

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
Faculty of Computer Science and Engineering



THỰC TẬP ĐỒ ÁN ĐA NGÀNH

Báo cáo đề tài

Khu vườn thông minh

Teacher:	Bùi Xuân Giang	
Student:	Nguyễn Hữu Hưng	1812496
	Nguyễn Việt Hưng	1812499
	Đoàn Nguyên Khang	1812533
	Trần Quang Huy	1812427
	Võ Phạm Long Huy	1812427

Contents

1	Giới thiệu đề tài, phân tích chức năng	3
1.1	Đề tài	3
1.2	Yêu cầu	3
1.3	Phân tích chức năng	3
2	Phân công công việc	5
2.1	Đánh giá chung	5
2.2	Chi tiết giai đoạn	5
3	Tìm hiểu công nghệ	6
3.1	Giao thức MQTT	6
3.1.1	MQTT là gì ?	6
3.1.2	Kiến trúc mức cao của MQTT	6
3.1.3	Topic	7
3.1.4	MQTT Bridge:	7
3.2	IoT gateway	8
3.2.1	IoT gateway là gì?	8
3.2.2	Các chức năng của Iot gateway	9
3.3	ReactJs	10
3.3.1	Giới thiệu	10
3.3.2	React là gì?	10
3.3.3	Tại sao sử dụng React?	11
3.4	Python	12
3.4.1	Python là gì?	12
3.4.2	Ưu điểm	12
3.4.3	Hạn chế	12
4	Thiết kế hệ thống	13
4.1	Use case	13
4.2	Đặc tả use case	14
4.2.1	Login	14
4.2.2	Logout	14
4.2.3	Create account	14
4.2.4	View temperature, soil humidity, air humidity in garden	15
4.2.5	Turn on/off water pump motor	15
4.2.6	Search/sort garden of user	16
4.2.7	Change account information.	16
4.2.8	Checking information of garden	17
4.2.9	View history turn on/off motor	17
4.2.10	View account information	18
4.2.11	Delete account	18
4.2.12	Quy định mức giới hạn của độ ẩm, nhiệt độ	19
4.2.13	Sửa đổi mức giới hạn của độ ẩm, nhiệt độ	19
4.2.14	Pump water	19
4.2.15	Measure soil humidity	19
4.2.16	Measure temperature	20
4.2.17	Measure real time	20
4.2.18	Send data	20
4.3	Giao diện dự kiến	21
4.3.1	Giao diện chung	21
4.3.2	Giao diện admin	22
4.3.3	Giao diện user	23



4.4	Cơ sở dữ liệu	27
4.4.1	Mô hình dữ liệu	27
4.4.2	Mô tả dữ liệu	28
4.5	Giao tiếp server	29
5	Mô tả chi tiết	31
5.1	Dịch vụ đăng ký xác thực	31
5.2	Dịch vụ quản lý	32
5.3	Dịch vụ người dùng	33
6	Kết luận và hướng phát triển	35
6.1	Kết luận	35
6.2	Hướng phát triển	35
7	Tài liệu tham khảo	36
	Tài liệu	36

1 Giới thiệu đề tài, phân tích chức năng

1.1 Đề tài

Phát triển hệ thống theo dõi nhiệt độ, độ ẩm của đất áp dụng cho nhiều khu vườn khác nhau. Bật tắt thiết bị tưới nước phù hợp khi độ ẩm của đất thực tế không đủ theo thiết lập cho phép. Ghi nhận lại các hoạt động.

1.2 Yêu cầu

Hệ thống cần hiện thực thành công 5 module dưới đây:

Module 1: Nhận và hiển thị dữ liệu từ thiết bị:

- User: Xem nhiệt độ, độ ẩm của khu vườn ở thời điểm hiện tại
- Admin: Xem/ điều chỉnh/ xóa những tiêu chuẩn theo từng loại cây trồng.

Module 2: Kiểm tra dữ liệu nhận được vượt quá ngưỡng cho phép.:

- User: Kiểm tra dữ liệu từ thiết bị so sánh với ngưỡng an toàn cho phép – user đặt ra.

Module 3: Điều khiển thiết bị.

- User: Sau khi kiểm tra, nếu vượt ngưỡng cho phép điều khiển máy bơm mini.
- Admin: Xem/ điều chỉnh/ xóa những tiêu chuẩn theo từng loại cây trồng.

Module 4: Ghi nhận hoạt động:

- System: Sau mỗi lần kiểm tra, hệ thống lưu lại kết quả (sử dụng Real time clock).

Module 5: Tổ chức lưu trữ dữ liệu nghiệp vụ gồm dữ liệu các thiết bị và các kiểu thực thể khác trong hệ thống, lập trình web app backend và kết nối được với web app client, iot server, business server và thiết bị.

1.3 Phân tích chức năng

Ứng dụng bao gồm hai nhóm người dùng là admin và user

	Admin	User
Mô tả	Người quản trị hệ thống	Người quản lý, sử dụng hệ thống khu vườn thông minh
Vai trò, nhiệm vụ	Quản lý tất cả các tài khoản user, các khu vườn và toàn bộ các thiết bị, cảm biến có trong hệ thống	Có quyền truy cập thông tin và tham gia quản lý , điều chỉnh những khu vườn mà mình được cấp quyền (do admin cấp)

Tương ứng với đó là ba nhóm chức năng

Chức năng đăng ký, xác thực	<ul style="list-style-type: none">- Đăng nhập (admin và user)- Đăng xuất (admin và user)- Tạo tài khoản (user)
Chức năng quản lý (admin)	<ul style="list-style-type: none">- Xem thông tin tài khoản user- Xóa tài khoản user- Thay đổi quyền truy cập thông tin khu vườn- Thêm, xóa khu vườn và các thiết bị- Thiết lập định mức nước, độ ẩm cho khu vườn
Chức năng dành cho user	<ul style="list-style-type: none">- Xem nhiệt độ, độ ẩm của khu vườn- Bật/tắt máy bơm- Thay đổi thông tin cá nhân- Xem lịch sử hệ thống- Xem lịch sử hoạt động của bản thân- Lập lịch tự động bật tắt máy bơm

2 Phân công công việc

2.1 Đánh giá chung

Đánh giá chung					
Phương diện	Hữu Hưng	Việt Hưng	Quang Huy	Nguyễn Khang	Long Huy
Mức độ tham gia	13	13	13	13	3
Đánh giá chung	100%	100%	100%	100%	0%

2.2 Chi tiết giai đoạn

Giai đoạn 1	Từ đầu đến giữa kỳ				
Công việc	Hữu Hưng	Việt Hưng	Quang Huy	Nguyễn Khang	Long Huy
Phân tích đề tài, yêu cầu chức năng	1	1	1	1	
Tìm hiểu công nghệ	1	1	1	1	1
Thiết kế use-case	1	1	1	1	1
Đặc tả use-case	1	1	1	1	1
Thiết kế giao diện dự kiến	1	1		1	
Thiết kế cơ sở dữ liệu			1		
Mô tả dữ liệu			1		
Làm các chức năng back-end			1		
Hiện thực Login UI	1	1			
Hiện thực Admin UI	1			1	
Hiện thực User UI	1	1			
Hiện thực server MQTT			1		
Demo giữa kỳ	1	1	1	1	
Tổng (13)	9	8	9	7	3

Giai đoạn 2	Từ giữa kỳ đến cuối kỳ				
Công việc	Hữu Hưng	Việt Hưng	Quang Huy	Nguyễn Khang	Long Huy
Sửa UI		1		1	
Thêm đồ thị		1		1	
Filter, search, sort chức năng	1			1	
Hoàn thiện tính năng	1	1	1	1	
Sửa Backend			1		
Sửa lỗi	1	1	1	1	
Viết báo cáo	1	1	1	1	
Tổng (7)	4	5	4	6	

3 Tìm hiểu công nghệ

3.1 Giao thức MQTT

3.1.1 MQTT là gì ?

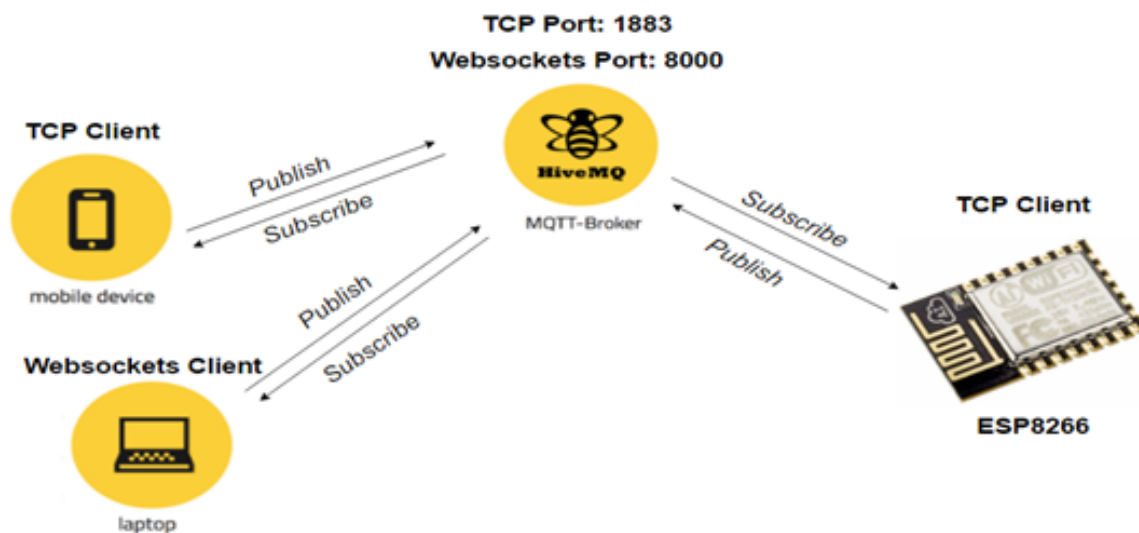
MQTT (Message Queuing Telemetry Transport) là một giao thức truyền thông điệp (message) theo mô hình publish/subscribe, sử dụng băng thông thấp, độ tin cậy cao và có khả năng hoạt động trong điều kiện đường truyền không ổn định.

3.1.2 Kiến trúc mức cao của MQTT

Kiến trúc mức cao (high-level) của MQTT gồm 2 phần chính là Broker (MQTT server) và Clients (devices/software components):

- Broker được coi như trung tâm, nó là điểm giao của tất cả các kết nối đến từ client. Nhiệm vụ chính của broker là nhận message từ publisher, xếp các message theo hàng đợi rồi chuyển chúng tới một địa chỉ cụ thể.
- Client thì được chia thành 2 nhóm là publisher và subscriber. Client chỉ làm ít nhất một trong 2 việc là xuất (publish) các message lên một topic cụ thể hoặc đăng ký (subscribe) một topic nào đó để nhận message từ topic này

Mỗi client sẽ đăng ký theo dõi các kênh thông tin (topic) hoặc gửi dữ liệu lên kênh thông tin đó. Quá trình đăng ký này gọi là “subscribe” và hành động một client gửi dữ liệu lên kênh thông tin được gọi là “publish”. Mỗi khi kênh thông tin đó được cập nhật dữ liệu (dữ liệu này có thể đến từ các client khác) thì những client nào đã đăng ký theo dõi kênh này sẽ nhận được dữ liệu cập nhật đó



Mô hình giao thức MQTT được sử dụng

Figure 1: Mô hình giao thức MQTT được sử dụng

3.1.3 Topic

Topic có thể coi như một "đường truyền" logic giữa 2 điểm là publisher và subscriber. Về cơ bản, khi message được publish vào một topic thì tất cả những subscriber của topic đó sẽ nhận được message này.

Giao thức MQTT cho phép khai báo các topic kiểu phân cấp.

