

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN I



BÁO CÁO CUỐI KÌ IOT

Đề tài: Hệ thống tưới tiêu thông minh tự động
Nhóm 06 – Lớp 06

Giảng viên hướng dẫn

Kim Ngọc Bách

Thành viên nhóm

Mã sinh viên

Nguyễn Đức Đạt

B22DCCN194

Lưu Minh Hiễn

B22DCCN290

Nguyễn Đức Khởi

B22DCCN470

Vũ Hồng Linh

B22DCCN494

Hà Nội – 2025

Lời cảm ơn

Lời đầu tiên, nhóm chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy **TS. Kim Ngọc Bách** - giảng viên hướng dẫn môn IoT và Ứng dụng.

Những bài giảng tâm huyết và những lời khuyên, tư vấn quý báu của Thầy trên lớp đã giúp chúng em định hướng đúng đắn và giải quyết được nhiều vướng mắc trong quá trình thực hiện đề án “IoT Smart Garden” (Hệ thống tưới tiêu thông minh tự động). Kiến thức thực tiễn mà Thầy truyền đạt chính là nền tảng quan trọng để chúng em hoàn thiện sản phẩm ngày hôm nay.

Sự tâm huyết và kiến thức thực tiễn mà Thầy truyền đạt không chỉ giúp chúng em hoàn thành tốt đề án môn học này mà còn trang bị cho chúng em những hành trang vững chắc, tư duy làm việc chuyên nghiệp cho con đường nghề nghiệp sau này.

Mặc dù đã rất cố gắng, nhưng do kiến thức và kinh nghiệm còn hạn chế nên đề tài khó tránh khỏi những thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp, nhận xét từ Thầy để đề tài được hoàn thiện hơn nữa.

Một lần nữa, chúng em xin chân thành cảm ơn Thầy!

Người thực hiện đề tài

Nguyễn Đức Đạt
Lưu Minh Hiên
Nguyễn Đức Khởi
Vũ Hồng Linh

Mục lục

CHƯƠNG 1: TỔNG QUAN	5
1.1. Đặt vấn đề	5
1.2. Phạm vi triển khai	6
1.3. Tiêu chí của nhóm	7
1.4. Kết quả mong đợi	8
CHƯƠNG 2: THU THẬP YÊU CẦU TỪ CÁC BÊN LIÊN QUAN	9
2.1. Yêu cầu từ Người dùng thiết bị (Hộ gia đình)	9
2.2. Yêu cầu từ góc độ Quản lý & Đầu tư	10
2.3. Yêu cầu từ góc độ Kỹ thuật	10
2.4. Phương pháp Thu thập Yêu cầu	11
CHƯƠNG 3: PHÂN TÍCH YÊU CẦU CHỨC NĂNG	12
3.1. Mô tả tổng quát các chức năng cơ bản và luồng hoạt động của hệ thống	12
3.2. Phân tích và Đặc tả Use-case	13
3.2.1. Các tác nhân (Actors) trong hệ thống	14
3.2.2. Mô tả chi tiết các Use-case	14
3.3. Sơ đồ tuần tự (Sequence diagram)	16
3.3.1. Sơ đồ tuần tự cho chức năng đăng nhập	16
3.3.2. Sơ đồ tuần tự cho chức năng bật bơm từ xa	16
3.3.3. Sơ đồ tuần tự cho chức năng tự động bật bơm tự động	17
3.3.4. Sơ đồ tuần tự cho chức năng cập nhật firmware từ xa	17
3.3.5. Sơ đồ tuần tự cho chức năng gửi dữ liệu cảm biến theo định kỳ	18
CHƯƠNG 4: CÁC YÊU CẦU PHI CHỨC NĂNG	19
4.1. Hiệu năng (Performance)	19
4.2. Bảo mật (Security)	19

4.3. Độ tin cậy (Reliability).....	20
4.4. Khả năng mở rộng (Scalability)	21
4.5. Chi phí và Năng lượng (Cost & Power).....	21
CHƯƠNG 5: CÔNG NGHỆ SỬ DỤNG.....	22
5.1. Phần cứng	22
5.1.1. ESP32	22
5.1.2. Cảm biến đo độ ẩm đất (FC28)	25
5.1.3. Cảm biến đo độ ẩm không khí (DHT11)	26
5.1.4. Module Relay Shield 5V (loại 1 kênh)	27
5.2. Phần mềm	29
CHƯƠNG 6: THIẾT KẾ HỆ THỐNG.....	32
6.1. Sơ đồ khối hệ thống	32
6.2. Sơ đồ nguyên lý	34
6.3. Thiết kế cơ sở dữ liệu.....	35
6.3.1. Mô hình Dữ liệu Tổng quan (Schema Overview).....	35
6.3.2. Chi tiết các Collection.....	35
6.3.3. Mối quan hệ giữa các Collection (Entity Relationship)	39
TÀI LIỆU THAM KHẢO	40

CHƯƠNG 1: TỔNG QUAN

1.1. Đặt vấn đề

Trong kỷ nguyên Cách mạng Công nghiệp 4.0, sự bùng nổ của Internet vạn vật (IoT - Internet of Things) đang định hình lại cách thức vận hành của mọi lĩnh vực, từ công nghiệp, y tế đến đời sống hàng ngày. Nông nghiệp – ngành kinh tế được coi là "trụ đỡ" của Việt Nam – cũng không nằm ngoài xu thế đó. Việc chuyển đổi từ phương thức canh tác truyền thống sang nền nông nghiệp thông minh (Smart Agriculture) đang trở thành một yêu cầu cấp thiết để nâng cao năng lực cạnh tranh và phát triển bền vững.

Thực tế cho thấy, nền nông nghiệp Việt Nam hiện nay vẫn đang đối mặt với nhiều thách thức lớn và ngày càng gay gắt:

- **Biến đổi khí hậu:** Tình trạng hạn hán, xâm nhập mặn diễn biến phức tạp khiến nguồn nước ngọt trở nên khan hiếm, đòi hỏi việc sử dụng nước tưới phải cực kỳ tiết kiệm và hiệu quả.
- **Lãng phí tài nguyên:** Phương pháp tưới tiêu thủ công truyền thống thường dựa vào kinh nghiệm cảm tính, dẫn đến tình trạng tưới thừa hoặc thiếu nước, gây lãng phí điện năng, nguồn nước và ảnh hưởng đến sức khỏe cây trồng.
- **Thiếu hụt nhân lực:** Quá trình đô thị hóa nhanh chóng khiến nguồn lao động nông nghiệp ngày càng giảm sút, trong khi chi phí nhân công lại tăng cao.

Xuất phát từ những yêu cầu thực tiễn đó, nhóm chúng tôi đã nghiên cứu và phát triển đề tài “**Hệ thống IoT tưới tiêu thông minh**”.

Đây không chỉ đơn thuần là một hệ thống tưới tự động hẹn giờ (Timer) thông thường, mà là một giải pháp công nghệ toàn diện dựa trên dữ liệu thực tế. Hệ thống sử dụng mạng lưới các cảm biến môi trường (độ ẩm đất, nhiệt độ, độ ẩm không khí) để liên tục thu thập dữ liệu tại thời gian thực. Các dữ liệu này được xử lý bởi vi điều khiển trung tâm để đưa ra quyết định tưới tiêu chính xác đến từng thời điểm, đảm bảo cây trồng luôn nhận được lượng nước vừa đủ theo nhu cầu sinh trưởng.

Hơn nữa, hệ thống còn tích hợp khả năng kết nối Internet, cho phép người nông dân giám sát toàn bộ khu vườn và điều khiển máy bơm từ xa thông qua ứng dụng trên điện thoại thông minh (Mobile App), bất kể họ đang ở đâu. Giải pháp này hướng tới mục tiêu "kép": vừa giải phóng sức lao động cho con người, vừa tối ưu hóa chi phí sản xuất, góp phần từng bước hiện đại hóa nền nông nghiệp nước nhà theo hướng công nghệ cao và bền vững.

1.2. Phạm vi triển khai

Mục tiêu của nhóm là triển khai hệ thống tưới tiêu thông minh trong phạm vi khu vườn của một hộ gia đình, với diện tích bề mặt $\leq 40m^2$. Hệ thống được thiết kế tối ưu cho môi trường đô thị hoặc nhà phố, tập trung vào tính an toàn điện và sự tiện lợi khi vận hành.

a) Số lượng và Cấu hình Thiết bị Hệ thống được tối giản hóa để đảm bảo hiệu quả kinh tế nhưng vẫn đáp ứng đầy đủ các tính năng thông minh:

- **Cảm biến (Sensor):**
 - Sử dụng **01 Cảm biến độ ẩm đất (FC28)**: Loại cảm biến điện dung chống ăn mòn, giúp đo độ ẩm chính xác và bền bỉ hơn trong môi trường đất ẩm ướt lâu dài.
 - Sử dụng **01 Cảm biến nhiệt độ & độ ẩm không khí (DHT11)**: Để thu thập thông tin thời tiết bổ trợ.
 - Vị trí lắp đặt: Cắm tại các điểm đại diện (trung tâm hoặc khu vực trồng cây nhạy cảm) của khu vườn.
- **Cơ cấu chấp hành (Actuator):**
 - Sử dụng **01 Máy bơm nước mini 12V DC**: Thay thế hoàn toàn máy bơm 220V truyền thống để đảm bảo an toàn tuyệt đối về điện cho người sử dụng.
 - Sử dụng **01 Module Relay 5V**: Đóng vai trò công tắc trung gian, nhận lệnh từ vi điều khiển để đóng ngắt nguồn 12V cho máy bơm.
 - Nguồn cấp nước: Bơm sẽ hút nước từ bể chứa/téc nước có sẵn của gia đình hoặc đầu nối từ vòi nước chờ.
- **Bộ điều khiển trung tâm (Gateway/Controller):**
 - Sử dụng **01 Vi điều khiển ESP32 DevKit V1**: Đây là "bộ não" của hệ thống, tích hợp sẵn Wi-Fi để kết nối Internet. ESP32 chịu trách nhiệm thu thập dữ liệu từ cảm biến, xử lý logic tự động và nhận lệnh điều khiển từ xa.

b) Quy mô mạng

- **Cục bộ:** Hệ thống hoạt động trong phạm vi phủ sóng Wi-Fi của hộ gia đình.
- **Công nghệ kết nối:** Sử dụng mạng Wi-Fi gia đình (băng tần 2.4GHz) làm hạ tầng truyền dẫn chính. Toàn bộ dữ liệu và lệnh điều khiển sẽ được truyền tải qua giao thức **MQTT** (Message Queuing Telemetry Transport) để đảm bảo tốc độ nhanh và độ tin cậy cao.

c) Môi trường hoạt động

- **Nông nghiệp quy mô nhỏ (Sân vườn/Ban công):** Các thiết bị phần cứng (Cảm biến, Máy bơm) sẽ tiếp xúc trực tiếp với môi trường ngoài trời (nắng, mưa, độ ẩm cao).
 - **Giải pháp bảo vệ:** Bộ điều khiển trung tâm (ESP32 + Relay) sẽ được đặt trong hộp kỹ thuật chống nước (IP65). Các mối nối dây

điện được bọc kín hoặc sử dụng đầu nối chuyên dụng để chống oxy hóa.

- **An toàn thông tin:** Hệ thống đảm bảo các tiêu chuẩn bảo mật cơ bản cho thiết bị IoT gia đình, bao gồm xác thực kết nối Wi-Fi (WPA2) và mã hóa dữ liệu truyền tải.

