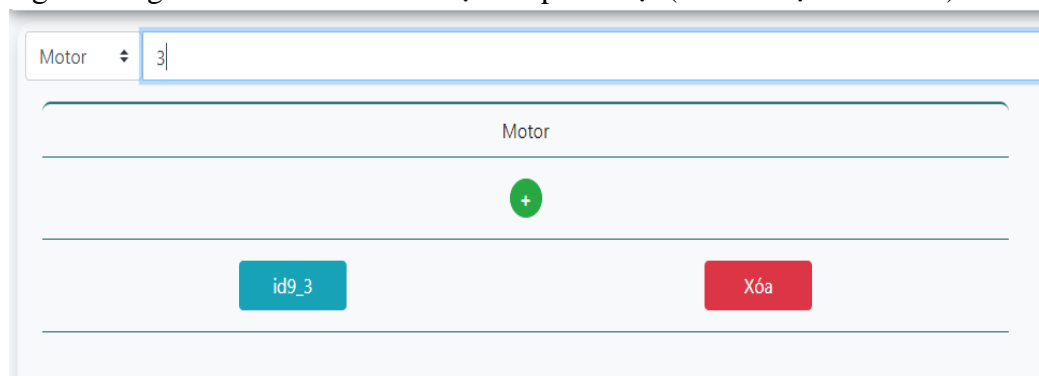


i) Nhóm chức năng về quản lí thiết bị cho admin:

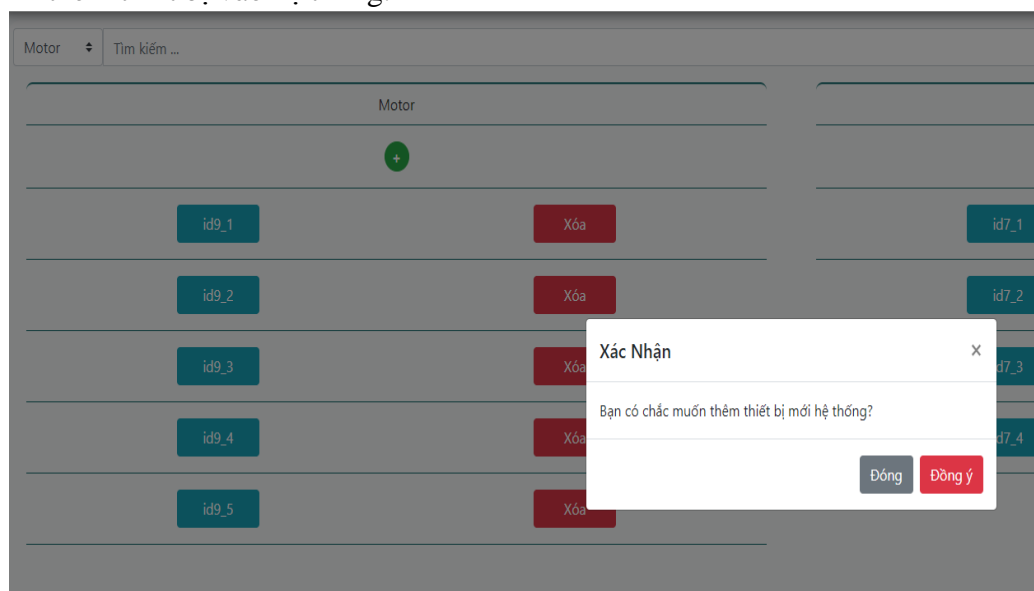
- Trang quản lí thiết bị hiển thị danh sách 2 bảng danh sách motor và cảm biến.



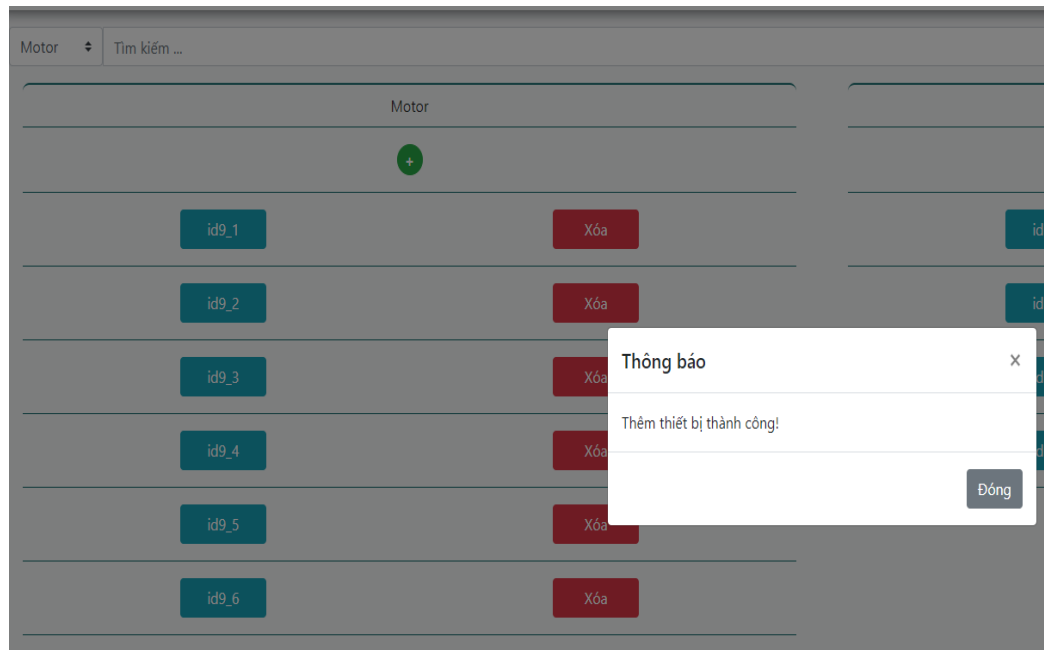
- Người dùng có thể tìm kiếm thiết bị theo phân loại (motor hoặc cảm biến).



- Người dùng có thể thêm thiết bị bằng cách chọn button dấu cộng màu xanh để thêm thiết bị vào hệ thống.



- Sau khi thêm thiết bị thành công, hệ thống hiển thị thông báo thêm thiết bị thành công và cập nhật lại danh sách loại thiết bị đó.



2. Phương hướng phát triển trong tương lai:

- Nâng cấp, cải thiện tính thẩm mỹ cho giao diện người dùng.
- Refactor lại mã nguồn ứng dụng.
- Ứng dụng machine learning vào việc phân tích, dự đoán độ ẩm đất phục vụ cho việc tưới tự động.
- Thử nghiệm trên các hệ thống IoT thật.
- Mở rộng thêm nhiều thiết bị hơn như: cảm biến ánh sáng, nhiệt độ,...

3. Kết luận:

Qua một quá trình thực hiện đồ án, nhóm chúng em đã hiện thực được sản phẩm sau cùng với đầy đủ các mục tiêu, yêu cầu cơ bản mà môn học đề ra. Do thời gian thực hiện đồ án có giới hạn nên sản phẩm của chúng em có thể chưa làm hài lòng hết toàn bộ.

Chúng em đã vận dụng được một số kiến thức về khoa học máy tính như: lập trình web, mạng máy tính, hệ thống IoT,... và có một cái nhìn mới về đồ án đa ngành có sự liên kết giữa CS và CE. Chúng em hy vọng trong tương lai, chúng em có thể tạo ra các sản phẩm khác có tính ứng dụng cao, mang lại lợi ích cho khách hàng.

Sau cùng, nhóm xin gửi lời cảm ơn đến thầy Trần Khải Thiện đã giúp đỡ nhóm trong quá trình phát triển đồ án.