Giả sử chúng ta có một hệ thống cảm biến đo thông tin môi trường trong khu vườn của chúng ta. Như vậy, các topic phục vụ truyền tải thông tin môi trường cho khu vườn của chúng ta có thể được khai báo như sau:

- garden/temperature: topic thông tin nhiệt độ khu vườn
- garden/humidity: topic thông tin độ ẩm khu vườn
- garden/light: topic thông tin độ sáng khu vườn

3.1.4 MQTT Bridge:

MQTT Bridge là một tính năng của MQTT Broker cho phép các MQTT Broker có thể kết nối và trao đổi dữ liệu với nhau. Để sử dụng tính năng này, ta cần tối thiểu 2 Broker, trong đó, một Broker bất kỳ sẽ được cấu hình thành Bridge. Khi cấu hình MQTT bridge, ta cần lưu ý tới các thông số sau:

- **address:** địa chỉ của broker cần kết nối
- **bridge-protocol-version:** phiên bản của giao thức MQTT đang sử dụng chung cho 2 broker.
- **topic:** phần này định nghĩa 3 thông số: tên topic được trao đổi giữa 2 broker, chiều trao đổi (1 chiều hay 2 chiều) và topic mapping giữa 2 broker.

3.2 IoT gateway

3.2.1 IoT gateway là gì?

IoT gateway là một máy tính nhúng làm cầu nối kết nối giữa các cảm biến (sensors), tác nhân (actors) tới mạng diện rộng Internet hoặc mạng nội bộ Intranet.

Một số công nghệ truyền thông được sử dụng bởi các cảm biến: Bluetooth, WiFi, BLE, Zigbee, Z-Wave, 6LoWPAN, NFC, WiFi Direct, GSM, LTE, LoRa, NB-IoT, LTE-M,...

Mô hình của IoT gateway

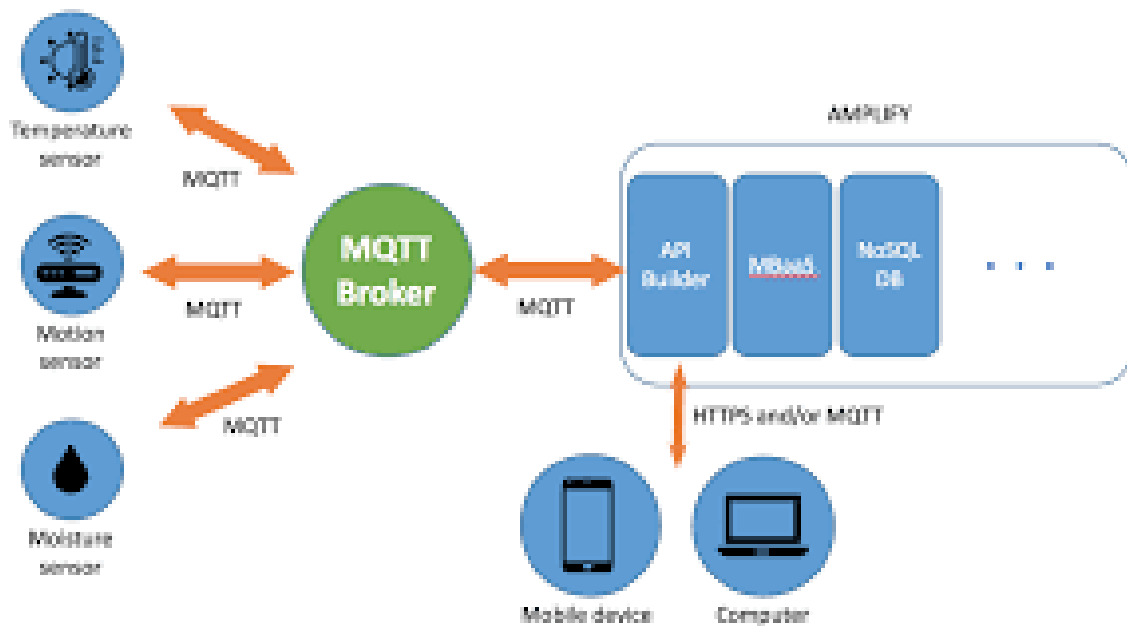


Figure 2: Mô hình của IoT gateway

Một số cảm biến có thể tạo ra hàng chục ngàn điểm dữ liệu mỗi giây. IoT gateway cung cấp một nơi để xử lý trước dữ liệu cục bộ ở vùng biên trước khi nó được gửi lên đám mây. Khi dữ liệu được tổng hợp, thu gọn và phân tích một cách khôn ngoan ở vùng biên, nó sẽ giảm thiểu khối lượng dữ liệu cần chuyển tiếp lên đám mây, điều này có thể tác động lớn đến thời gian hồi đáp và chi phí đường truyền mạng.

Một lợi ích khác của cổng IoT là nó có thể cung cấp cơ chế bảo mật bổ sung cho mạng IoT và dữ liệu mà nó vận chuyển. Vì gateway quản lý thông tin di chuyển theo cả hai chiều, nó có thể bảo vệ dữ liệu khi di chuyển lên đám mây khỏi các rò rỉ, hạn chế các thiết bị IoT bị xâm phạm bởi nguồn tấn công bên ngoài với các tính năng như phát hiện giả mạo, mã hóa, tạo số ngẫu nhiên bằng phần cứng và công cụ mã hóa.

3.2.2 Các chức năng của Iot gateway

Một IoT Gateway đa năng có thể thực hiện rất nhiều các nhiệm vụ, một số ví dụ như:

- Tạo điều kiện giao tiếp với các thiết bị cũ hoặc không có kết nối internet.
- Bộ nhớ đệm dữ liệu, caching và media streaming.
- Xử lý trước dữ liệu (data pre-processing), làm sạch, lọc và tối ưu hóa.
- Tổng hợp dữ liệu thô.
- Liên lạc Device-to-Device / Machine-to-Machine.
- Tính năng kết nối mạng và lưu trữ dữ liệu trực tiếp.
- Trực quan hóa dữ liệu và phân tích dữ liệu cơ bản thông qua các ứng dụng IoT Gateway.
- Tính năng lịch sử dữ liệu ngắn hạn.
- Bảo mật: quản lý truy cập người dùng và các tính năng bảo mật mạng.
- Quản lý cấu hình thiết bị.
- Chẩn đoán hệ thống.

3.3 ReactJs

3.3.1 Giới thiệu

React.js là một thư viện Javascript đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Nếu như AngularJS là một Framework cho phép nhúng code javascript trong code html thông qua các attribute như ng-model, ng-repeat...thì với react là một library cho phép nhúng code html trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS. Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn

3.3.2 React là gì?

React là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được. React được sử dụng tại Facebook trong production, và www.instagram.com được viết hoàn toàn trên React.

Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của lần render này với lần render trước và cập nhật ít thay đổi nhất trên DOM. Trước khi đến cài đặt và cấu hình, chúng ta sẽ đi đến một số khái niệm cơ bản:

- **Virtual DOM:** công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này. Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM tree thật. React sử dụng cơ chế one-way data binding – luồng dữ liệu 1 chiều. Dữ liệu được truyền từ parent đến child thông qua props. Luồng dữ liệu đơn giản giúp chúng ta dễ dàng kiểm soát cũng như sửa lỗi. Với các đặc điểm ở trên, React dùng để xây dựng các ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện. Ví dụ như Facebook: trên Newsfeed của bạn cùng lúc sẽ có các status khác nhau và mỗi status lại có số like, share, comment liên tục thay đổi. Khi đó React sẽ rất hữu ích để sử dụng.
- **JSX:** JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. Đặc điểm: Faster: Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javascript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript. Safer: an toàn hơn. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt. Easier: Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng
- **Components:** React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác. Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component. Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo.
- **Props và State:**
 - Props: giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến.

- State: thể hiện trạng thái của ứng dụng, khi state thay đổi thì component đồng thời render lại để cập nhật UI.

3.3.3 Tại sao sử dụng React?

Dễ sử dụng

React là một thư viện GUI nguồn mở JavaScript tập trung vào một điều cụ thể; hoàn thành nhiệm vụ UI hiệu quả. Nó được phân loại thành kiểu “V” trong mô hình MVC (Model-View-Controller).

Là lập trình viên JavaScript, bạn sẽ dễ dàng hiểu được những điều cơ bản về React. Bạn thậm chí có thể bắt đầu phát triển các ứng dụng dựa trên web bằng cách sử dụng react chỉ trong vài ngày.

Để củng cố hiểu biết của mình, bạn hãy thử khám phá thêm nhiều hướng dẫn về React. Chúng mang đến nhiều thông tin về cách sử dụng công cụ: videos, hướng dẫn và dữ liệu làm phong phú góc nhìn của bạn.

Nó hỗ trợ Reusable Component trong Java

React cho phép bạn sử dụng lại components đã được phát triển thành các ứng dụng khác có cùng chức năng. Tính năng tái sử dụng component là một lợi thế khác biệt cho các lập trình viên.

Viết component dễ dàng hơn

React component dễ viết hơn vì nó sử dụng JSX, mở rộng cú pháp tùy chọn cho JavaScript cho phép bạn kết hợp HTML với JavaScript.

JSX là một sự pha trộn tuyệt vời của JavaScript và HTML. Nó làm rõ toàn bộ quá trình viết cấu trúc trang web. Ngoài ra, phần mở rộng cũng giúp render nhiều lựa chọn dễ dàng hơn.

SX có thể không là phần mở rộng cú pháp phổ biến nhất, nhưng nó được chứng minh là hiệu quả trong việc phát triển components đặc biệt hoặc các ứng dụng có khối lượng lớn.

Hiệu suất tốt hơn với Virtual DOM

React sẽ cập nhật hiệu quả quá trình DOM (Document Object Model – Mô hình đối tượng tài liệu). Như bạn có thể biết, quá trình này có thể gây ra nhiều thất vọng trong các dự án ứng dụng dựa trên web. May mắn là React sử dụng virtual DOMs, vì vậy bạn có thể tránh được vấn đề này.

Công cụ cho phép bạn xây dựng các virtual DOMs và host chúng trong bộ nhớ. Nhờ vậy, mỗi khi có sự thay đổi trong DOM thực tế, thì virtual sẽ thay đổi ngay lập tức

Hệ thống này sẽ ngăn DOM thực tế để buộc các bản cập nhật được liên tục. Do đó, tốc độ của ứng dụng sẽ không bị gián đoạn.

Thân thiện với SEO

React cho phép bạn tạo giao diện người dùng có thể được truy cập trên các công cụ tìm kiếm khác nhau. Tính năng này là một lợi thế rất lớn vì không phải tất cả các khung JavaScript đều thân thiện với SEO.

Ngoài ra, vì React có thể tăng tốc quá trình của ứng dụng nên có thể cải thiện kết quả SEO. Cuối cùng tốc độ web đóng một vai trò quan trọng trong tối ưu hóa SEO.

Tuy nhiên, bạn cần lưu ý rằng React chỉ là một thư viện JavaScript. Nghĩa là nó không thể tự làm mọi thứ. Sử dụng các thư viện bổ sung có thể cần thiết cho các mục tiêu quản lý, định tuyến và tương tác.

3.4 Python

3.4.1 Python là gì?

Là một ngôn ngữ lập trình mã nguồn mở đa mục đích, một loại ngôn ngữ bậc cao, thông dịch và hướng đối tượng. Đây là một ngôn ngữ lập trình thông dịch được tạo ra bởi Guido van Rossum vào năm 1990.

Điểm đặc biệt của ngôn ngữ Lập trình Python có cú pháp khá dễ hiểu, dễ đọc và dễ học. Trong việc phát triển ứng dụng thì ngôn ngữ này cũng rất linh hoạt. Python hỗ trợ mẫu đa lập trình, bao gồm lập trình hướng đối tượng, lập trình hàm và mệnh lệnh hoặc là các phong cách lập trình theo thủ tục. Python là ngôn ngữ lập trình động nên không cần sử dụng các kiểu dữ liệu khai báo.

3.4.2 Ưu điểm

- **Đơn giản:** Là một ngôn ngữ có hình thức sáng sủa, cấu trúc rõ ràng, cú pháp ngắn gọn giúp người lập trình dễ dàng đọc và tìm hiểu.

- **Tốc độ xử lý khá nhanh:** được đánh giá nhanh hơn so với ngôn ngữ PHP. Với tốc độ xử lý cực nhanh, Python có thể tạo ra những chương trình từ những script siêu nhỏ tới những phần mềm cực lớn như Blender 3D..