1.3. Tiêu chí của nhóm

Để đánh giá một cách khách quan xem hệ thống tưới cây thông minh có hoạt động tốt và đạt được mục tiêu đề ra hay không, nhóm chúng em xây dựng các chỉ số hiệu suất chính (KPIs) có thể định lượng được. Các KPIs này giúp đo lường và kiểm chứng mức độ thành công của dự án. Các KPIs cụ thể cho hệ thống này bao gồm:

a) Độ chính xác (Accuracy):

- **Mục tiêu:** Đảm bảo dữ liệu từ cảm biến phản ánh đúng tình trạng của đất.
- **Chỉ số đo lường:** Sai số của giá trị độ ẩm đất mà cảm biến đo được không được vượt quá $\pm 5\%$ so với kết quả đo bằng thiết bị chuyên dụng.

b) Độ trễ (Latency):

- **Mục tiêu:** Hệ thống phải có khả năng phản hồi nhanh chóng với các lệnh từ người dùng.
- **Chỉ số đo lường:** Thời gian từ khi người dùng nhấn nút bật/tắt trên ứng dụng di động đến khi van nước thực sự thay đổi trạng thái (mở hoặc đóng) phải dưới 5 giây.

c) Độ tin cậy (Reliability):

- **Mục tiêu:** Hệ thống phải hoạt động ổn định và liên tục trong điều kiện thực tế.
- **Chỉ số đo lường:**
 - Tỷ lệ thời gian hoạt động (uptime) của hệ thống phải đạt trên 98% trong suốt quá trình thử nghiệm.
 - Hệ thống phải có khả năng tự động kết nối lại mạng Wi-Fi trong vòng 1 phút khi có sự cố mất kết nối tạm thời.

d) Chi phí (Cost):

- **Mục tiêu:** Xây dựng một hệ thống hiệu quả với chi phí hợp lý, phù hợp cho quy mô hộ gia đình.
- **Chỉ số đo lường:**
 - **Chi phí đầu tư (CAPEX):** Tổng chi phí để mua sắm toàn bộ linh kiện và vật tư phụ không vượt quá ngân sách đề ra là 1,500,000 VNĐ.
 - **Chi phí vận hành (OPEX):** Chi phí vận hành bao gồm tiền điện duy trì hệ thống IOT, tiền nước tưới tiêu và chi phí bảo hành nếu

có hỏng hóc.

e) Khả năng mở rộng và tiết kiệm tài nguyên:

- Hệ thống phải hỗ trợ nhân đôi số lượng cảm biến trong 3 năm tới mà không cần thay đổi hạ tầng mạng.
- Giảm ít nhất 30-50% lượng nước tưới so với phương pháp truyền thống.

1.4. Kết quả mong đợi

Dự án hướng tới việc hoàn thiện một sản phẩm IoT ứng dụng thực tế với các kết quả cụ thể sau:

Về mặt Hệ thống & Kỹ thuật:

- Xây dựng thành công hệ thống phần cứng hoạt động ổn định, bền bỉ trong môi trường ngoài trời, đảm bảo an toàn điện tuyệt đối (sử dụng nguồn 12V).
- Thiết lập kênh giao tiếp dữ liệu liên tục, tin cậy giữa thiết bị và máy chủ thông qua giao thức MQTT, với độ trễ thấp (< 5 giây).
- Phát triển hoàn thiện ứng dụng điều khiển (Web/Mobile App) với giao diện trực quan, hiển thị thông số thời gian thực và vẽ biểu đồ lịch sử chính xác.

Về mặt Ứng dụng & Hiệu quả:

- **Tự động hóa thông minh:** Hệ thống có khả năng tự ra quyết định tưới tiêu chính xác dựa trên ngưỡng độ ẩm đất thực tế, thay thế hoàn toàn thao tác thủ công.
- **Tiết kiệm tài nguyên:** Giảm thiểu lượng nước tưới dư thừa và tiết kiệm thời gian, công sức chăm sóc cây trồng cho hộ gia đình.
- **Giám sát mọi lúc mọi nơi:** Người dùng có thể theo dõi tình trạng khu vườn và can thiệp điều khiển kịp thời từ xa thông qua smartphone, ngay cả khi không có mặt ở nhà.
- **Hỗ trợ ra quyết định:** Dữ liệu lịch sử về độ ẩm và nhiệt độ được lưu trữ sẽ là cơ sở quan trọng để người dùng điều chỉnh chế độ chăm sóc phù hợp nhất cho từng loại cây trồng.

CHƯƠNG 2: THU THẬP YÊU CẦU TỪ CÁC BÊN LIÊN QUAN

Việc xác định rõ ràng và đầy đủ các yêu cầu từ các bên liên quan là bước quan trọng nhất để đảm bảo sự thành công của dự án. Nhóm phát triển chúng em đã tiến hành khảo sát và phân tích yêu cầu từ ba góc độ chính: Người sử dụng cuối, Nhà đầu tư/Quản lý và Yêu cầu Kỹ thuật.

2.1. Yêu cầu từ Người dùng thiết bị (Hộ gia đình)

Đối tượng: Chủ nhà và các thành viên trong gia đình. Đây là những người trực tiếp vận hành và thụ hưởng lợi ích từ hệ thống, thường không yêu cầu kiến thức chuyên sâu về kỹ thuật.

Các yêu cầu cụ thể:

- **Giao diện Giám sát Trực quan:**
 - Ứng dụng di động/Web cần có giao diện thân thiện (User-friendly Dashboard).
 - Hiển thị các thông số môi trường (Độ ẩm đất, Nhiệt độ, Độ ẩm không khí) dưới dạng biểu đồ trực quan theo thời gian thực (Real-time chart), cập nhật liên tục mà không cần tải lại trang.
 - Hiển thị rõ ràng trạng thái hoạt động hiện tại của máy bơm (ĐANG TƯỚI / ĐÃ TẮT).
- **Khả năng Điều khiển Linh hoạt:**
 - Cung cấp nút bấm "Một chạm" trên ứng dụng để Bật/Tắt máy bơm ngay lập tức trong các tình huống khẩn cấp hoặc theo nhu cầu đột xuất.
 - Phản hồi trạng thái điều khiển nhanh chóng (< 3 giây) để người dùng biết lệnh đã được thực thi.
- **Tính Tự động hóa (Automation):**
 - Hệ thống phải hoạt động theo cơ chế "Set and Forget" (Cài đặt một lần và tự chạy).
 - Tự động tưới khi độ ẩm đất xuống thấp hơn ngưỡng cài đặt và tự động ngắt khi đất đủ ẩm, đảm bảo cây trồng luôn được chăm sóc đúng lúc.
- **Hệ thống Cảnh báo (Notification):**
 - Gửi thông báo đẩy (Push Notification) về điện thoại khi có sự kiện quan trọng: Máy bơm bật/tắt, độ ẩm quá thấp/quá cao, hoặc thiết bị mất kết nối mạng.

2.2. Yêu cầu từ góc độ Quản lý & Đầu tư

Đối tượng: Chủ hộ gia đình, đóng vai trò là người ra quyết định đầu tư và quản lý chi phí.

Các yêu cầu cụ thể:

- **Hiệu quả Đầu tư (ROI):**
 - Chi phí phần cứng (CAPEX) phải được tối ưu hóa, nằm trong ngân sách dự kiến (< 1,500,000 VNĐ) để phù hợp với quy mô hộ gia đình.
 - Giá trị mang lại phải rõ ràng: Tiết kiệm hóa đơn nước hàng tháng, giảm thời gian chăm sóc, và quan trọng nhất là tăng tỷ lệ sống và năng suất của cây trồng.
- **Độ tin cậy và Chi phí Vận hành (OPEX):**
 - Hệ thống phải hoạt động bền bỉ 24/7 ngoài trời, chịu được nắng mưa.
 - Tiêu thụ điện năng thấp.
 - Dễ dàng bảo trì, thay thế linh kiện (như cảm biến, máy bơm) với chi phí rẻ khi có hỏng hóc.
- **Khả năng Mở rộng (Scalability):**
 - Kiến trúc hệ thống phải cho phép dễ dàng thêm mới thiết bị (ví dụ: mở rộng thêm một khu vực trồng cây mới) mà không cần đập đi xây lại toàn bộ phần mềm hay phần cứng.

2.3. Yêu cầu từ góc độ Kỹ thuật

Đối tượng: Nhóm phát triển dự án.

Các yêu cầu cụ thể:

- **Hạ tầng Kết nối:**
 - Tận dụng mạng Wi-Fi gia đình sẵn có (băng tần 2.4GHz) để kết nối thiết bị, không phát sinh thêm chi phí hạ tầng mạng riêng.
 - Đảm bảo khả năng tự động kết nối lại (Auto-reconnect) khi mạng Wi-Fi chập chờn.
- **Giao thức Truyền thông:**
 - Sử dụng giao thức **MQTT** (Message Queuing Telemetry Transport) cho giao tiếp giữa thiết bị (ESP32) và Server để đảm bảo tốc độ cao, độ trễ thấp và tiết kiệm băng thông.
 - Sử dụng **HTTP/Socket.IO** cho giao tiếp giữa Server và Ứng dụng người dùng để đảm bảo tính tương tác thời gian thực.
- **An toàn và Bảo mật:**

- Dữ liệu truyền tải phải được đóng gói theo chuẩn JSON.
- Thiết bị phải có cơ chế xác thực cơ bản để tránh việc bị điều khiển bởi người lạ trong cùng mạng Wi-Fi.
- Hệ thống điện (Máy bơm) phải sử dụng nguồn điện áp thấp (12V DC) để đảm bảo an toàn tuyệt đối cho người sử dụng, tránh nguy cơ rò rỉ điện lưới 220V.

2.4. Phương pháp Thu thập Yêu cầu

Để đảm bảo tính thực tiễn của các yêu cầu trên, nhóm đã áp dụng các phương pháp:

- **Phỏng vấn sâu:** Trao đổi trực tiếp với các thành viên gia đình để hiểu rõ thói quen chăm sóc cây và những khó khăn hiện tại (quên tưới, tưới quá nhiều nước...).
- **Khảo sát hiện trường:** Đo đạc thực tế tại khu vực vườn để xác định vị trí lắp đặt tối ưu cho cảm biến (tránh nơi ngập úng cục bộ), vị trí nguồn nước và nguồn điện, từ đó đưa ra giải pháp thi công dây dẫn an toàn và thẩm mỹ.

CHƯƠNG 3: PHÂN TÍCH YÊU CẦU CHỨC NĂNG

3.1. Mô tả tổng quát các chức năng cơ bản và luồng hoạt động của hệ thống

Hệ thống tưới cây thông minh hoạt động dựa trên một luồng khép kín, kết hợp giữa việc thu thập dữ liệu từ môi trường thực tế và khả năng điều khiển từ xa của người dùng. Luồng hoạt động của hệ thống có thể được mô tả qua các bước chính như sau:

Bước 1: Thu thập dữ liệu độ ẩm từ Cảm biến

- **Cảm biến độ ẩm đất và cảm biến độ ẩm không khí** sẽ được lắp đặt để theo dõi trạng thái của cây trồng. Cảm biến sẽ liên tục đo độ ẩm và chuyển đổi thành tín hiệu điện (dưới dạng điện trở hoặc điện áp).

Bước 2: Xử lý và Gửi dữ liệu tại Vi điều khiển ESP

- Tín hiệu điện từ cảm biến được gửi đến **vi điều khiển ESP**.
- Mạch ESP sẽ đọc và chuyển đổi tín hiệu này thành một giá trị số cụ thể mà hệ thống có thể hiểu được (ví dụ: phần trăm độ ẩm).
- Sau đó, ESP sẽ đóng gói dữ liệu này và gửi lên **Cloud Server** thông qua mạng Wi-Fi của gia đình, sử dụng các giao thức mạng phổ biến như MQTT hoặc HTTP.

Bước 3: Xử lý, Lưu trữ và Phản hồi từ Cloud Server

- **Cloud Server** nhận dữ liệu từ ESP gửi lên.
- Dữ liệu này ngay lập tức được xử lý và lưu trữ vào **Cơ sở dữ liệu (Database)** để ghi nhận lịch sử và phục vụ cho việc phân tích sau này.
- Đồng thời, Cloud Server cung cấp các giao diện lập trình ứng dụng (API) hoặc Socket để **Ứng dụng di động (UI)** có thể truy vấn và lấy dữ liệu này về.

Bước 4: Hiển thị và Tương tác trên Giao diện người dùng (UI)

- Ứng dụng di động trên điện thoại của người dùng sẽ gọi tới API của Cloud Server để lấy dữ liệu mới nhất.
- Dữ liệu (độ ẩm, trạng thái máy bơm, v.v.) sẽ được **hiển thị một cách trực quan** trên giao diện ứng dụng.
- Người dùng có thể theo dõi các thông số và tương tác với hệ thống bằng cách nhấn các nút điều khiển (ví dụ: nút "Bật/Tắt máy bơm thủ công").

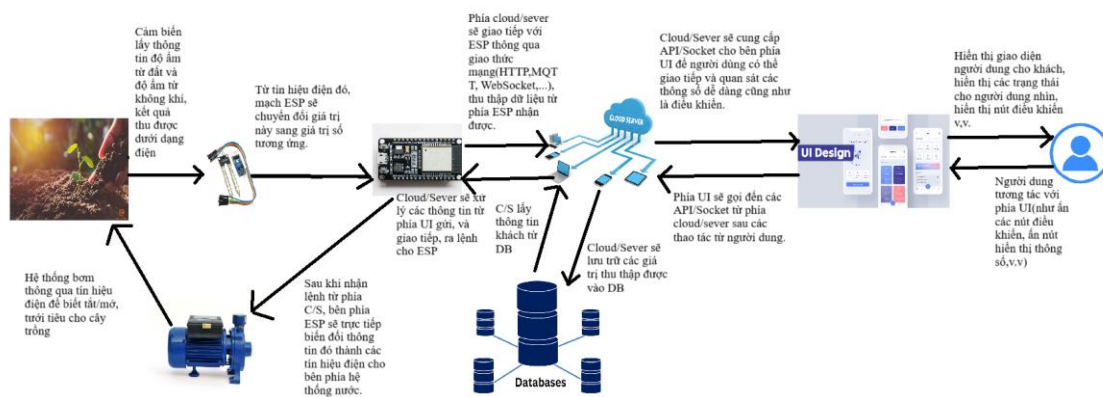
Bước 5: Luồng điều khiển ngược từ Người dùng đến Thiết bị

- Khi người dùng thực hiện một thao tác trên ứng dụng (ví dụ: nhấn nút "Bật"), một lệnh điều khiển sẽ được gửi từ ứng dụng đến **Cloud Server**.

- Cloud Server xác thực và xử lý lệnh này, sau đó gửi một thông điệp điều khiển xuống cho **vi điều khiển ESP** tương ứng.

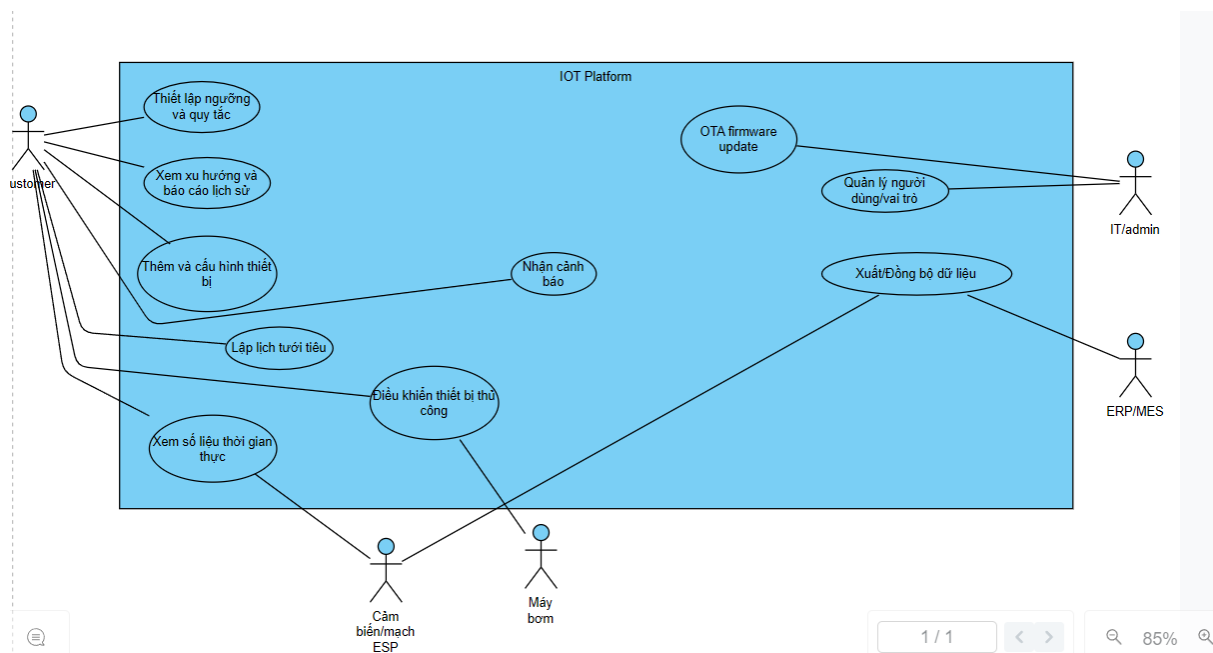
Bước 6: Thực thi lệnh và Hoàn thành chu trình

- Vi điều khiển ESP** nhận lệnh từ Cloud Server.
- ESP sẽ xử lý lệnh này và xuất ra tín hiệu điện để điều khiển **hệ thống bơm** (thường thông qua một mạch relay).
- Máy bơm sẽ được bật/tắt theo đúng lệnh, thực hiện việc tưới tiêu cho cây trồng.
- Quá trình này làm thay đổi độ ẩm của đất, và cảm biến sẽ lại ghi nhận giá trị mới, bắt đầu lại một chu trình giám sát mới.



Hình 3.1.1: Mô tả sơ lược về luồng hoạt động của hệ thống

3.2. Phân tích và Đặc tả Use-case



Hình 3.2: Sơ đồ mô tả Use-case

3.2.1. Các tác nhân (Actors) trong hệ thống

- **Customer (Khách hàng/Chủ nhà):** Đây là người dùng chính và quan trọng nhất của hệ thống. **Customer** là người sở hữu khu vườn, có nhu cầu giám sát và tự động hóa việc tưới tiêu. Họ tương tác với hệ thống chủ yếu thông qua giao diện ứng dụng di động để xem thông tin, điều khiển thủ công và thiết lập các quy tắc tự động.
- **IT/admin (Quản trị viên):** Đây là người dùng có vai trò kỹ thuật, chịu trách nhiệm quản lý và bảo trì hệ thống. **IT/admin** có các quyền hạn nâng cao như cập nhật phần mềm cho thiết bị từ xa (OTA firmware update) và quản lý tài khoản người dùng. Trong bối cảnh dự án hộ gia đình, vai trò này và vai trò Customer có thể là cùng một người, nhưng được tách ra để thể hiện sự phân chia về chức năng quản trị.
- **ERP/MES (Hệ thống Doanh nghiệp):** Đây là một tác nhân hệ thống (system actor), đại diện cho một phần mềm bên ngoài (ví dụ: một bảng tính Google Sheets, một hệ thống quản lý nhà thông minh khác). Vai trò của **ERP/MES** là truy cập vào nền tảng IoT thông qua API để **xuất và đồng bộ dữ liệu**, phục vụ cho các mục đích phân tích hoặc lưu trữ lớn hơn.
- **Tác nhân phụ:**
 - **Cảm biến/mạch ESP (Sensor/ESP board):** Đây là tác nhân phần cứng, đóng vai trò là "giác quan" của hệ thống. **Cảm biến** có nhiệm vụ đo đạc các thông số thực tế từ môi trường (cụ thể là độ ẩm đất) và gửi dữ liệu đó về cho nền tảng IoT xử lý. Nó là nguồn cung cấp dữ liệu đầu vào cho các chức năng như xem số liệu và tự động hóa.
 - **Máy bơm (Pump):** Đây là tác nhân phần cứng, đóng vai trò là "cơ bắp" của hệ thống. **Máy bơm** nhận lệnh trực tiếp từ hệ thống (do người dùng điều khiển hoặc do quy tắc tự động kích hoạt) và thực hiện hành động vật lý là bơm nước để tưới cây. Nó là đối tượng thực thi đầu ra của hệ thống.

3.2.2. Mô tả chi tiết các Use-case

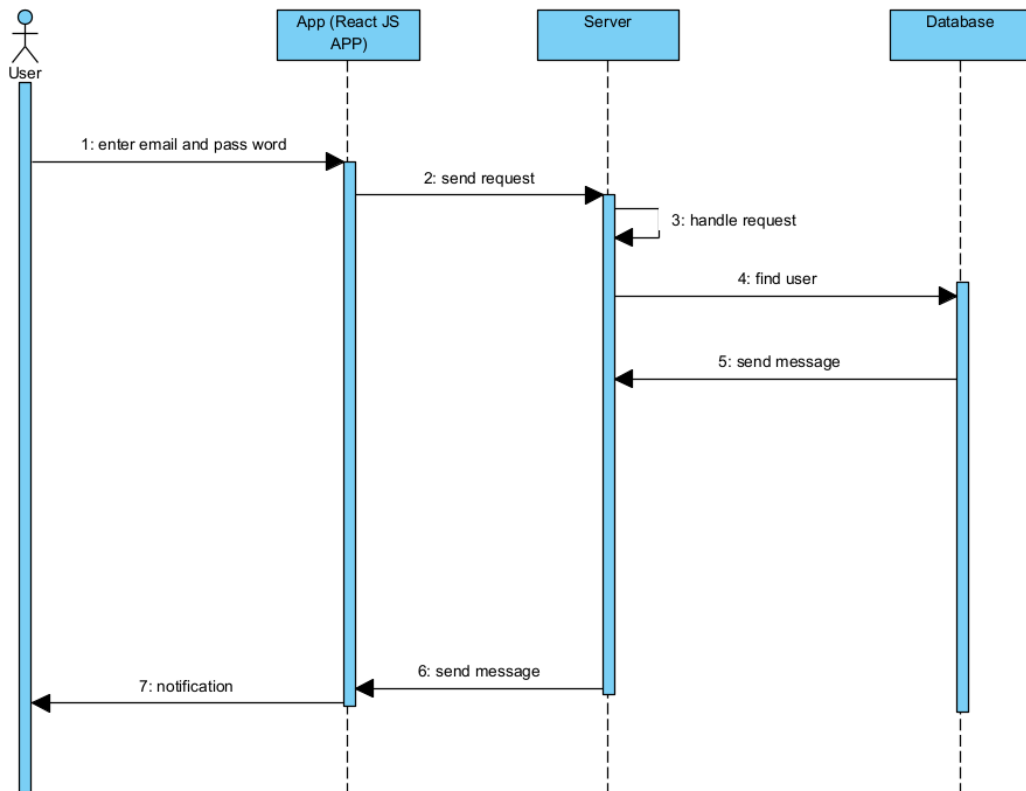
- **Xem số liệu thời gian thực:** Cho phép khách hàng xem ngay lập tức các thông số hiện tại của khu vườn, chẳng hạn như độ ẩm đất và trạng thái bật/tắt của máy bơm, thông qua giao diện ứng dụng.
- **Xem xu hướng và báo cáo lịch sử:** Cung cấp cho khách hàng các biểu đồ hoặc báo cáo về dữ liệu đã được thu thập trong quá khứ (ví dụ: độ ẩm

đất trong 7 ngày qua, lịch sử các lần tưới). Chức năng này giúp người dùng hiểu rõ hơn về nhu cầu nước của khu vườn.

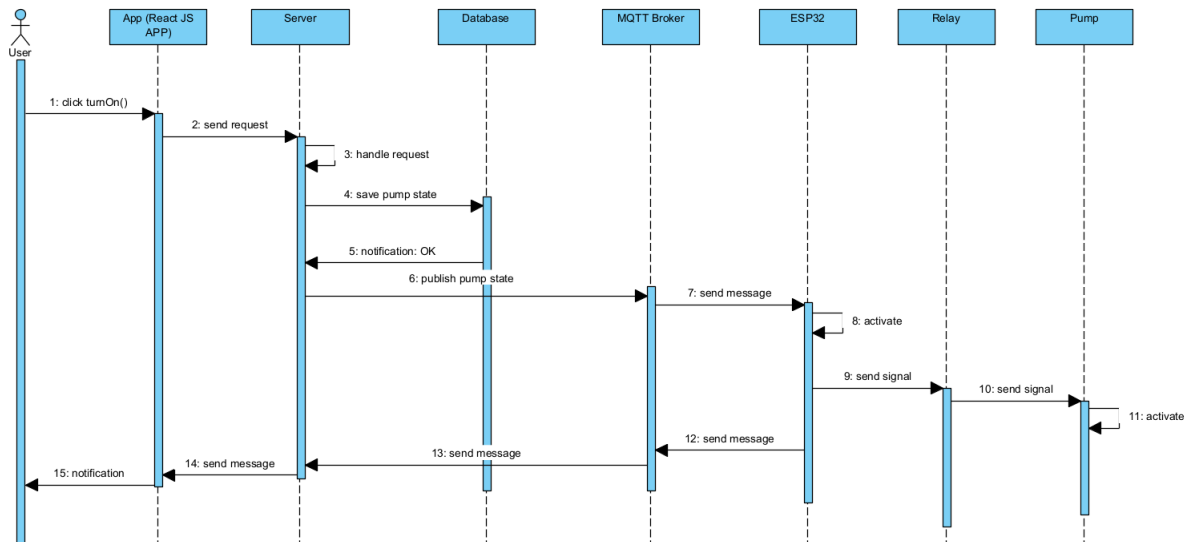
- **Điều khiển thiết bị thủ công:** Cho phép khách hàng bật hoặc tắt máy bơm nước ngay lập tức từ xa chỉ bằng một nút nhấn trên ứng dụng, bỏ qua chế độ tự động khi cần thiết.
- **Thiết lập ngưỡng và quy tắc:** Đây là chức năng cốt lõi của chế độ tự động. Nó cho phép khách hàng cài đặt các quy tắc logic cho hệ thống, ví dụ: "Nếu độ ẩm đất xuống dưới 40%, hãy tự động tưới trong 5 phút".
- **Lập lịch tưới tiêu:** Cho phép khách hàng thiết lập một lịch tưới cố định theo thời gian (ví dụ: tưới vào lúc 6 giờ sáng và 5 giờ chiều mỗi ngày).
- **Thêm và cấu hình thiết bị:** Cung cấp khả năng cho khách hàng tự thêm một thiết bị (cảm biến, máy bơm) mới vào hệ thống và cấu hình các thông số ban đầu cho nó.
- **Nhận cảnh báo:** Hệ thống sẽ tự động gửi thông báo (notification) đến điện thoại của khách hàng khi có sự kiện quan trọng xảy ra, chẳng hạn như độ ẩm đất quá thấp, máy bơm gặp sự cố, hoặc thiết bị mất kết nối.
- **Quản lý người dùng/vai trò:** Cho phép quản trị viên tạo, xóa hoặc chỉnh sửa tài khoản người dùng và phân quyền truy cập cho họ vào hệ thống.
- **OTA firmware update (Cập nhật phần mềm từ xa):** Cung cấp khả năng cho quản trị viên nâng cấp phiên bản phần mềm cho các thiết bị ESP từ xa thông qua mạng internet mà không cần phải can thiệp vật lý.
- **Xuất/Đồng bộ dữ liệu:** Cung cấp một giao diện lập trình ứng dụng (API) để các hệ thống phần mềm bên ngoài có thể tự động truy vấn và lấy dữ liệu (ví dụ: dữ liệu độ ẩm, lịch sử tưới) từ nền tảng IoT để phục vụ cho các mục đích phân tích hoặc lưu trữ riêng.