- **Chất lượng:** Thư viện có tiêu chuẩn cao, Python có khối cơ sở dữ liệu khá lớn nhằm cung cấp giao diện cho tất cả các CSDL thương mại.

- **Thuận tiện:** Python được biên dịch và chạy trên tất cả các nền tảng lớn. Nó có trên tất cả các nền tảng hệ điều hành từ UNIX, MS – DOS, Mac OS, Windows và Linux và các OS khác thuộc họ Unix. Tương thích mạnh mẽ với Unix, hardware, third-party software với số lượng thư viện khổng lồ (400 triệu người sử dụng)

- **Mở rộng:** Với tính năng này, Python cho phép người lập trình có thể thêm hoặc tùy chỉnh các công cụ nhằm tối đa hiệu quả có thể đạt được trong công việc.

- **GUI Programming:** Giúp cho việc thực hiện ảnh minh họa di động một cách tự nhiên và sống động hơn.

3.4.3 Hạn chế

- Python không có các thuộc tính như :protected,private hay public, không có vòng lặp do...while và switch...case.

- Mặc dù tốc độ xử lý của Python nhanh hơn PHP nhưng không bằng Java và C++.

4 Thiết kế hệ thống

4.1 Use case

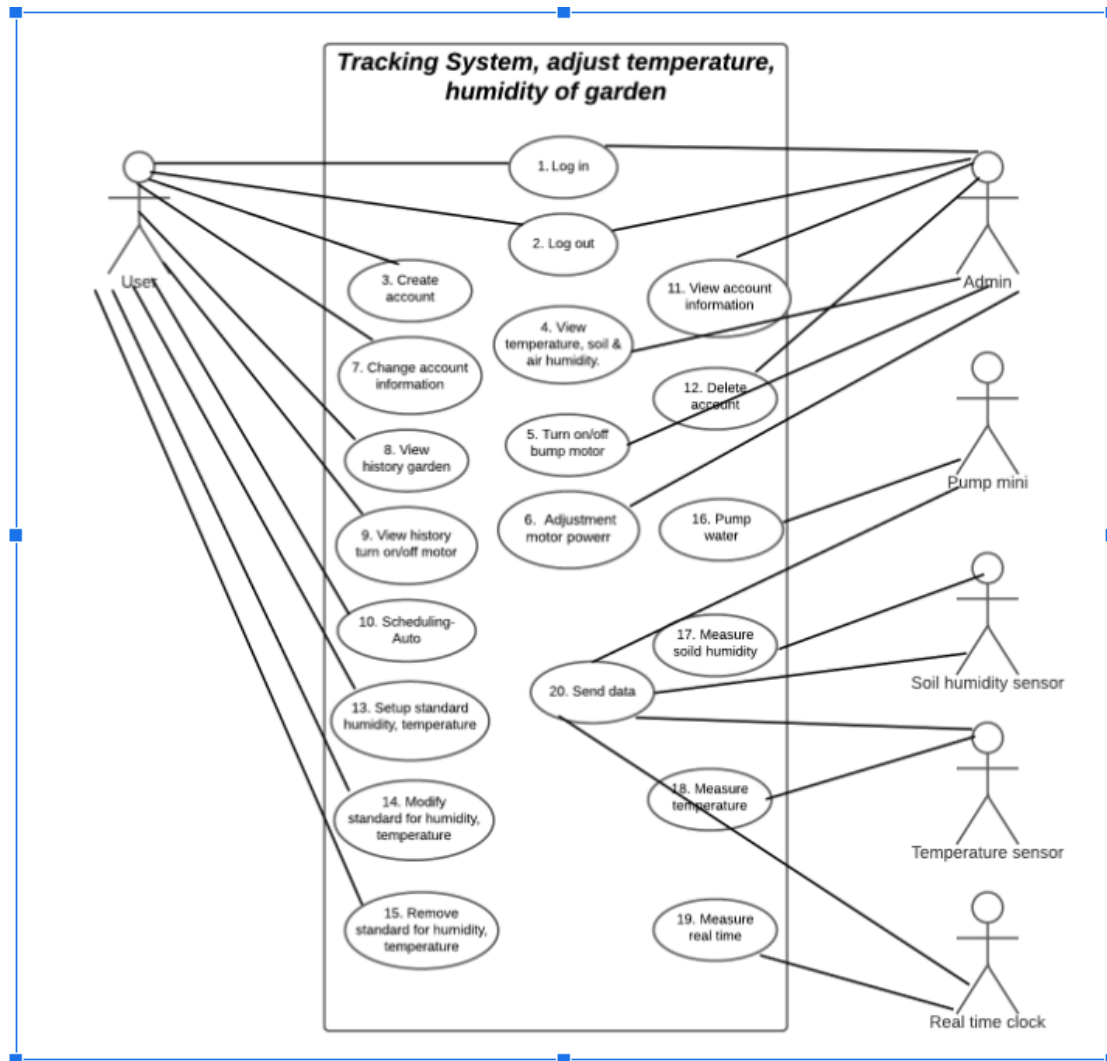


Figure 3: Use case diagram

4.2 Đặc tả use case

4.2.1 Login

Use-case name	Login
Actor	Admin, user
Description	Admin hoặc user đăng nhập vào ứng dụng
Preconditions	Admin được cấp tài khoản, user đã tạo tài khoản trước đó
Postconditions	Admin hoặc user vào được giao diện chính của ứng dụng
Normal Flows	1. Admin, user nhập username và password. 2. Chọn vai trò admin hay user (mặc định là user) 3. Nhấn nút “Log In” 4. Website chuyển tới giao diện tương ứng
Exceptions	Nếu admin, user chưa có tài khoản / nhập sai mật khẩu thì hiển thị thông báo “Invalid account or password”
Alternative Flows	Không có

4.2.2 Logout

Use-case name	Logout
Actor	Admin, user
Description	Admin hoặc user đăng xuất khỏi ứng dụng
Preconditions	Admin hoặc user đã đăng nhập vào ứng dụng
Postconditions	Admin hoặc user trở về giao diện đăng nhập của ứng dụng
Normal Flows	1. Admin, user Nhấn nút “Logout” ở góc trên bên trái màn hình. 2. Ứng dụng quay lại giao diện đăng nhập.
Exceptions	Không có
Alternative Flows	Không có

4.2.3 Create account

Use-case name	Create account
Actor	User
Description	User tạo tài khoản để đăng nhập vào ứng dụng
Preconditions	Tài khoản chưa được đăng ký trước đó
Postconditions	Một tài khoản mới được tạo và user có thể sử dụng nó để đăng nhập vào ứng dụng
Normal Flows	1. Ở giao diện đăng nhập, nhấn vào nút “Sign up”. 2. Nhập đầy đủ thông tin (full name, username, password,...). 3. Nhấn nút “Save”.
Exceptions	Không có
Alternative Flows	Không có

4.2.4 View temperature, soil humidity, air humidity in garden

Use-case name	View temperature, soil humidity, air humidity in garden
Actor	User/Admin
Description	User/Admin đăng nhập vào để xem nhiệt độ, độ ẩm đất, độ ẩm không khí trong khu vườn.
Preconditions	- User đã đăng nhập. - User có quyền truy cập vào khu vườn
Postconditions	Ứng dụng hiển thị thông tin nhiệt độ, độ ẩm đất, độ ẩm không khí của khu vườn
Normal Flows	1. Đăng nhập. 2. Ở giao diện chính của user, chọn khu vườn mà mình muốn xem (nếu có từ 2 khu vườn). 3. Nhiệt độ của khu vườn hiển thị dưới dạng “Nhiệt độ: XX°C” (XX trong khoảng từ 0-50) 4. Độ ẩm của đất được hiển thị dưới dạng “Độ ẩm đất: xxx” (xxx trong khoảng từ 0-100). 5. Độ ẩm của không khí được hiển thị dưới dạng “Độ ẩm không khí: xx” (xx trong khoảng từ 20-90).
Exceptions	User chưa được cấp quyền quản lý khu vườn thì hiển thị “No data”.
Alternative Flows	Không có

4.2.5 Turn on/off water pump motor

Use-case name	Turn on/off water pump motor
Actor	User/Admin
Description	User/Admin có quyền bật, tắt máy bơm nước.
Preconditions	- User đã đăng nhập. - User có quyền truy cập vào khu vườn
Postconditions	Thiết bị motor sẽ được bật hoặc tắt.
Normal Flows	1. Đăng nhập. 2. Ở giao diện chính của user, chọn khu vườn mà mình muốn xem (nếu có từ 2 khu vườn). 3. Nhấn chọn button “Motor”. 4. Nhấn chọn vào nút Bật/tắt motor. 5. Nhấn “Save”.
Exceptions	User chưa được cấp quyền quản lý khu vườn thì hiển thị “No data”.
Alternative Flows	Không có

4.2.6 Search/sort garden of user

Use-case name	Search/sort garden of user
Actor	User/Admin
Description	User/Admin tìm kiếm, sắp xếp khu vườn
Preconditions	- User/Admin đã đăng nhập. - User/Admin có quyền truy cập vào khu vườn
Postconditions	Tìm kiếm được khu vườn
Normal Flows	1. Đăng nhập. 2. Ở giao diện chính của user, nhập văn bản muốn search). 3. Nhấn chọn button “Search”, sort name theo ASC, DESC. Search theo value, status hoặc
Exceptions	Không ra được khu vườn muốn search.
Alternative Flows	Không có

4.2.7 Change account information.

Use-case name	Change account information
Actor	User
Description	User có thể thay đổi các thông tin của tài khoản: tên đăng nhập, mật khẩu, email, số điện thoại,.....
Preconditions	User đã đăng nhập vào hệ thống.
Postconditions	Thông tin tài khoản được chỉnh sửa
Normal Flows	1. Ở giao diện chính dành cho user, chọn “My Account”. 2. Chọn vào “Change information” và chọn vào các trường thông tin cần chỉnh sửa. 3. Chỉnh sửa thông tin và nhấn “Save”.
Exceptions	Không có.
Alternative Flows	Không có.

4.2.8 Checking information of garden

Use-case name	Checking information of garden
Actor	User
Description	User có thể xem lịch sử, tìm kiếm các giá trị của soil, nhiệt độ, độ ẩm không khí của khu vườn dưới dạng biểu đồ.
Preconditions	- User đã đăng nhập. - Dữ liệu đã được lưu vào Database
Postconditions	Ứng dụng hiển thị lịch sử đo các thông số của khu vườn
Normal Flows	1. Ở giao diện chính của User , chọn "History". 2. Chọn một trong các button: "Temperature" để xem lịch sử nhiệt độ; "Air humidity" để xem lịch sử độ ẩm không khí và "Soil humidity" để xem lịch sử độ ẩm đất. 3. Web sẽ hiển thị một biểu đồ thời gian và đại lượng tương ứng, thông tin về đại lượng đo được trong khu vườn trong vòng một ngày. 4. Có thể chọn qua các ngày khác hoặc qua khu vườn khác mà tài khoản đang quản lý.
Exceptions	Nếu khu vườn chưa ghi nhận dữ liệu, chưa được đo thì sẽ hiện thông báo "No Data"
Alternative Flows	Không có.

4.2.9 View history turn on/off motor

Use-case name	View history turn on/off motor
Actor	User
Description	User có thể xem lịch sử bật tắt, và điều chỉnh công suất động cơ bơm nước..
Preconditions	- User đã đăng nhập. - Dữ liệu đã được lưu vào Database
Postconditions	Ứng dụng hiển thị lịch sử bật tắt và điều chỉnh công suất động cơ bơm nước.
Normal Flows	1. Ở giao diện chính của User , chọn "History". 2. Chọn button "History Motor", hiển thị các lần bật tắt và điều chỉnh công suất do user và cả tự động của hệ thống. 3. Kéo xuống, chọn các khu vườn khác để xem.
Exceptions	Nếu khu vườn chưa ghi nhận dữ liệu, chưa được đo thì sẽ hiện thông báo "No Data"
Alternative Flows	Không có.