3.3. Sơ đồ tuần tự (Sequence diagram)

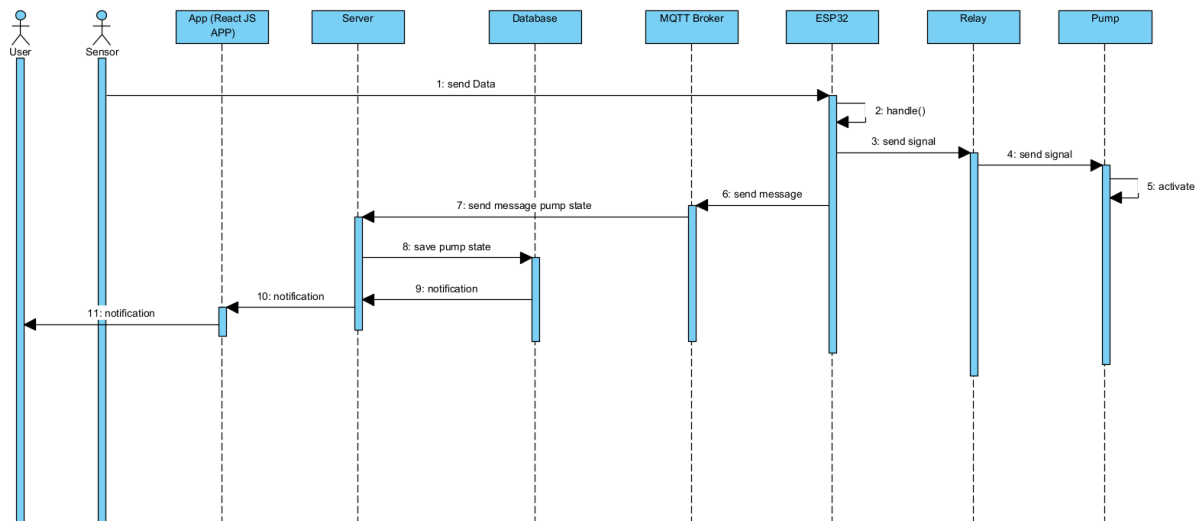
3.3.1. Sơ đồ tuần tự cho chức năng đăng nhập



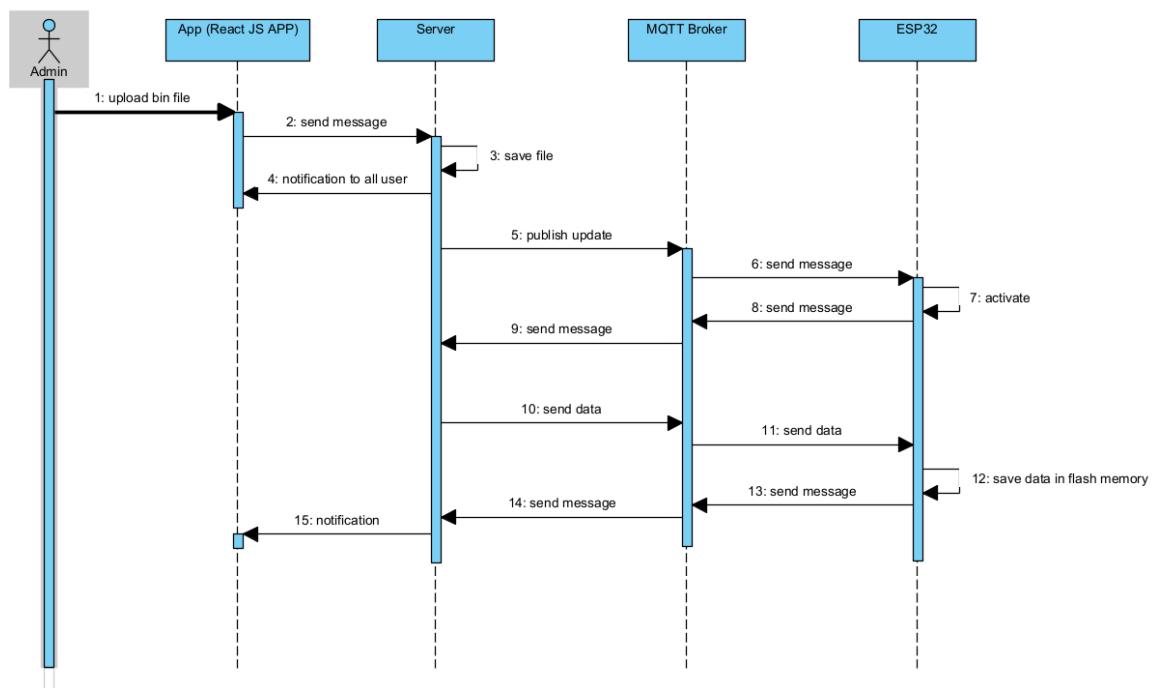
3.3.2. Sơ đồ tuần tự cho chức năng bật bơm từ xa



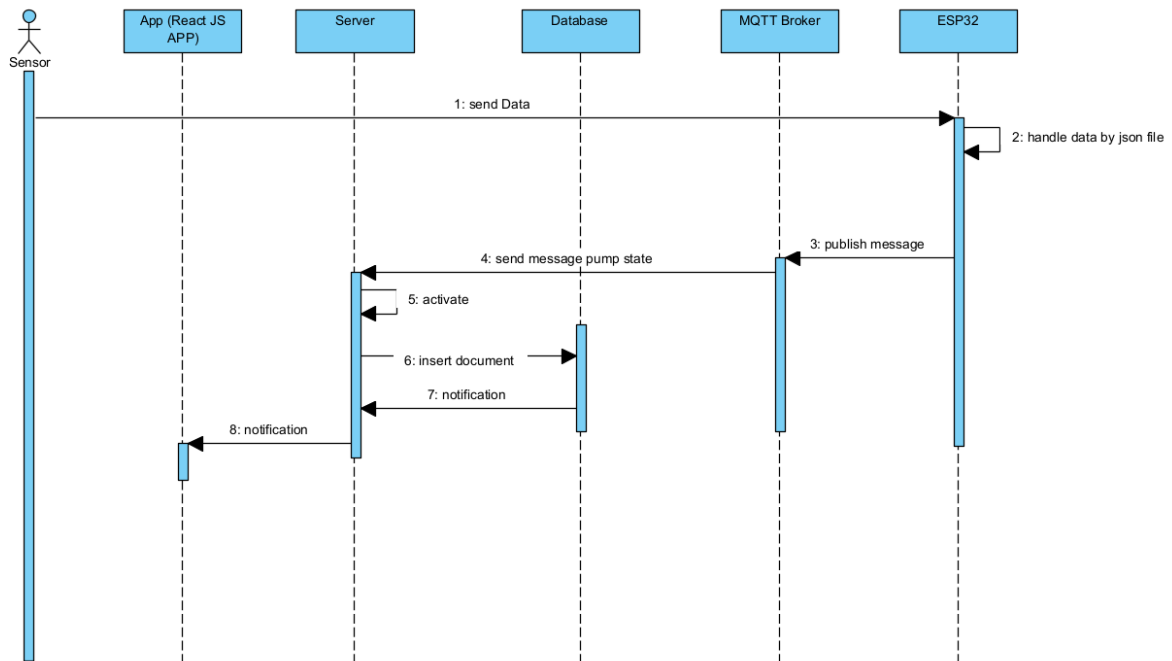
3.3.3. Sơ đồ tuần tự cho chức năng tự động bật bơm tự động



3.3.4. Sơ đồ tuần tự cho chức năng cập nhật firmware từ xa



3.3.5. Sơ đồ tuần tự cho chức năng gửi dữ liệu cảm biến theo định kỳ



CHƯƠNG 4: CÁC YÊU CẦU PHI CHỨC NĂNG

Phần này đặc tả các ràng buộc về chất lượng, hiệu năng, độ tin cậy và các yếu tố khác để đảm bảo hệ thống tưới cây thông minh hoạt động ổn định và hiệu quả trong thực tế. Các yêu cầu này đóng vai trò quyết định đến trải nghiệm người dùng và sự bền vững của hệ thống.

4.1. Hiệu năng (Performance)

Hiệu năng của hệ thống đo lường tốc độ và khả năng đáp ứng của các chức năng, ảnh hưởng trực tiếp đến cảm nhận của người dùng.

- **Độ trễ điều khiển (Control Latency):**

- **Yêu cầu:** Thời gian từ lúc người dùng nhấn nút "Bật/Tắt" thủ công trên ứng dụng đến khi máy bơm thực sự thay đổi trạng thái phải **dưới 3 giây**.
- **Diễn giải:** Một độ trễ thấp là cực kỳ quan trọng để mang lại trải nghiệm tương tác tức thì. Nếu thời gian chờ quá lâu, người dùng sẽ có cảm giác hệ thống bị "treo" hoặc không nhận lệnh. Họ có thể nhấn nút nhiều lần, dẫn đến việc gửi đi các lệnh chồng chéo và gây ra hoạt động không mong muốn. Phản hồi nhanh chóng khẳng định với người dùng rằng hệ thống đang hoạt động ổn định và nghe theo lệnh của họ.

- **Tần suất cập nhật dữ liệu (Data Update Frequency):**

- **Yêu cầu:** Dữ liệu độ ẩm từ cảm biến phải được cập nhật lên hệ thống và hiển thị trên ứng dụng với tần suất không ít hơn **15 phút một lần**.
- **Diễn giải:** Tần suất này là một sự cân bằng hợp lý giữa việc cung cấp thông tin kịp thời và việc tối ưu tài nguyên. Cập nhật quá thường xuyên (ví dụ: mỗi giây) sẽ gây tốn pin không cần thiết cho thiết bị, tạo ra lưu lượng mạng lớn và làm đầy cơ sở dữ liệu một cách nhanh chóng. Ngược lại, cập nhật quá thưa (ví dụ: mỗi giờ) có thể khiến người dùng không nắm bắt được những thay đổi đột ngột của điều kiện đất, làm giảm tính hiệu quả của việc giám sát.

4.2. Bảo mật (Security)

Dù là một hệ thống tại nhà, các yếu tố bảo mật cơ bản là bắt buộc để bảo vệ dữ liệu và sự riêng tư của người dùng, cũng như an toàn cho mạng gia đình.

- **Mã hóa kết nối (Connection Encryption):**

- **Yêu cầu:** Toàn bộ dữ liệu trao đổi giữa thiết bị ESP và Nền tảng IoT (Cloud) phải được mã hóa bằng giao thức **TLS/SSL**. Kết nối Wi-Fi tại nhà phải sử dụng chuẩn bảo mật **WPA2** trở lên.
- **Diễn giải:** Việc mã hóa đảm bảo rằng dữ liệu (ví dụ: thông tin độ ẩm, lệnh điều khiển) không thể bị "đọc trộm" khi đang được truyền trên mạng. Điều này ngăn chặn nguy cơ kẻ xấu nghe lén thông tin hoặc, nghiêm trọng hơn, lợi dụng lỗ hổng để tấn công vào các thiết bị khác trong cùng mạng Wi-Fi của gia đình.
- **Xác thực (Authentication & Authorization):**
 - **Yêu cầu:** Người dùng phải **đăng nhập bằng tài khoản và mật khẩu** để sử dụng ứng dụng. Mỗi thiết bị ESP phải có một **mã xác thực (token) duy nhất** để kết nối với Nền tảng IoT.
 - **Diễn giải:** Việc xác thực người dùng đảm bảo chỉ những thành viên được cho phép mới có quyền xem và điều khiển hệ thống. Trong khi đó, mã xác thực thiết bị đóng vai trò như "mật khẩu" của riêng phần cứng, ngăn chặn kẻ xấu tạo ra một thiết bị ảo để gửi dữ liệu giả mạo hoặc cố gắng chiếm quyền điều khiển thiết bị thật.

4.3. Độ tin cậy (Reliability)

Độ tin cậy đo lường khả năng hoạt động ổn định và tự phục hồi của hệ thống khi gặp sự cố.

- **Khả năng tự phục hồi kết nối (Auto-reconnect):**
 - **Yêu cầu:** Khi bị mất kết nối Wi-Fi, thiết bị ESP phải có khả năng **tự động thử kết nối lại** sau mỗi 1-2 phút cho đến khi thành công.
 - **Diễn giải:** Sóng Wi-Fi trong gia đình, đặc biệt là ở khu vực sân vườn, có thể không ổn định. Yêu cầu này đảm bảo hệ thống không bị "chết" hoàn toàn chỉ vì một lần rớt mạng tạm thời. Nếu không có cơ chế này, mỗi lần mạng chập chờn, người dùng sẽ phải khởi động lại thiết bị bằng tay, điều này làm mất đi hoàn toàn tính "tự động" và tiện lợi của hệ thống.
- **Khả năng phục hồi sau mất điện (Power Outage Recovery):**
 - **Yêu cầu:** Sau khi có điện trở lại, thiết bị ESP phải **tự động khởi động lại, kết nối vào mạng và quay trở lại trạng thái hoạt động** trước đó mà không cần bất kỳ sự can thiệp nào từ người dùng.
 - **Diễn giải:** Đây là yêu cầu bắt buộc đối với một hệ thống tự động hóa. Nó đảm bảo rằng sau một sự cố mất điện, hệ thống có thể tự

mình tiếp tục nhiệm vụ tưới tiêu theo lịch hoặc theo quy tắc đã định sẵn, duy trì lợi ích "cài đặt và quên đi" (set-and-forget) cho người dùng.

4.4. Khả năng mở rộng (Scalability)

- **Kiến trúc module (Modular Architecture):**
 - **Yêu cầu:** Cấu trúc phần mềm và thiết kế phần cứng phải cho phép **dễ dàng mở rộng** trong tương lai, cụ thể là có thể thêm ít nhất **2-3 cảm biến hoặc máy bơm mới** mà không cần phải thiết kế lại toàn bộ hệ thống.
 - **Diễn giải:** Điều này có nghĩa là mã nguồn cần được viết theo dạng các hàm hoặc lớp có thể tái sử dụng. Ví dụ, việc thêm một cảm biến mới chỉ đơn giản là khai báo một đối tượng cảm biến mới và cấu hình chân cắm, thay vì phải viết lại toàn bộ logic đọc dữ liệu. Việc suy nghĩ trước về khả năng mở rộng sẽ tiết kiệm rất nhiều công sức nếu người dùng muốn nâng cấp hệ thống sau này.

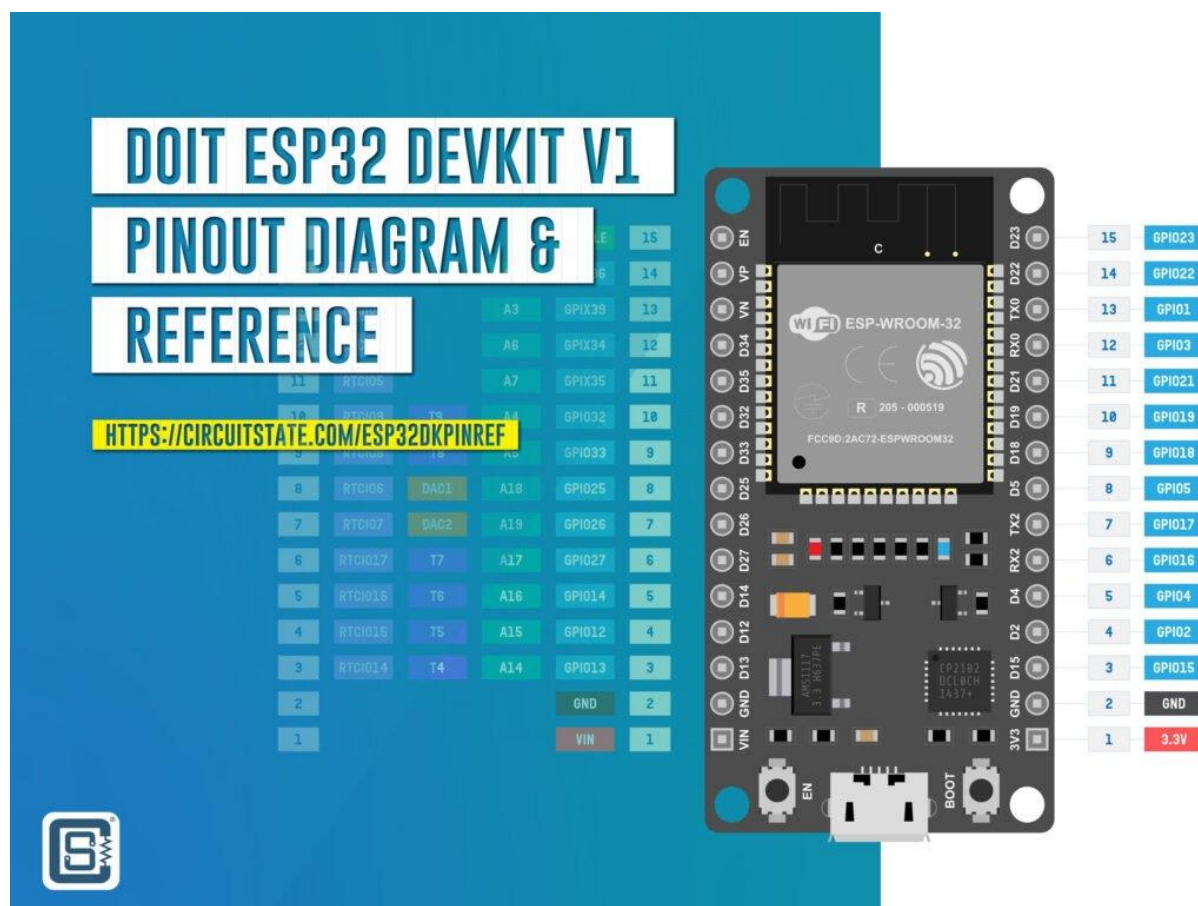
4.5. Chi phí và Năng lượng (Cost & Power)

- **Chi phí (Cost):**
 - **Yêu cầu:** Tổng chi phí đầu tư ban đầu (CAPEX) cho phần cứng phải **dưới 500,000 VNĐ**. Chi phí vận hành (OPEX) hàng tháng (tiền điện) phải ở mức không đáng kể.
 - **Diễn giải:** Việc tuân thủ ngân sách chặt chẽ là một ràng buộc quan trọng, đảm bảo giải pháp này thực sự là một dự án "nhà làm" (DIY) thiết thực và có tính khả thi cao về mặt tài chính cho một hộ gia đình.
- **Tối ưu năng lượng (Power Optimization):**
 - **Yêu cầu:** Phần mềm trên thiết bị ESP phải tận dụng **chế độ ngủ sâu (deep sleep)** giữa các lần đo và gửi dữ liệu.
 - **Diễn giải:** Thiết bị không cần phải hoạt động với 100% công suất liên tục. Bằng cách cho vi điều khiển "ngủ" trong phần lớn thời gian và chỉ "thức dậy" vài giây để đo đạc rồi gửi dữ liệu, hệ thống có thể giảm mức tiêu thụ năng lượng xuống hàng chục, thậm chí hàng trăm lần. Điều này đặc biệt quan trọng nếu trong tương lai hệ thống được cấp nguồn bằng pin hoặc năng lượng mặt trời.

CHƯƠNG 5: CÔNG NGHỆ SỬ DỤNG

5.1. Phần cứng

5.1.1. ESP32



Hình 5.1: Kiến trúc tổng quan của bo mạch ESP32 DEVKIT V1

Để đáp ứng các yêu cầu khắt khe về hiệu năng xử lý thời gian thực, khả năng kết nối IoT đa giao thức và độ ổn định trong môi trường công nghiệp/nông nghiệp, nhóm đã lựa chọn bo mạch phát triển **ESP32 DevKit V1** (phiên bản do DOIT sản xuất) làm bộ điều khiển trung tâm. Đây là một giải pháp SoC (System on Chip) toàn diện, tích hợp cao được phát triển trên nền tảng chip ESP32-WROOM-32 của Espressif Systems.

a) Tổng quan kiến trúc: ESP32 là một vi điều khiển mạnh mẽ, được thiết kế để giải quyết các bài toán IoT phức tạp mà các dòng vi điều khiển 8-bit truyền thống (như Arduino Uno/Nano) không thể đáp ứng. Kiến trúc của nó bao gồm hai lõi xử lý (Dual-core) hoạt động độc lập, cho phép phân chia tác vụ: một lõi xử lý giao thức mạng (Wi-Fi/Bluetooth stack) và một lõi xử lý logic ứng dụng, đảm bảo hệ thống không bị "treo" khi mất kết nối mạng.

b) Thông số kỹ thuật chi tiết (Datasheet Spec):

- **Vi xử lý (Microprocessor):**
 - Chip: ESP32-D0WDQ6 (trong module ESP-WROOM-32).
 - Kiến trúc: Xtensa® 32-bit LX6 hai lõi (Dual-core).
 - Xung nhịp: Có thể điều chỉnh từ 80 MHz đến **240 MHz**.
 - Hiệu năng: Lên đến 600 DMIPS (Dhrystone Million Instructions Per Second).
- **Bộ nhớ (Memory):**
 - **ROM:** 448 KB (cho quá trình khởi động và các hàm lõi).
 - **SRAM:** 520 KB (cho dữ liệu và lệnh).
 - **Flash:** 4 MB (SPI Flash gắn ngoài, dùng để lưu trữ chương trình firmware và file hệ thống SPIFFS/LittleFS).
- **Kết nối không dây (Wireless Connectivity):**
 - **Wi-Fi:**
 - Chuẩn: 802.11 b/g/n (802.11n lên đến 150 Mbps).
 - Tần số: 2.4 GHz ~ 2.5 GHz.
 - Chế độ: Station (STA), SoftAP, và SoftAP+STA.
 - Bảo mật: WPA/WPA2/WPA3.
 - **Bluetooth:**
 - Hỗ trợ v4.2 BR/EDR và Bluetooth Low Energy (BLE) - rất hữu ích cho việc cấu hình thiết bị qua điện thoại khi chưa có Wi-Fi.
- **Quản lý năng lượng (Power Management):**
 - Điện áp hoạt động (Chip): 2.3V ~ 3.6V (Logic 3.3V).
 - Nguồn cấp cho bo mạch (Input): 5V DC qua cổng Micro USB hoặc chân VIN (tích hợp IC ổn áp AMS1117-3.3 chuyển đổi 5V xuống 3.3V, dòng tối đa 800mA).
 - Dòng tiêu thụ trung bình: ~80mA (khi có Wi-Fi).
 - Chế độ Deep Sleep: Tiêu thụ chỉ ~10 μ A (phù hợp cho các ứng dụng chạy pin).

c) Các giao tiếp ngoại vi (Peripherals) sử dụng trong dự án: Bo mạch ESP32 cung cấp tới 30 chân GPIO, hỗ trợ nhiều giao thức giao tiếp phần cứng. Trong dự án này, các chân chức năng sau được khai thác:

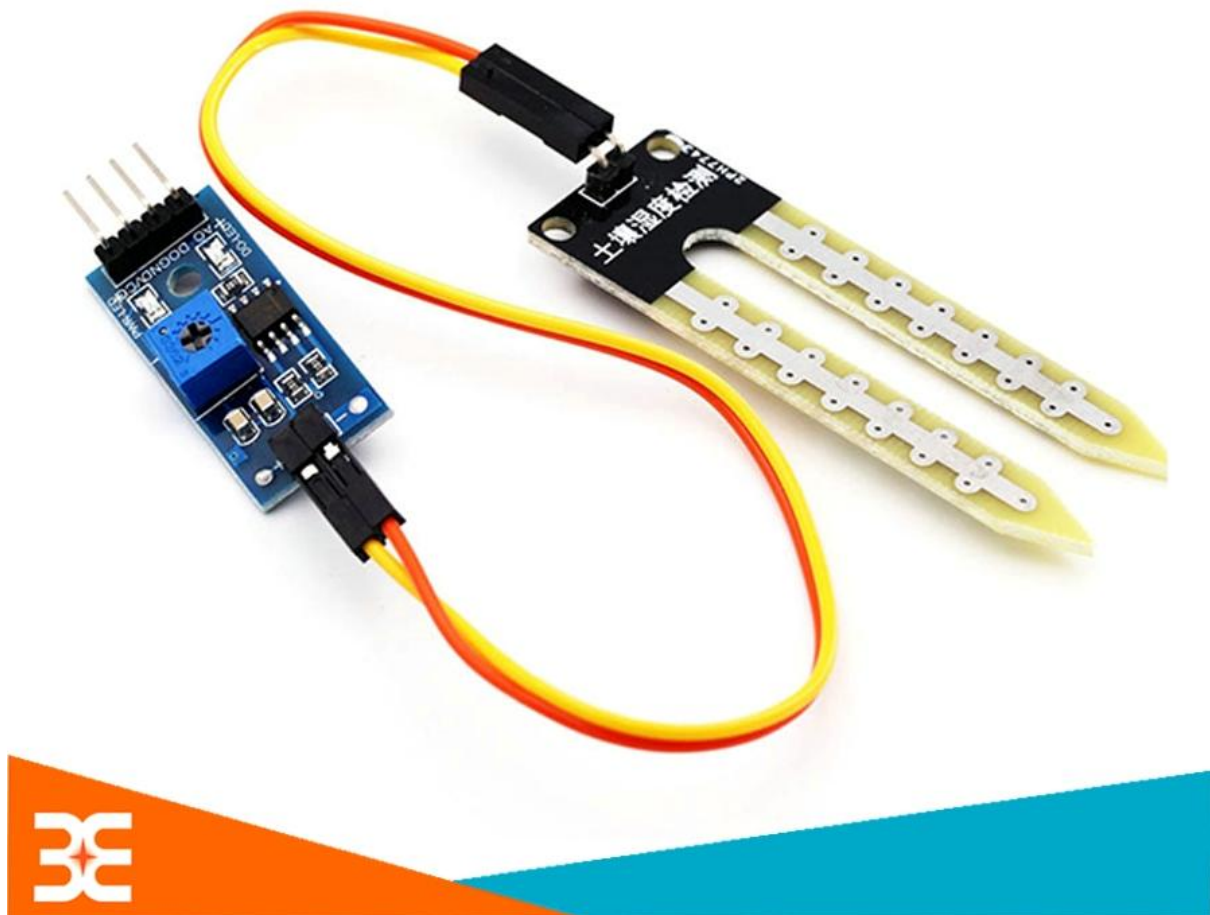
- **Bộ chuyển đổi Tương tự - Số (ADC - Analog to Digital Converter):**
 - Sử dụng **ADC1_CH6 (GPIO 34)**: ESP32 tích hợp bộ ADC 12-bit (SAR ADC), cho phép đọc giá trị điện áp với độ phân giải 4096 mức (0-4095). Điều này giúp đọc tín hiệu từ *Cảm biến độ ẩm đất* chính xác hơn nhiều so với ADC 10-bit của Arduino (0-1023).
 - *Lưu ý kỹ thuật*: Chỉ sử dụng ADC1 khi Wi-Fi đang hoạt động (ADC2 không khả dụng khi dùng Wi-Fi).
- **Giao tiếp GPIO (General Purpose Input/Output):**
 - **GPIO 4**: Cấu hình Input/Output cho giao thức 1-wire (tùy biến) để đọc dữ liệu nhiệt độ/độ ẩm từ cảm biến *DHT11*.

- **GPIO 5:** Cấu hình Digital Output để điều khiển mức logic (High/Low) cho *Module Relay*. Dòng ra của chân GPIO (max 12mA) đủ để kích hoạt transistor/optocoupler trên module relay.

d) Lý do lựa chọn ESP32 so với ESP8266/Arduino:

1. **Đa nhiệm (Multitasking):** Nhờ FreeRTOS chạy trên 2 lõi, ESP32 có thể vừa duy trì kết nối MQTT ổn định, vừa đọc cảm biến liên tục mà không bị chặn (blocking).
2. **Số lượng chân:** Nhiều chân Analog và Digital hơn, cho phép mở rộng thêm nhiều cảm biến trong tương lai (như cảm biến ánh sáng, mực nước) mà không cần mạch mở rộng.
3. **Bluetooth:** Tích hợp sẵn Bluetooth giúp dễ dàng phát triển tính năng "Provisioning" (cài đặt Wi-Fi cho thiết bị qua App điện thoại) sau này.

5.1.2. Cảm biến đo độ ẩm đất (FC28)



Hình 5.1.2: Cảm biến FC28

Tổng quan: Cảm biến phát hiện độ ẩm đất, bình thường đầu ra mức thấp, khi đất thiếu nước đầu ra sẽ mức cao. Module có thể sử dụng để tưới nước tự động. Độ nhạy của cảm biến độ ẩm đất có thể điều chỉnh được (Bằng cách điều chỉnh biến trở màu xanh trên board mạch)

Phần đầu dò được cắm vào đất để phát hiện độ ẩm, khi độ ẩm của đất đạt ngưỡng thiết lập, đầu ra DO sẽ chuyển từ mức thấp lên mức cao.

Thông Số Kỹ Thuật Module Cảm Biến Độ Ẩm Đất:

- Điện áp làm việc 3.3V ~ 5V
- Có lỗ cố định để lắp đặt thuận tiện
- PCB có kích thước nhỏ 3.2 x 1.4 cm

- Sử dụng chip LM393 để so sánh, ổn định làm việc
- Đầu kết nối sử dụng 3 dây
- VCC 3.3V ~ 5V
- GND GND của nguồn ngoài
- DO Đầu ra tín hiệu số (mức cao hoặc mức thấp)
- AO Đầu ra tín hiệu tương tự (Analog)

5.1.3. Cảm biến đo độ ẩm không khí (DHT11)



Hình 5.1.3: Cảm biến DHT11

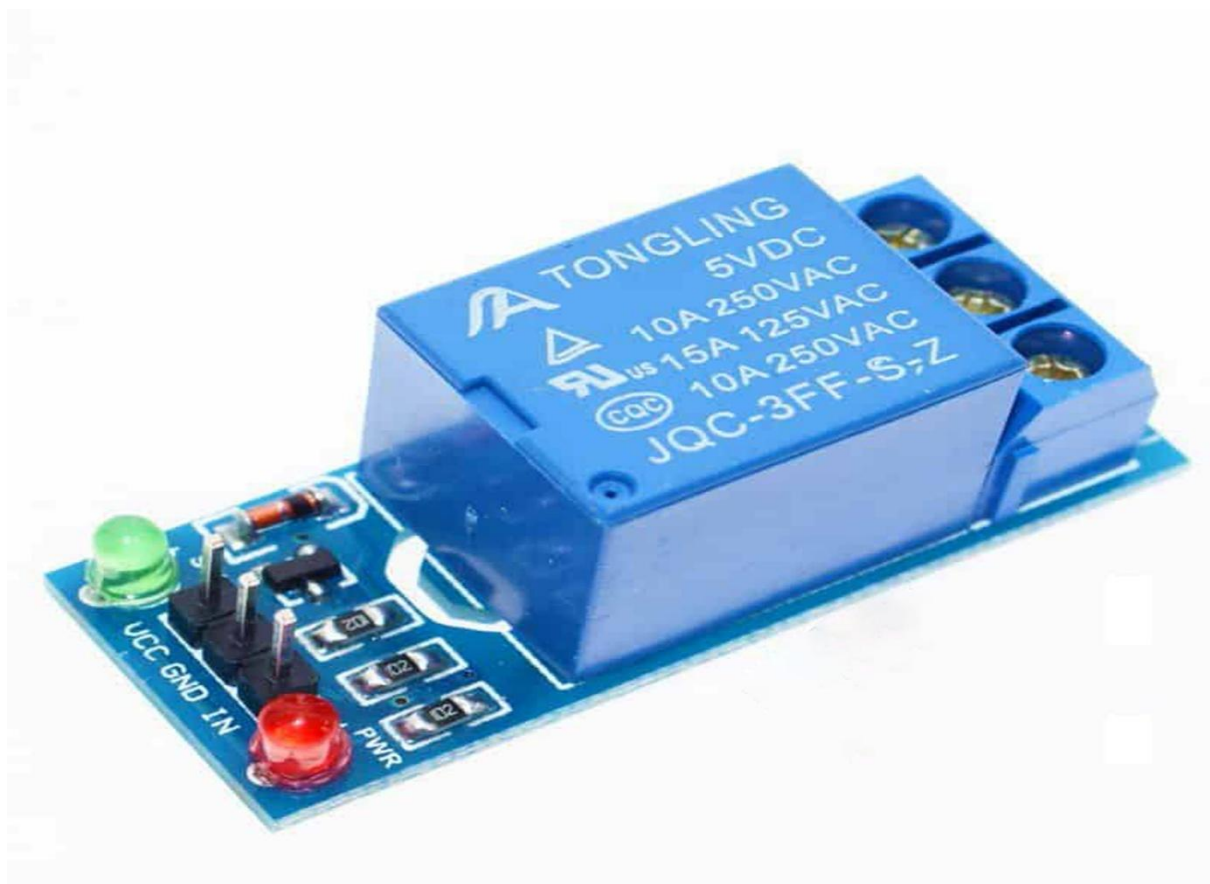
Tổng quan: DHT11 là cảm biến nhiệt độ và độ ẩm giao tiếp với 1 chân dữ liệu, DHT11 đo được giá trị độ ẩm từ 20% đến 90%RH và nhiệt độ từ 0oC đến 50oC, độ chính xác: $\pm 5\%RH$ và $\pm 2oC$.

Xin lưu ý chỉ sử dụng cảm biến độ ẩm nhiệt độ DHT11 trong môi trường độ ẩm thuần là hơi nước, các môi trường đặc biệt ủ kín như ủ tỏi đen, ủ yếm khí...sẽ sinh ra nấm và vi khuẩn bám lên bề mặt cảm biến làm hư hỏng cảm biến.

Thông số kĩ thuật:

- Điện áp hoạt động: 5VDC.
- Chuẩn giao tiếp: TTL, 1 wire.
- Khoảng đo độ ẩm: 20%-80%RH sai số $\pm 5\%RH$.
- Khoảng đo nhiệt độ: 0-50°C sai số $\pm 2^{\circ}C$.
- Tần số lấy mẫu tối đa 1Hz (1 giây / lần).
- Kích thước: 28mm x 12mm x10m.
- DHT11 có 4 chân: VCC, DATA, NC, GND
- Module DHT11 đã được gắn sẵn điện trở và led báo nguồn, nên có 3 chân.
- VCC: Nguồn 3.3 - 5.5VDC
- DATA: Chân dữ liệu
- GND: Nối đất, cực âm

5.1.4. Module Relay Shield 5V (loại 1 kênh)



Hình 5.1.4: Module Relay Shield 5V (loại 1 kênh)

Tổng quan: Module 1 Relay gồm 1 rơ le hoạt động tại điện áp 5VDC, chịu được hiệu điện thế lên đến 250VAC 10A. Module 1 relay kích mức thấp được thiết kế chắc chắn, khả năng cách điện tốt. Trên module đã có sẵn mạch kích relay sử dụng transistor và IC cách ly quang giúp cách ly hoàn toàn mạch điều khiển (vi điều khiển) với rơ le bảo đảm vi điều khiển hoạt động ổn định.