4.2.10 View account information

Use-case name	View account information
Actor	Admin
Description	Admin xem chi tiết thông tin tài khoản của user.
Preconditions	Admin đã đăng nhập.
Postconditions	Admin xem được tất cả thông tin tài khoản
Normal Flows	1. Ở giao diện dành cho admin, chọn “User Account”. 2. Nhấn chọn tài khoản muốn xem thông tin.
Exceptions	Không có.
Alternative Flows	Không có.

4.2.11 Delete account

Use-case name	Delete account
Actor	Admin
Description	Admin xóa tài khoản của user.
Preconditions	Admin đã đăng nhập.
Postconditions	Xóa thành công user
Normal Flows	1. Ở giao diện dành cho admin, chọn “User Account”. 2. Nhấn chọn tài khoản muốn xóa. 3. Nhấn chọn biểu tượng thùng rác.
Exceptions	Không có.
Alternative Flows	Không có.

4.2.12 Quy định mức giới hạn của độ ẩm, nhiệt độ

Use-case name	Quy định mức giới hạn của độ ẩm, nhiệt độ
Actor	User
Description	User quy định mức giới hạn của độ ẩm, nhiệt độ
Preconditions	User đã đăng nhập.
Postconditions	Đặt thành công tiêu chuẩn cho khu vườn
Normal Flows	1. Ở giao diện dành cho User. 2. Chọn tạo Giới hạn cho độ ẩm, nhiệt độ 3. Chọn tạo. 4. User nhập thông tin cần thiết và nhấn nút Save
Exceptions	Không có.
Alternative Flows	Không có.

4.2.13 Sửa đổi mức giới hạn của độ ẩm, nhiệt độ

Use-case name	Xóa mức giới hạn của độ ẩm, nhiệt độ
Actor	User
Description	User xóa mức giới hạn của độ ẩm, nhiệt độ
Preconditions	User đã đăng nhập.
Postconditions	Sửa đổi tiêu chuẩn của khu vườn
Normal Flows	1. Ở giao diện dành cho User. 2. Chọn tạo Giới hạn cho độ ẩm, nhiệt độ 3. Chọn Độ ẩm/nhiệt độ mà mình muốn xóa 4. Nhấn nút delete
Exceptions	Không có.
Alternative Flows	Không có.

4.2.14 Pump water

Use-case name	Pump water
Actor	Pump mini
Description	Pump mini có nhiệm vụ bơm nước cho vườn
Preconditions	Người dùng bật máy bơm hoặc bơm tự động theo lịch
Postconditions	Sửa đổi tiêu chuẩn của khu vườn
Normal Flows	
Exceptions	Không có.
Alternative Flows	Không có.

4.2.15 Measure soil humidity

Use-case name	Measure soil humidity
Actor	Soil humidity sensor
Description	Máy cảm biến độ ẩm có nhiệm vụ đo độ ẩm đất
Preconditions	Máy cảm biến độ ẩm được bật
Postconditions	Máy cảm biến độ ẩm đo chính xác độ ẩm của đất
Normal Flows	
Exceptions	Không có.
Alternative Flows	Không có.

4.2.16 Measure temperature

Use-case name	Measure temperature
Actor	Temperature sensor
Description	Máy cảm biến nhiệt độ có nhiệm vụ đo nhiệt độ không khí
Preconditions	Máy cảm biến nhiệt độ được bật
Postconditions	Máy cảm biến nhiệt độ đo chính xác nhiệt độ không khí
Normal Flows	
Exceptions	Không có.
Alternative Flows	Không có.

4.2.17 Measure real time

Use-case name	Measure real time
Actor	Real time clock
Description	Đồng hồ hiện thời gian thực
Preconditions	Đồng hồ được bật
Postconditions	Đồng hồ trả về đúng thời gian thực trong ngày
Normal Flows	
Exceptions	Không có.
Alternative Flows	Không có.

4.2.18 Send data

Use-case name	Send data
Actor	Soil humidity sensor, Temperature sensor, Real time clock
Description	Các sensor và đồng hồ gửi dữ liệu về cho server
Preconditions	Các sensor và đồng hồ được bật
Postconditions	Server nhận dữ liệu chính xác và kịp thời
Normal Flows	
Exceptions	Không có.
Alternative Flows	Không có.