Có sẵn header rất tiện dụng khi kết nối với vi điều khiển. Mô-đun này tuân thủ các tiêu chuẩn an toàn quốc tế, khu vực đầu vào và khu vực tải có các khe cách ly

Thông số kỹ thuật

- Mức logic: 0V (GND)
- Điều khiển đóng ngắt điện DC hoặc AC, bạn có thể điều khiển tải AC 220 V 10A
- Có tiếp điểm thường mở và thường đóng:
- NO: thường mở (khi kích tiếp điểm đóng lại)
- COM: chung
- NC: Thường đóng (khi kích tiếp điểm mở ra)
- Đầu vào:

- Điện áp nuôi : 5VDC
- Tín hiệu vào điều khiển: 0V
 - Tín hiệu là 0: thì Relay đóng
 - Tín hiệu là 1 : thì Relay mở
- Đầu ra:
 - Tiếp điểm relay 220V 10A (Lưu ý tiếp điểm , không phải điện áp ra)
 - NC: Thường đóng
 - NO: Thường mở
 - COM: Chân chung

5.2. Phần mềm

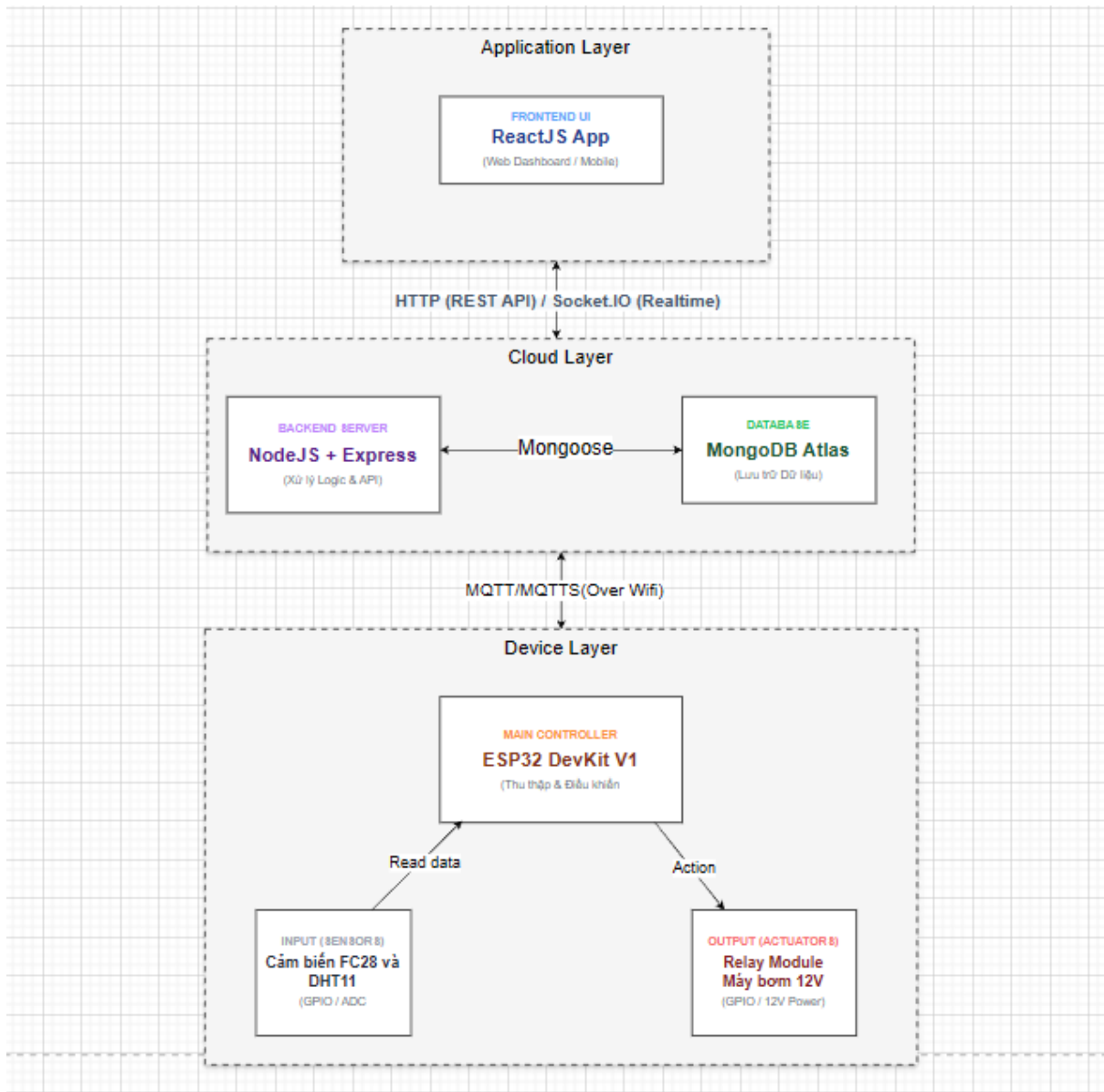
- **Môi trường phát triển:**
 - **Arduino IDE:** Được sử dụng để lập trình và nạp chương trình điều khiển và vi điều khiển (ESP32/Arduino). Đây là công cụ chính để viết mã, biên dịch và quản lý kết nối với phần cứng IoT.
 - **Visual Studio Code:** Trình soạn thảo mã nguồn hiện đại, hỗ trợ nhiều ngôn ngữ lập trình và có nhiều tiện ích (extension). VS Code được sử dụng để phát triển cả frontend(ReactJS) và backend(NodeJS), đồng thời dễ dàng quản lý dự án với Git.
- **Frontend:**
 - **ReactJS:** Dùng để xây dựng giao diện người dùng (UI), cho phép hiển thị trực quan trạng thái của các thiết bị IoT, đồng thời cung cấp các nút điều khiển để tương tác trực tiếp với hệ thống.
 - **Ngôn ngữ:** JavaScript - Ngôn ngữ chính để viết Logic cho giao diện, xử lý sự kiện và gọi API đến Backend Server.
- **Backend:**
 - **NodeJS(22.17.1) với ExpressJS:** NodeJS cung cấp môi trường chạy JavaScript phía Server. ExpressJS là framework nhẹ trên NodeJS, được sử dụng để xây dựng các API RESTful, giúp kết nối frontend với cơ sở dữ liệu và các thiết bị.
 - **Ngôn ngữ: JavaScript:** Dùng để viết các dịch vụ Backend, xử lý yêu cầu từ Frontend, giao tiếp với cơ sở dữ liệu, và quản lý luồng dữ liệu từ các thiết bị IoT.
- **Database:**

- **MongoDB Atlas:** Dịch vụ cơ sở dữ liệu NoSQL trên nền tảng đám mây. MongoDB Atlas giúp lưu trữ dữ liệu từ các thiết bị IoT (ví dụ: nhiệt độ, độ ẩm, trạng thái thiết bị), đồng bộ với Backend và hỗ trợ truy xuất nhanh chóng để hiển thị trên frontend. - Version Control:
- **Git (Github Server):** Dùng để quản lý mã nguồn, theo dõi thay đổi và hỗ trợ làm việc nhóm. Github Server đóng vai trò là kho lưu trữ trung tâm, giúp các thành viên dễ dàng chia sẻ, kiểm soát phiên bản, và cộng tác hiệu quả.
- **Giao thức:**
 - **Giao thức MQTT (Message Queuing Telemetry Transport):**
 - **Khái niệm:** MQTT là một giao thức nhắn tin (messaging protocol) hoạt động theo mô hình Publish/Subscribe (Xuất bản/Đăng ký), được thiết kế cực kỳ nhẹ (lightweight) để tối ưu hóa băng thông và năng lượng.
 - **Vai trò trong hệ thống:**
 - Đây là giao thức chính để giao tiếp giữa Thiết bị (ESP32) và Cloud Server.
 - ESP32 đóng vai trò là một MQTT Client, thực hiện việc Publish (gửi) các dữ liệu cảm biến (nhiệt độ, độ ẩm) lên một Topic cụ thể trên MQTT Broker.
 - Đồng thời, ESP32 cũng Subscribe (đăng ký) vào một Topic điều khiển để nhận lệnh Bật/Tắt máy bơm từ Server ngay lập tức.
 - **Ưu điểm nổi bật:**
 - **Tiết kiệm băng thông:** Kích thước gói tin rất nhỏ, phù hợp với mạng Wi-Fi chậm chạp hoặc băng thông thấp.
 - **Thời gian thực (Real-time):** Độ trễ rất thấp, giúp lệnh điều khiển từ App xuống thiết bị được thực thi gần như tức thì.
 - **Độ tin cậy:** Hỗ trợ các mức chất lượng dịch vụ (QoS) để đảm bảo tin nhắn không bị mất.
 - **Giao thức HTTP (Hypertext Transfer Protocol)**

- **Khái niệm:** HTTP là giao thức truyền tải siêu văn bản, hoạt động theo mô hình Request/Response (Yêu cầu/Phản hồi), là nền tảng của World Wide Web.
- **Vai trò trong hệ thống:**
 - Được sử dụng chủ yếu để kết nối giữa Frontend (ReactJS App) và Backend (NodeJS Server) thông qua chuẩn thiết kế RESTful API.
 - Khi người dùng mở ứng dụng, Frontend sẽ gửi các Request (GET, POST, PUT...) lên Server để lấy danh sách lịch sử tưới, thông tin người dùng, hoặc gửi yêu cầu đăng nhập.
 - Server xử lý và trả về Response chứa dữ liệu JSON cho Frontend hiển thị.
- **Ưu điểm nổi bật:**
 - **Phổ biến và Tương thích:** Được hỗ trợ bởi mọi trình duyệt web và thiết bị di động.
 - **Dễ dàng phát triển:** Cấu trúc rõ ràng, dễ dàng tích hợp với các cơ sở dữ liệu và hệ thống khác.

CHƯƠNG 6: THIẾT KẾ HỆ THỐNG

6.1. Sơ đồ khối hệ thống



Hình 6.1: Sơ đồ khối hệ thống

Dựa trên yêu cầu bài toán và công nghệ lựa chọn, nhóm đã thiết kế hệ thống theo mô hình kiến trúc 3 tầng (3-Tier Architecture), bao gồm: Tầng Ứng dụng (Application Layer), Tầng Đám mây (Cloud Layer) và Tầng Thiết bị (Device Layer). Các tầng này hoạt động độc lập và giao tiếp với nhau thông qua các giao thức chuẩn hóa, đảm bảo tính ổn định, bảo mật và khả năng mở rộng.

(Chèn hình ảnh sơ đồ khối của bạn tại đây)

1. Tầng Ứng dụng (Application Layer) Đây là tầng giao diện người dùng, nơi người dùng cuối tương tác với hệ thống.

- **Thành phần chính: Frontend UI (ReactJS App):** Ứng dụng được xây

dựng bằng thư viện ReactJS, hoạt động trên nền tảng Web Dashboard và Mobile.

- **Chức năng:**
 - Hiển thị dữ liệu thời gian thực (nhiệt độ, độ ẩm) dưới dạng biểu đồ trực quan.
 - Cung cấp các nút điều khiển để gửi lệnh Bật/Tắt máy bơm thủ công.
 - Quản lý cấu hình hệ thống và cài đặt ngưỡng tự động.
- **Giao tiếp:** Kết nối với Tầng Đám mây thông qua:
 - **HTTP (REST API):** Cho các tác vụ yêu cầu/phản hồi truyền thống (đăng nhập, lấy lịch sử).
 - **Socket.IO (Realtime):** Để nhận dữ liệu cập nhật tức thì từ server mà không cần tải lại trang.

2. Tầng Đám mây (Cloud Layer) Đây là tầng trung gian chịu trách nhiệm xử lý logic nghiệp vụ và lưu trữ dữ liệu.

- **Backend Server (NodeJS + Express):** * Đóng vai trò là bộ não trung tâm xử lý các API request từ Frontend.
 - Hoạt động như một cầu nối (Bridge) để nhận dữ liệu MQTT từ thiết bị và đẩy xuống Frontend qua Socket.IO.
- **Database (MongoDB Atlas):** Cơ sở dữ liệu NoSQL trên nền tảng đám mây.
 - Chịu trách nhiệm lưu trữ toàn bộ dữ liệu cảm biến theo thời gian (Time-series data), thông tin người dùng và trạng thái thiết bị.
 - Kết nối với Backend thông qua thư viện **Mongoose**.