4.3 Giao diện dự kiến

4.3.1 Giao diện chung

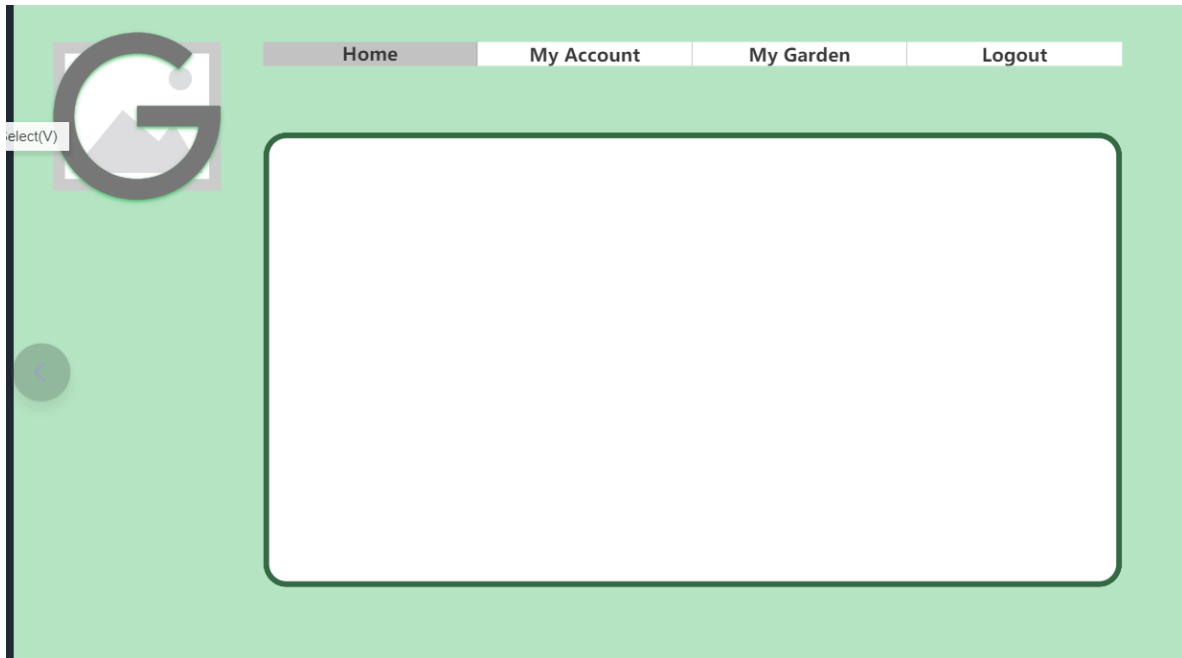


Figure 4: Giao diện trang chủ

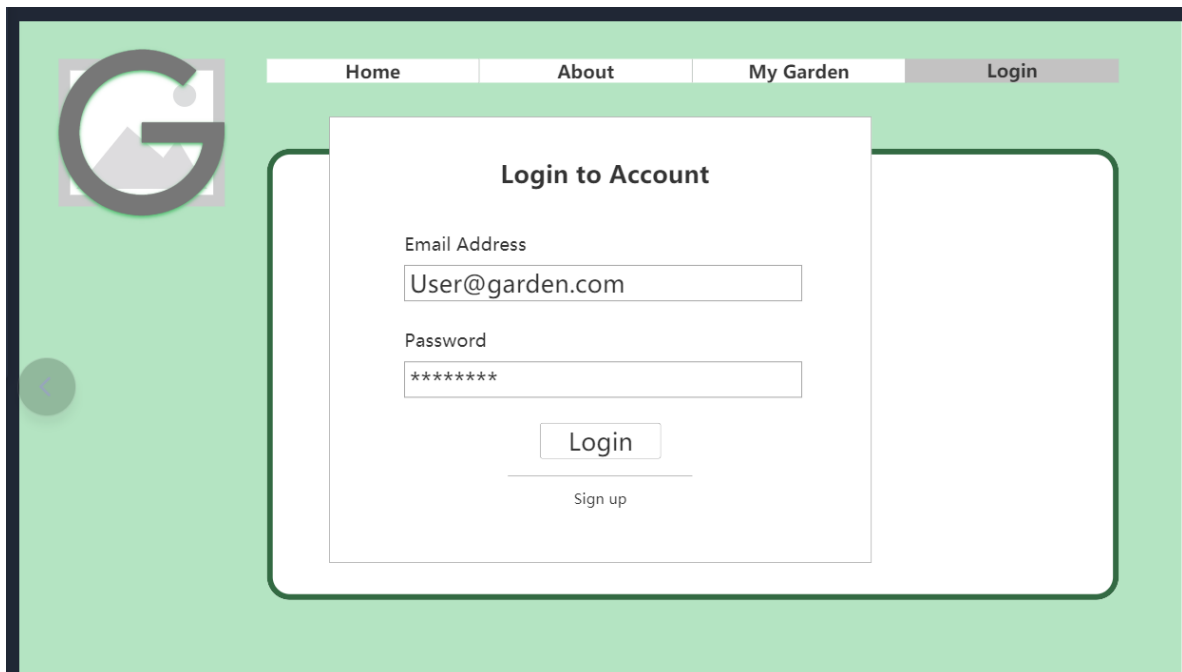


Figure 5: Giao diện đăng nhập

4.3.2 Giao diện admin

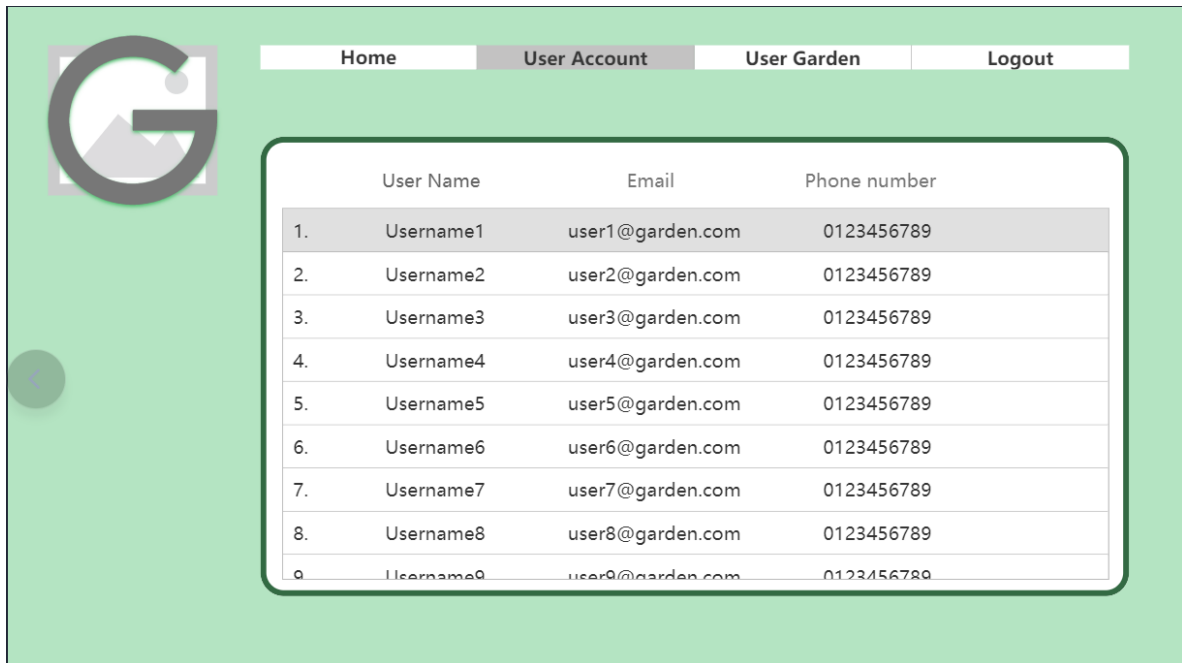


Figure 6: Giao diện admin quản lý tài khoản user

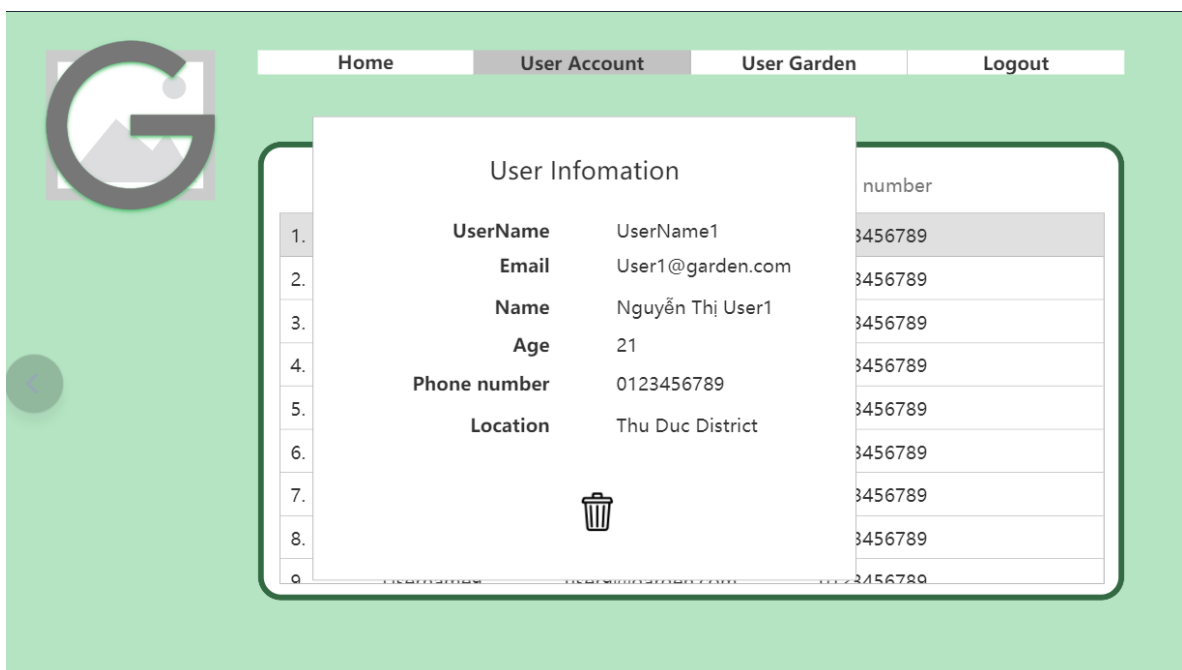


Figure 7: Giao diện admin xem thông tin user

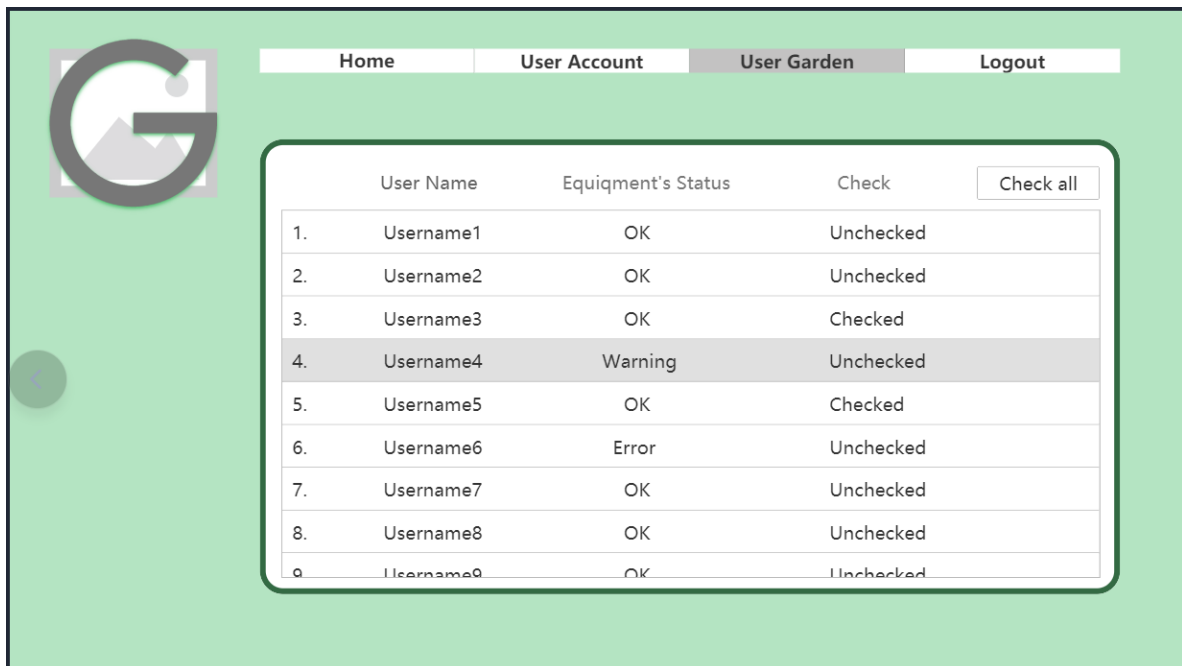


Figure 8: Giao diện admin quản lý khu vườn user

4.3.3 Giao diện user

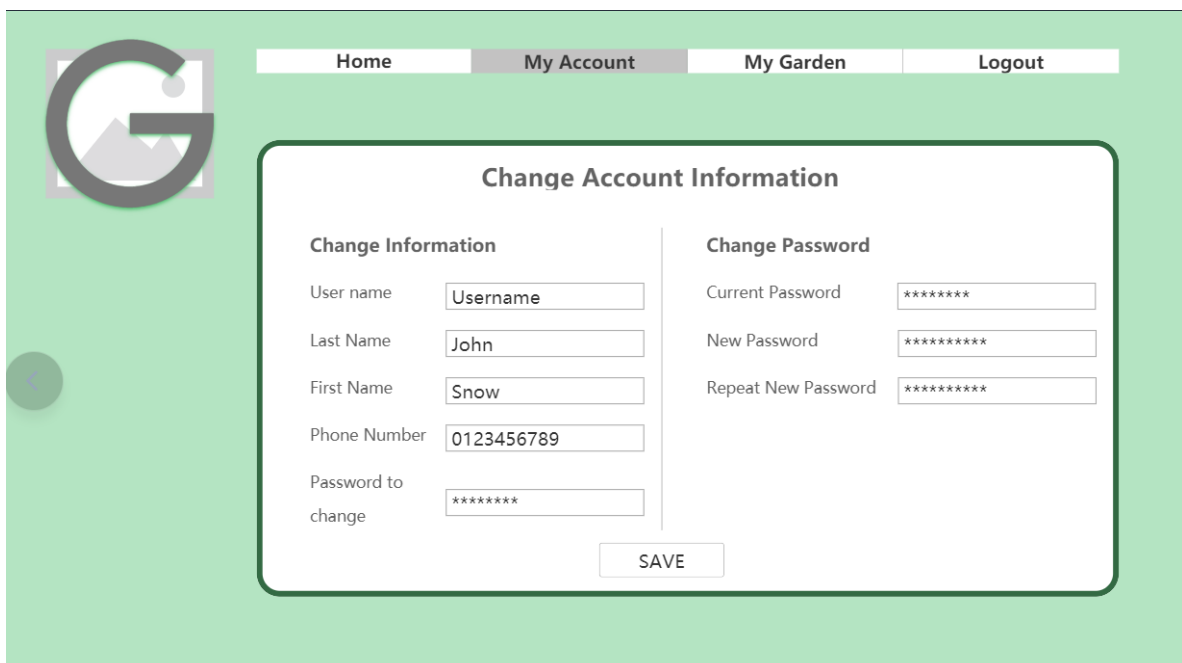


Figure 9: Giao diện user chỉnh sửa thông tin

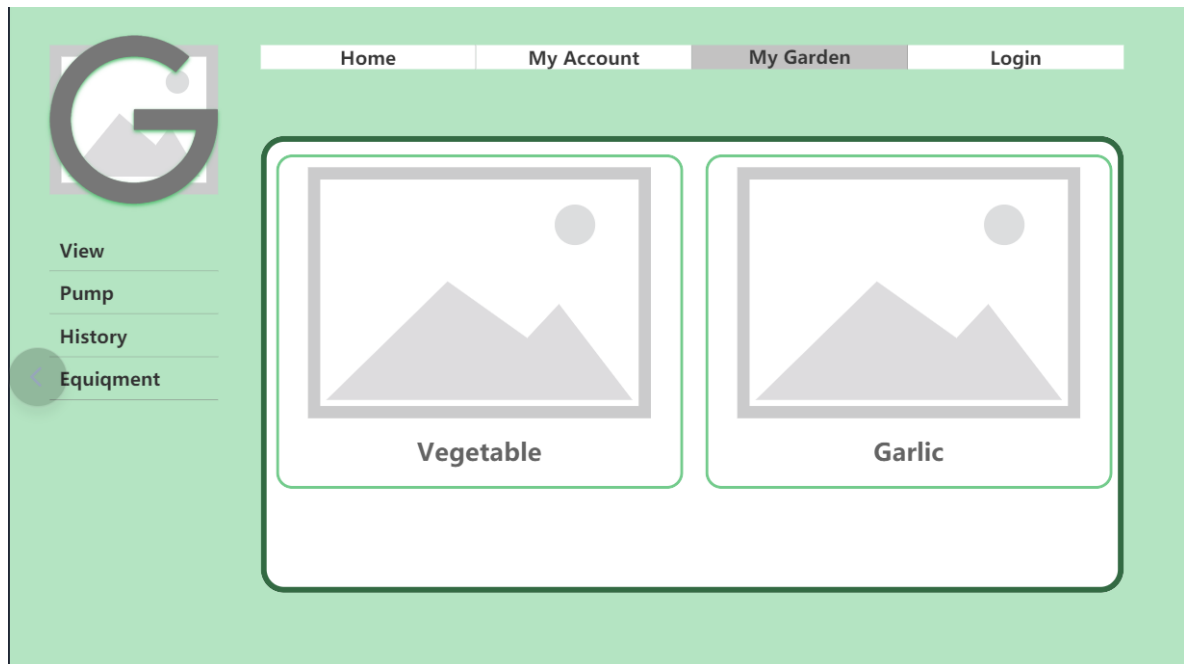


Figure 10: Giao diện user xem hình ảnh khu vườn

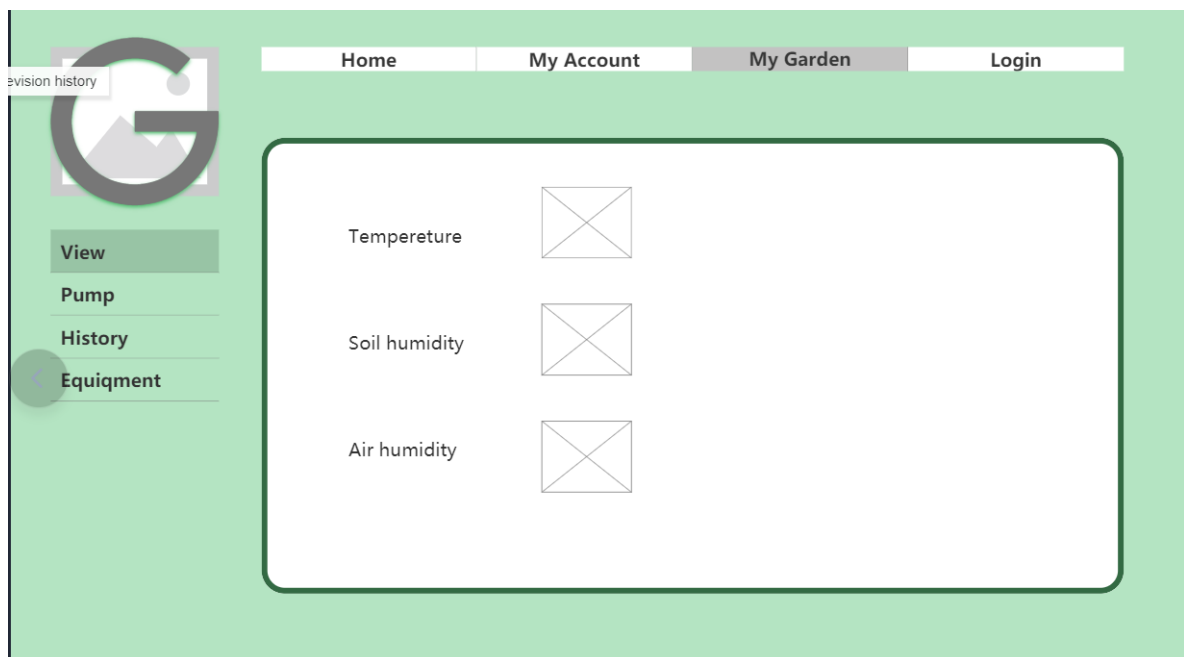


Figure 11: Giao diện user quản lý các số liệu của khu vườn

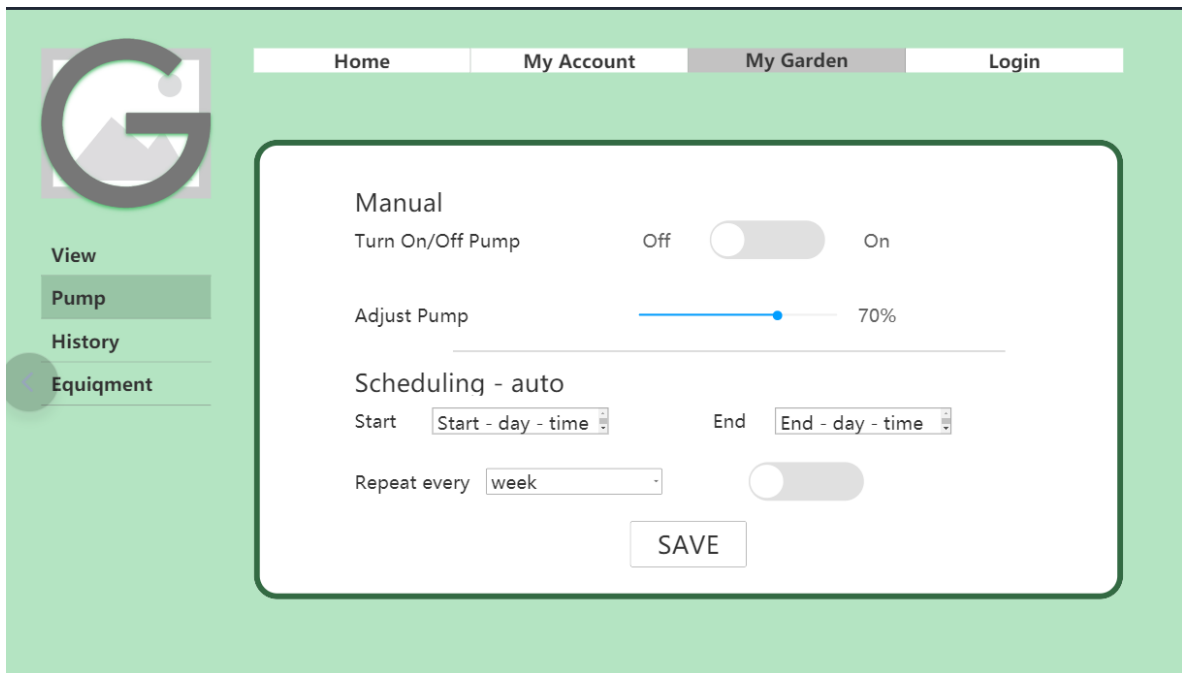


Figure 12: Giao diện user điều chỉnh máy bơm

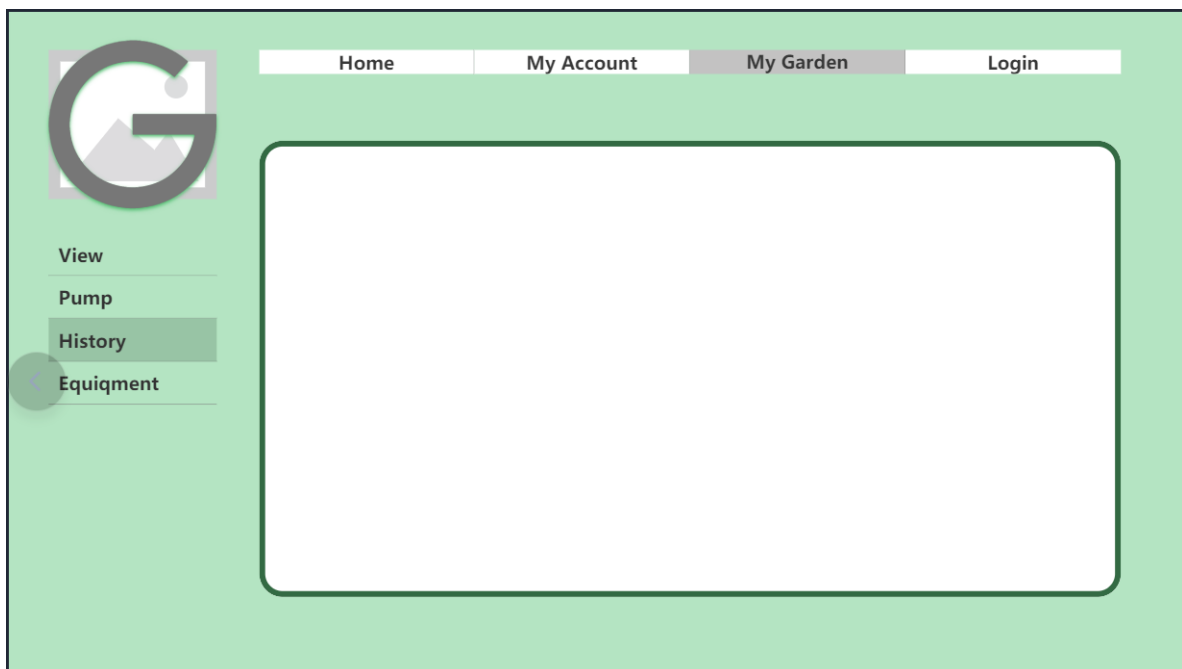
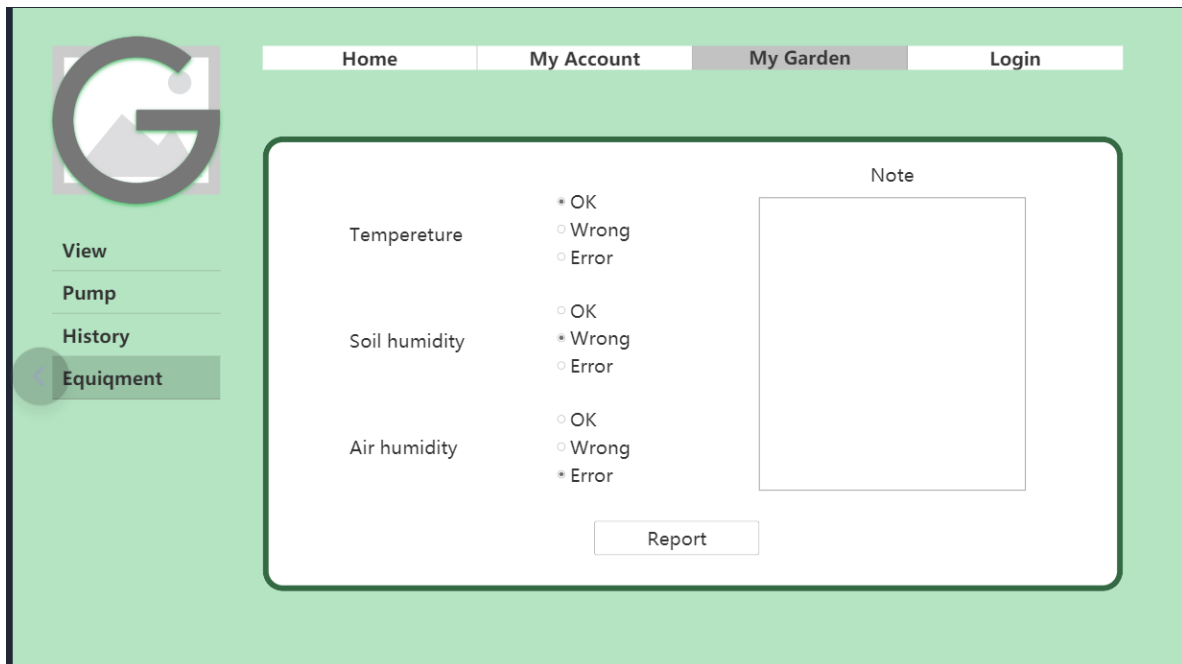


Figure 13: Giao diện user xem lịch sử



The image shows a web application interface for managing equipment. It features a green background with a sidebar on the left and a main content area. The sidebar contains a large 'G' logo and a menu with options: View, Pump, History, and Equipment (which is highlighted). The main content area has a top navigation bar with links: Home, My Account, My Garden (active), and Login. Below this, there is a form for reporting equipment status. The form includes three rows of data: Temperature, Soil humidity, and Air humidity. Each row has three radio buttons for status selection: OK, Wrong, and Error. A 'Report' button is located at the bottom of the form. To the right of the form is a 'Note' section with a large text area.

Parameter	Status
Temperature	<input checked="" type="radio"/> OK <input type="radio"/> Wrong <input type="radio"/> Error
Soil humidity	<input type="radio"/> OK <input checked="" type="radio"/> Wrong <input type="radio"/> Error
Air humidity	<input type="radio"/> OK <input type="radio"/> Wrong <input checked="" type="radio"/> Error

Note

Figure 14: Giao diện user quản lý trang thiết bị

4.4 Cơ sở dữ liệu

Cơ sở dữ liệu sử dụng ngôn ngữ MySQL, đồng thời sử dụng Xampp để hỗ trợ Apache và dễ dàng quản lí database.

4.4.1 Mô hình dữ liệu

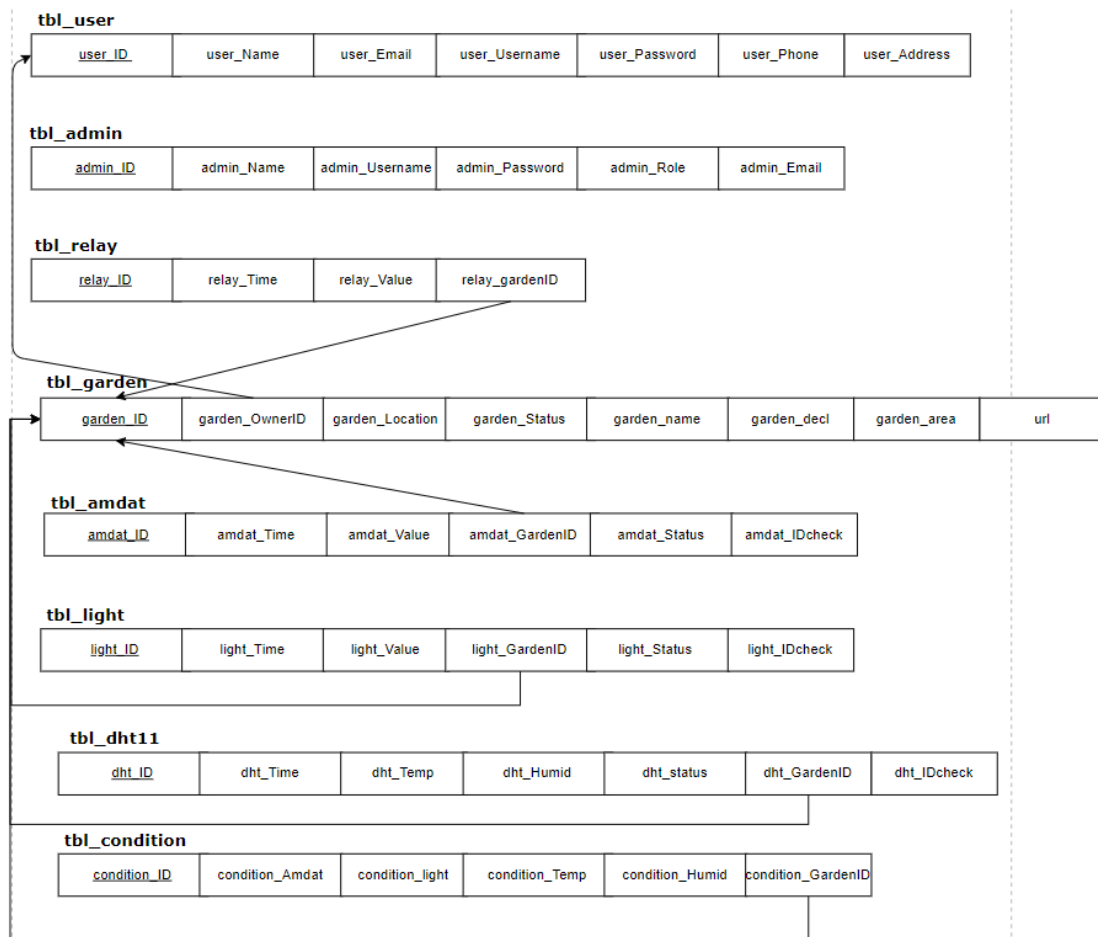


Figure 15: Mô hình dữ liệu quan hệ

4.4.2 Mô tả dữ liệu

- Bảng 1: tbl-admin : chứa thông tin của các admin của hệ thống.

tbl-admin		
Trường	Kiểu dữ liệu	Mô tả
admin-ID	int(11)	ID
admin-Name	varchar(255)	Tên admin
admin-Username	varchar(255)	Tên đăng nhập
admin-Password	varchar(255)	Mật khẩu
admin-Role	varchar(50)	Quyền quản trị
admin-Email	varchar(255)	Địa chỉ Email

- Bảng 2: tbl-amdat : chứa thông tin của thiết bị đo độ ẩm đất.

tbl-amdat		
Trường	Kiểu dữ liệu	Mô tả
amdat-ID	int(11)	ID
amdat-Time	timestamp	Thời gian đo được
amdat-Value	varchar(255)	Giá trị độ ẩm đất
amdat-GardenID	int(11)	ID của khu vườn
amdat-Status	varchar(150)	Trạng thái độ ẩm đất
amdat-IDcheck	varchar(150)	ID trên Adafruit

- Bảng 3: tbl-condition : chứa giá trị ngưỡng của các thiết bị.

tbl-condition		
Trường	Kiểu dữ liệu	Mô tả
conditon-ID	int(11)	ID
conditon-Amdat	varchar(255)	Ngưỡng độ ẩm đất
conditon-Light	varchar(255)	Ngưỡng độ sáng
condition-Temp	varchar(255)	Ngưỡng nhiệt độ
conditon-Humid	varchar(255)	Ngưỡng độ ẩm không khí

- Bảng 4: tbl-dht11 : chứa thông tin của thiết bị dht11.

tbl-dht11		
Trường	Kiểu dữ liệu	Mô tả
dht-ID	int(11)	ID
dht-Time	timestamp	Thời gian lúc đo được
dht-Temp	varchar(255)	Giá trị nhiệt độ
dht-Humid	varchar(255)	Giá trị độ ẩm không khí
dht-Status	varchar(150)	Trạng thái nhiệt độ và độ ẩm
amdat-IDcheck	varchar(150)	ID trên Adafruit

- Bảng 5: tbl-garden : chứa thông tin của khu vườn.

tbl-garden		
Trường	Kiểu dữ liệu	Mô tả
garden-ID	int(11)	ID khu vườn
garden-OwnerID	int(11)	ID của user
garden-Location	varchar(255)	Vị trí khu vườn
garden-Status	varchar(150)	Trạng thái khu vườn
garden-name	varchar(150)	Tên khu vườn
garden-decl	varchar(250)	Miêu tả khu vườn
garden-area	decimal	Diện tích khu vườn
url	varchar(250)	Ảnh khu vườn

- Bảng 6: tbl-light : chứa thông tin của thiết bị đo độ sáng.

tbl-light		
Trường	Kiểu dữ liệu	Mô tả
light-ID	int(11)	ID
light-Time	timestamp	Thời gian đo được
light-Value	varchar(255)	Giá trị độ sáng
light-GardenID	int(11)	ID của khu vườn
light-Status	varchar(150)	Trạng thái độ sáng
light-IDcheck	varchar(150)	ID trên Adafruit

- Bảng 7: tbl-relay : chứa thông tin của thiết bị relay.

tbl-relay		
Trường	Kiểu dữ liệu	Mô tả
relay-ID	int(11)	ID
relay-Time	timestamp	Thời gian đo được
relay-Value	varchar(50)	Giá trị ON/OFF
relay-GardenID	int(11)	ID của khu vườn

- Bảng 8: tbl-user: chứa thông tin của các user của hệ thống.

tbl-user		
Trường	Kiểu dữ liệu	Mô tả
user-ID	int(11)	ID
user-Name	varchar(255)	Tên admin
user-Username	varchar(255)	Tên đăng nhập
user-Password	varchar(255)	Mật khẩu
user-Address	varchar(50)	Địa chỉ nơi ở
user-Email	varchar(255)	Địa chỉ Email
user-Phone	varchar(255)	Số điện thoại

4.5 Giao tiếp server

- Server giao tiếp có chức năng kết nối giữa ứng dụng, database với server AdafruitIO. Server giao tiếp nhận nhiệm vụ truyền tải dữ liệu từ Server AdafruitIO đồng thời nhận dữ liệu từ ứng dụng thông qua các chức năng như: receive, publish, ...

- Server giao tiếp được chạy trực tiếp khi ứng dụng chạy, đồng thời là chạy Rest API cho ứng dụng.
- Server giao tiếp được thực hiện bằng thư viện MQTTClient Python.
- Receive:
 - Server giao tiếp sẽ subscribe vào Server AdafruitIO thông qua hàm Receive(). Trước khi gọi hàm Receive thì cần tạo một Object của Client mang các thuộc tính gồm: username và key. Tham số truyền vào trong hàm Receive của thư viện MQTTClient là tên của feed mà server giao tiếp cần lấy dữ liệu từ server AdafruitIO.
 - Sau khi nhận được dữ liệu từ server AdafruitIO, server giao tiếp sẽ đưa dữ liệu vào trong database của ứng dụng. Từ đó, ứng dụng có thể sử dụng dữ liệu của Server AdafruitIO thông qua việc nhận dữ liệu khi gọi các Rest API.
- Publish:
 - Khi user sử dụng ứng dụng thay đổi trạng thái của thiết bị, ví dụ như thay đổi relay từ ON sang OFF thì ứng dụng sẽ gọi API, sử dụng phương thức POST để cập nhập dữ liệu trên database.
 - Đồng thời khi server giao tiếp sẽ thực hiện chức năng Publish bằng cách gọi hàm Send-data() của thư viện MQTTClient, để publish dữ liệu đó lên server AdafruitIO. Trong hàm Send-data() ta cũng truyền tên của feed và dữ liệu cần publish.

5 Mô tả chi tiết

5.1 Dịch vụ đăng ký xác thực

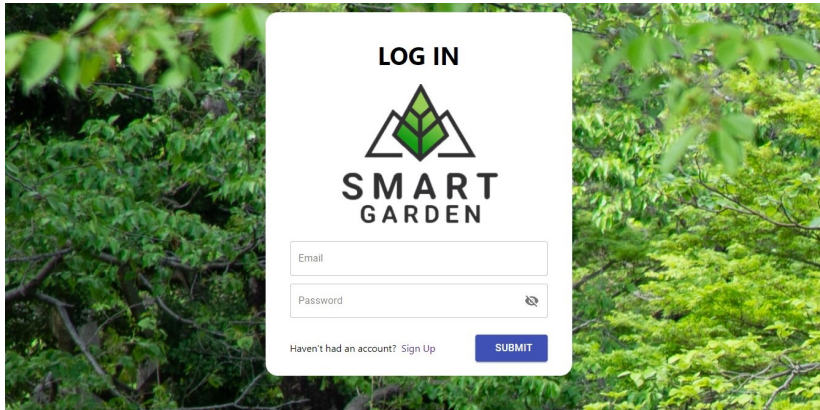


Figure 16: Login

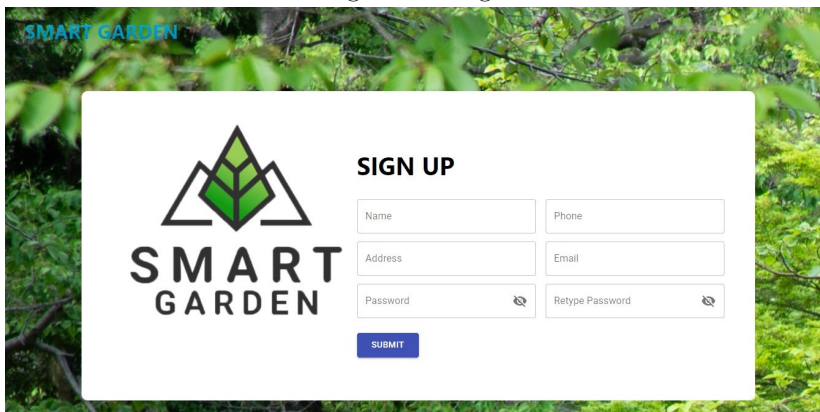


Figure 17: Sign Up

- Đây là giao diện đăng nhập, dành cho cả user và admin để đăng nhập vào hệ thống

Và giao chức năng đăng ký cho những user mới sử dụng đến hệ thống, thông tin đăng ký sẽ được lưu vào data và được quản lý bởi admin.

Khi đăng ký chỉ có thể đăng ký user, không thể đăng ký admin

5.2 Dịch vụ quản lý

- Đây là giao diện dashboard của Admin, nơi hiển thị các thông tin: bao gồm thông tin của các thiết bị và số lượng user đã đăng ký.

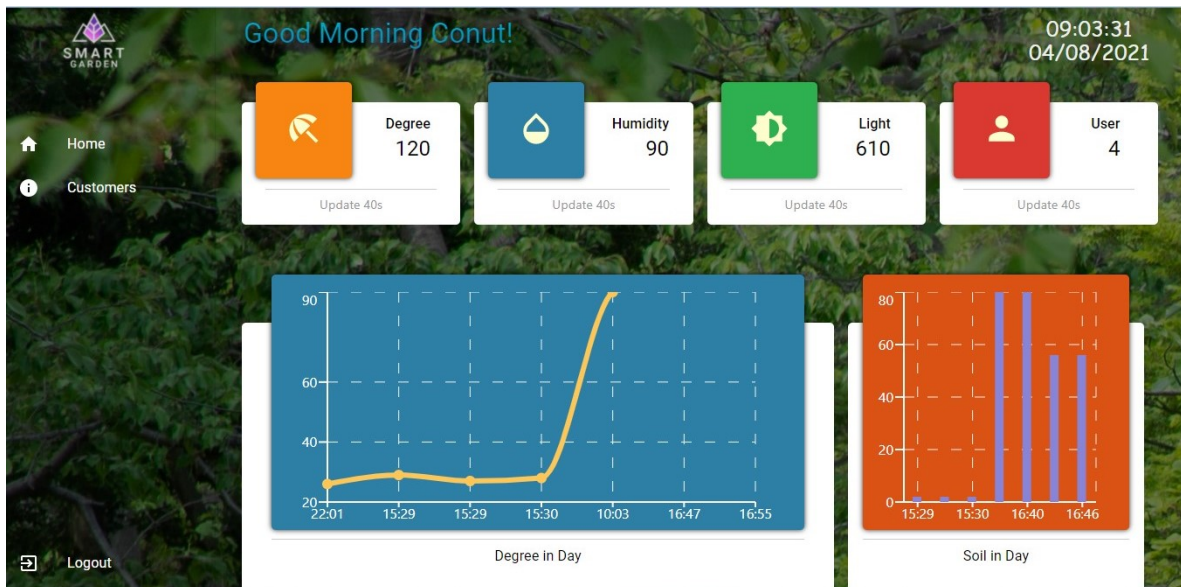


Figure 18: Dashboard

- Giao diện quản lý user hiển thị thông tin đăng ký của user và chức năng xóa user.