3. Tầng Thiết bị (Device Layer) Đây là tầng vật lý, tiếp xúc trực tiếp với môi trường tại khu vực canh tác.

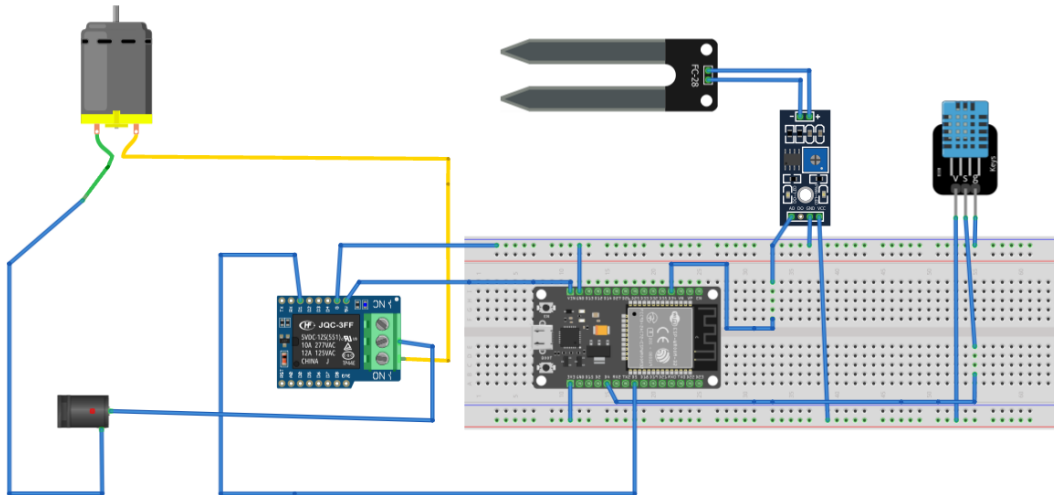
- **Bộ điều khiển trung tâm (Main Controller - ESP32 DevKit V1):**
 - Vi điều khiển chịu trách nhiệm thu thập dữ liệu và thực thi lệnh.
 - Kết nối Internet thông qua Wi-Fi và giao tiếp với Cloud bằng giao thức **MQTT/MQTTS** (Giao thức nhẹ, chuyên dụng cho IoT).
- **Khối Đầu vào (Input - Sensors):**
 - Gồm cảm biến độ ẩm đất (FC-28) và cảm biến nhiệt ẩm không khí (DHT11).
 - Gửi tín hiệu Analog/Digital về ESP32 (Read data).
- **Khối Đầu ra (Output - Actuators):**
 - Gồm Module Relay và Máy bơm 12V.
 - Nhận tín hiệu điều khiển (Action) từ ESP32 để thực hiện hành động tưới tiêu (GPIO/12V Power).

4. Luồng hoạt động chính:

- **Luồng Giám sát:** Cảm biến -> ESP32 -> (MQTT) -> Cloud Layer -> (Lưu DB & Socket.IO) -> Application Layer (Hiển thị).

- **Luồng Điều khiển:** Người dùng (App) -> (HTTP) -> Cloud Layer -> (MQTT) -> ESP32 -> Kích hoạt Relay -> Bơm chạy.

6.2. Sơ đồ nguyên lý



Hình 6.2: Sơ đồ nguyên lý

1. Khối Điều khiển Trung tâm (Main Controller): Đây là "bộ não" của toàn bộ hệ thống, chịu trách nhiệm thu thập dữ liệu môi trường, xử lý logic tự động và giao tiếp với Cloud Server.

- Thiết bị: Bo mạch phát triển ESP32 DevKit V1.
- Nguồn cấp: 5V DC thông qua cổng Micro USB (từ adapter điện thoại hoặc nguồn dự phòng).
- Chức năng:
 - Đọc tín hiệu Analog từ cảm biến độ ẩm đất.
 - Đọc tín hiệu Digital từ cảm biến nhiệt ẩm DHT11.
 - Điều khiển đóng/ngắt Relay để vận hành máy bơm.
 - Kết nối Wi-Fi để truyền nhận dữ liệu qua giao thức MQTT.

2. Khối Cảm biến (Input Block): Hệ thống sử dụng hai loại cảm biến để giám sát môi trường thời gian thực:

- Cảm biến Độ ẩm đất (Soil Moisture Sensor - FC-28):
 - Kết nối: Chân tín hiệu Analog (AO) được nối vào chân GPIO 34 (D34) của ESP32. Đây là chân ADC (Analog-to-Digital Converter) có độ phân giải 12-bit, cho phép đo độ ẩm đất chính xác.
 - Nguồn nuôi: 3.3V (Lấy từ chân 3V3 của ESP32).
- Cảm biến Nhiệt độ & Độ ẩm không khí (DHT11):
 - Kết nối: Chân dữ liệu (DATA) được nối vào chân GPIO 4 (D4) của ESP32.
 - Nguồn nuôi: 3.3V (Lấy từ chân 3V3 của ESP32).

3. Khối Chấp hành & Mạch Động lực (Actuator & Power Circuit): Để đảm bảo an toàn tuyệt đối cho vi điều khiển và người sử dụng, hệ thống sử dụng cơ chế cách ly nguồn điện cho thiết bị công suất lớn (máy bơm).

- Module Relay 5V (1 kênh): Đóng vai trò là công tắc trung gian.
 - Tín hiệu điều khiển: Chân IN nối vào chân GPIO 5 (D5) của ESP32. Khi ESP32 xuất mức tín hiệu HIGH/LOW, relay sẽ đóng/ngắt mạch.
 - Nguồn nuôi mạch điều khiển: 5V (Lấy từ chân VIN của ESP32).
- Máy bơm nước 12V DC:
 - Kết nối: Được đấu nối tiếp qua tiếp điểm thường mở (NO - Normally Open) của Relay.
 - Nguồn động lực: Sử dụng một Adapter 12V DC riêng biệt.
- Nguyên lý hoạt động an toàn: Dòng điện 12V chỉ chạy trong vòng kín giữa Adapter -> Relay -> Máy bơm -> Adapter. Nó hoàn toàn tách biệt với mạch 5V/3.3V của ESP32, giúp tránh hiện tượng nhiễu điện hoặc sụt áp làm treo vi điều khiển.

6.3. Thiết kế cơ sở dữ liệu

Hệ thống sử dụng MongoDB làm hệ quản trị cơ sở dữ liệu chính. Dữ liệu được tổ chức thành các Collection (Bộ sưu tập) với cấu trúc linh hoạt (JSON-like documents) để phù hợp với đặc thù dữ liệu IoT (thay đổi nhanh, khối lượng lớn).

6.3.1. Mô hình Dữ liệu Tổng quan (Schema Overview)

Hệ thống bao gồm 5 Collection chính:

1. Users: Lưu trữ thông tin người dùng (Khách hàng, Admin).
2. Devices: Quản lý thông tin và cấu hình của thiết bị phần cứng (ESP32).
3. SensorData: Lưu trữ dữ liệu môi trường (Time-series data).
4. Notifications: Lưu trữ lịch sử cảnh báo.
5. Firmwares: Quản lý các phiên bản phần mềm cập nhật (OTA).

6.3.2. Chi tiết các Collection

a) Users (Người dùng)

- Mục đích: Phục vụ chức năng Đăng ký, Đăng nhập, Phân quyền (Use-case: Quản lý người dùng).
- Cấu trúc:

Tên trường (Field)	Kiểu dữ liệu	Mô tả	Ràng buộc (Constraint)
-----------------------	-----------------	-------	------------------------

_id	ObjectId	Khóa chính (Primary Key)	Tự động tạo
username	String	Tên hiển thị của người dùng	Required
email	String	Địa chỉ email dùng để đăng nhập	Required, Unique
password	String	Mật khẩu đã được mã hóa (Hashed)	Required
role	String	Vai trò người dùng	Enum: ['customer', 'admin'], Default: 'customer'
phone	String	Số điện thoại liên hệ	Optional
createdAt	Date	Thời gian tạo tài khoản	Default: Date.now

b) Devices (Thiết bị)

- Mục đích: Phục vụ chức năng Thêm thiết bị, Cấu hình, Lập lịch, Điều khiển (Use-case: Thêm và cấu hình thiết bị, Điều khiển thủ công, Lập lịch).
- Cấu trúc:

Tên trường (Field)	Kiểu dữ liệu	Mô tả	Ràng buộc (Constraint)
_id	ObjectId	Khóa chính	Tự động tạo
userId	ObjectId	ID người sở hữu thiết bị	Ref: Users
hardwareId	String	Mã định danh phần cứng (MAC Address)	Required, Unique
name	String	Tên gọi nhớ (VD: Vườn Ban Công)	Required
status.isOnline	Boolean	Trạng thái kết nối	Default: false
status.pumpState	Boolean	Trạng thái bơm (true=Bật, false=Tắt)	Default: false

config.mode	String	Chế độ hoạt động	Enum: ['MANUAL', 'AUTO'], Default: 'MANUAL'
config.minSoilMoisture	Number	Ngưỡng độ ẩm thấp để bật bơm (Auto)	Default: 30 (%)
config.maxSoilMoisture	Number	Ngưỡng độ ẩm cao để tắt bơm (Auto)	Default: 80 (%)
schedules	Array	Danh sách lịch tưới (Timer)	[{ startTime: String, duration: Number }]
createdAt	Date	Thời gian thêm thiết bị	Default: Date.now

c) SensorData (Dữ liệu Cảm biến)

- Mục đích: Phục vụ chức năng Xem số liệu thời gian thực, Xem báo cáo lịch sử, Xuất dữ liệu.
- Đặc điểm: Dữ liệu chuỗi thời gian (Time-series), ghi liên tục.
- Cấu trúc:

Tên trường (Field)	Kiểu dữ liệu	Mô tả	Ràng buộc (Constraint)
_id	ObjectId	Khóa chính	Tự động tạo
deviceId	ObjectId	ID thiết bị gửi dữ liệu	Ref: Devices, Index: True
temperature	Number	Nhiệt độ môi trường (°C)	Required
humidity	Number	Độ ẩm không khí (%)	Required
soilMoisture	Number	Độ ẩm đất (%)	Required
pumpState	Boolean	Trạng thái bơm tại thời điểm đo	Optional
timestamp	Date	Thời gian ghi nhận dữ liệu	Default: Date.now, Index: True

d) Notifications (Cảnh báo)

- Mục đích: Phục vụ chức năng Nhận cảnh báo (Use-case: Nhận cảnh báo).
- Cấu trúc:

Tên trường (Field)	Kiểu dữ liệu	Mô tả	Ràng buộc (Constraint)
_id	ObjectId	Khóa chính	Tự động tạo
userId	ObjectId	ID người nhận thông báo	Ref: Users
deviceId	ObjectId	ID thiết bị phát sinh cảnh báo	Ref: Devices
type	String	Loại cảnh báo	Enum: ['ALERT', 'INFO', 'ERROR']
title	String	Tiêu đề thông báo	Required
message	String	Nội dung chi tiết	Required
isRead	Boolean	Trạng thái đã đọc/chưa đọc	Default: false
createdAt	Date	Thời gian tạo thông báo	Default: Date.now

e) Firmwares (Phần mềm thiết bị)

- Mục đích: Phục vụ chức năng Cập nhật phần mềm từ xa (Use-case: OTA firmware update).
- Cấu trúc:

Tên trường (Field)	Kiểu dữ liệu	Mô tả	Ràng buộc (Constraint)
_id	ObjectId	Khóa chính	Tự động tạo
version	String	Số phiên bản (VD: 1.0.2)	Required
fileUrl	String	Đường dẫn tải file .bin	Required
description	String	Mô tả những thay đổi	Optional
targetHardware	String	Dành cho loại phần cứng nào	Default: 'ESP32_DEVKIT_V1'
uploadedBy	ObjectId	Admin thực hiện upload	Ref: Users
createdAt	Date	Thời gian phát hành	Default: Date.now

6.3.3. Mối quan hệ giữa các Collection (Entity Relationship)

- Users (1) - (N) Devices: Một người dùng có thể sở hữu và quản lý nhiều thiết bị (nhiều khu vườn).
- Devices (1) - (N) SensorData: Một thiết bị sẽ gửi về rất nhiều bản ghi dữ liệu cảm biến theo thời gian.
- Users (1) - (N) Notifications: Một người dùng nhận được nhiều thông báo từ hệ thống.
- Devices (1) - (N) Notifications: Cảnh báo được sinh ra từ một thiết bị cụ thể.

TÀI LIỆU THAM KHẢO

1. Espressif Systems, "ESP32 Series Datasheet," V4.6, 2024. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datash_eet_en.pdf.
2. OASIS, "MQTT Version 5.0," OASIS Standard, 2019. [Online]. Available: <http://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>.
3. R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Ph.D. dissertation, Univ. of California, Irvine, 2000.
4. Aosong Electronics, "DHT11 Humidity & Temperature Sensor Datasheet." [Online].
5. Slide IOT và Ứng dụng – TS Kim Ngọc Bách
6. Bài giảng môn IOT và Ứng dụng PTIT