The management interface displays the following customer information:

Name	Email	Phone number	Action
nguyen_B	nguyenB@gmail.com	0123	
anguyen	anguyen@gmail.com	123456	
tranC	tranC@gmail.com	0123	
123	123@vinova.com.vn	1231231231	

Figure 19: Management

5.3 Dịch vụ người dùng

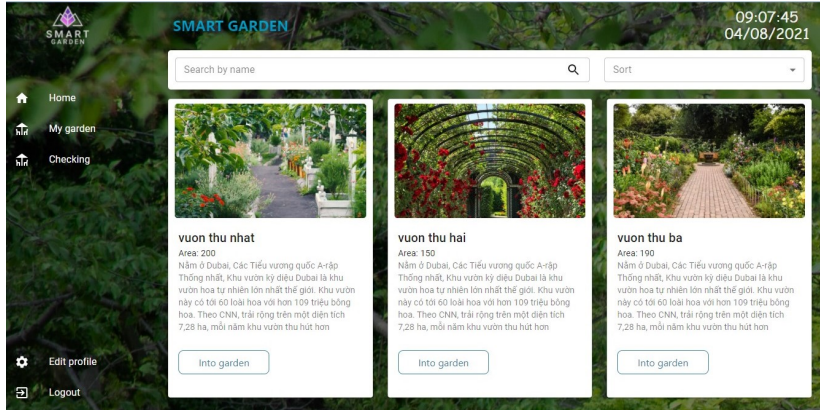


Figure 20: List Garden

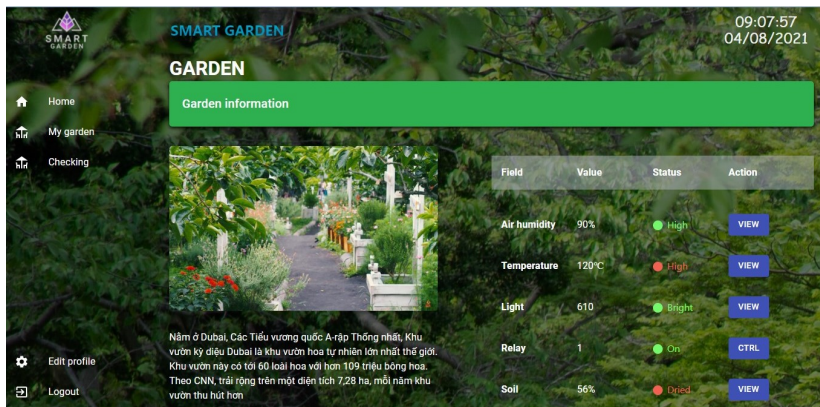


Figure 21: Thông tin của garden

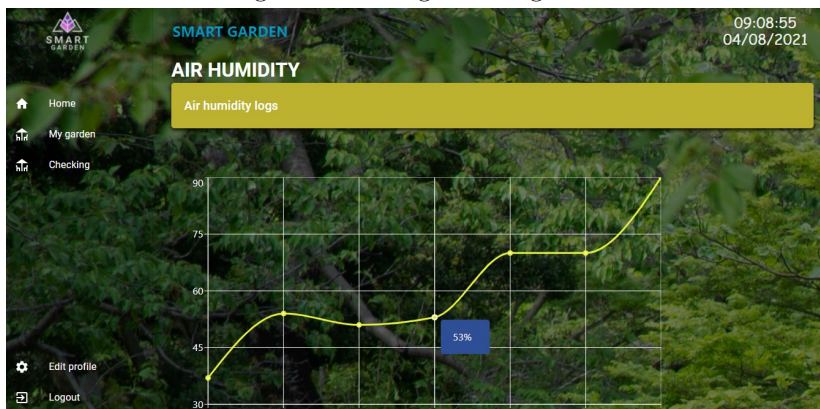


Figure 22: Dữ liệu của thiết bị

Người dùng sẽ có thể thấy được những khu vườn mà mình đang quản lý, ngoài ra còn có thể filter: là sắp xếp lại tên, hoặc tìm kiếm theo tên của khu vườn.

Khi vào một khu vườn, bên trong sẽ hiển thị các thông tin, dữ liệu của các thiết bị trong khu vườn, người dùng có thể xem, điều khiển các thiết bị.

Người dùng sẽ thấy được 7 giá trị mới nhất của thiết bị mà mình muốn xem thời tin, giá trị sẽ được thể hiện dưới dạng biểu đồ, và dạng log phía bên dưới để dễ quan sát (tương tự với các thiết bị lấy dữ liệu khác)

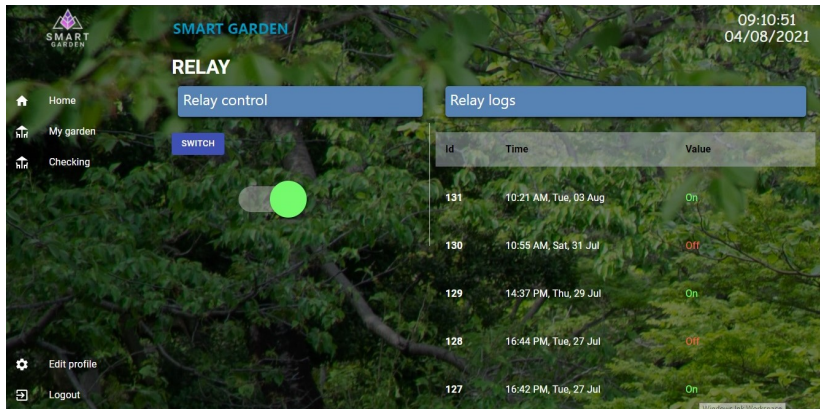


Figure 23: Điều khiển thiết bị

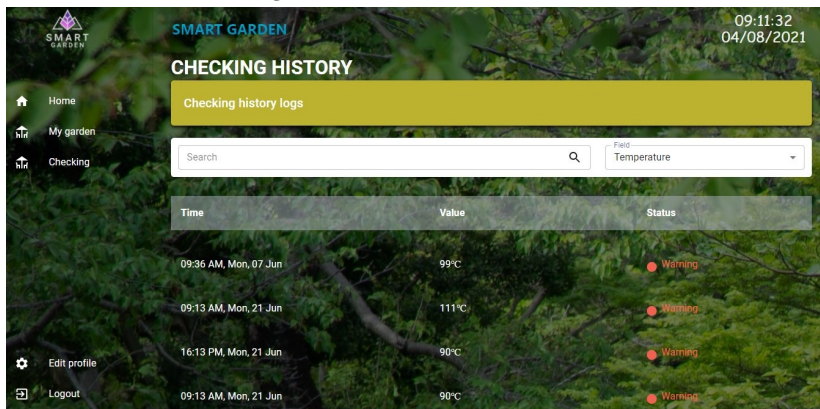


Figure 24: Kiểm tra dữ liệu

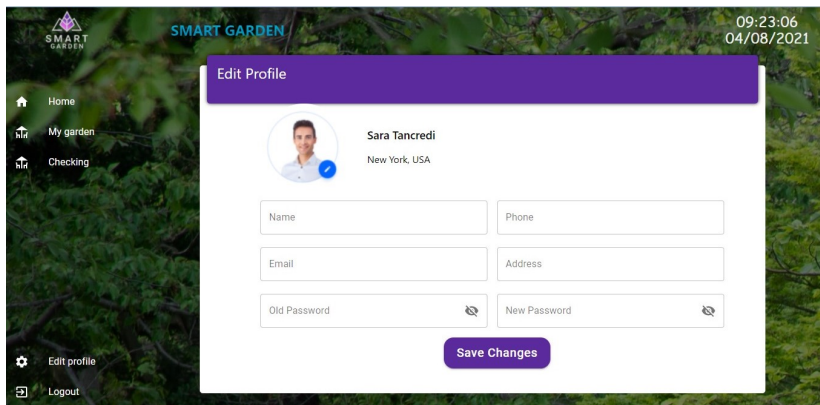


Figure 25: Cập nhập thông tin cá nhân

Ngoài các thiết bị trên, thì người dùng còn thiết bị dùng để điều khiển là Relay để điều khiển máy bơm nước. Giao diện sẽ gồm một bên chứa nút công tắc, và một bảng log cho biết những thời điểm đã sử dụng nút Relay.

Thêm một chức năng khác là kiểm tra lịch sử dữ liệu của thiết bị, user sẽ phải chọn field (soil, humidity, temperature,..), sau đó thì có thể tìm kiếm bất kì ngày, tháng, thời gian nào muốn tìm, dữ liệu sẽ hiển thị ở bảng bên dưới.

Chức năng cuối cùng của user là thay đổi thông tin cá nhân, ở đây người dùng có thể click vào ô nhập để biết được giá trị hiện tại là gì (riêng mật khẩu là bảo mật), sau đó có thể thay đổi thông tin cá nhân cho phù hợp.

6 Kết luận và hướng phát triển

6.1 Kết luận

Trong quá trình triển khai, nhóm em đã rút ra được những điều làm được và chưa làm được, cụ thể:

Làm được:

- Tổ chức làm việc nhóm đạt hiệu quả (gặp mặt trực tiếp và online bằng Google Meet).
- Báo cáo kết quả hoạt động hằng tuần với thầy

Yêu cầu		Tính năng hiện thực
Module 1	Nhận và hiển thị dữ liệu từ thiết bị	Use case 14: Pump water Use case 15: Measure soil humidity Use case 16 Measure temperature
Module 2	Kiểm tra kết quả input	Use case 5: Turn on/off water pump motor
Module 3	Điều khiển thiết bị	Use case 4: View temperature, soil humidity, air humidity in garden Use case 5: Turn on/off water pump motor
Module 4	Ghi nhận hoạt động	Use case 8: Checking information of garden Use case 9: View history turn on/off motor Use case 6 : Search/sort garden of user
Module 5	Tổ chức lưu trữ dữ liệu nghiệp vụ gồm dữ liệu các thiết bị và các kiểu thực thể khác trong hệ thống, lập trình web app và kết nối được web app client, iot server, business server và thiết bị	Hệ thống My Light IoT bao gồm: - Server MQTT - MySQL database - Trang web app

Chưa làm được:

- Use case 12: Kiểm tra dữ liệu nhận được vượt quá ngưỡng cho phép: Quy định mức giới hạn của độ ẩm, nhiệt độ
- Use case 13: Sửa đổi mức giới hạn của độ ẩm, nhiệt độ
- Use case 18: Send data

6.2 Hướng phát triển

- Hoàn thiện đủ tính năng
- Thay đổi, cải thiện những điểm chưa làm tốt : xây dựng UI đẹp, thu hút, nhiều hiệu ứng hơn.
- Tối ưu hóa, nâng cao hiệu suất.
- Hệ thống lại mã nguồn (source code)

7 Tài liệu tham khảo

References

- [1] *Những điều cần biết về giao thức MQTT*
URL: <https://techmaster.vn/posts/34394/iot-giao-thuc-mqtt-va-ung-dung-trong-iot>
Truy cập ngày 26/7/2021.
- [2] *Khái niệm cơ bản về IoT Gateway - Cấu trúc của IoT Gateway - Chức năng của một IoT Gateway.*
URL: <https://www.maytinh365.com/2019/07/IoT-Gateway-la-gi.html>
Truy cập ngày 26/7/2021.
- [3] *IOT Gateway là gì? Ý nghĩa và chức năng vận hành của thiết bị này?*
URL: <https://www.thegioimaychu.vn/blog/tong-hop/iot-gateway>
Truy cập ngày 25/7/2021.
- [4] *Bước đầu tìm hiểu về ReactJs*
URL: <https://viblo.asia/p/buoc-dau-tim-hieu-ve-reactjs-1VgZvwWYlAw>
Truy cập ngày 25/7/2021.
- [5] *Tìm hiểu về python cơ bản*
URL: <https://viblo.asia/p/tim-hieu-ve-python-co-ban-1-LzD5djkeZjY>
Truy cập ngày 27/7/2021.